UNIVERSITY OF CALIFORNIA
RIVERSIDE

Application of Direct Osmotherapy for Spinal Cord Injury and Other Aspects of Crowded Protein Osmotic Pressure

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Bioengineering

by

Christopher Samuel Hale

September 2018

Dissertation Committee:
Dr. Victor G. J. Rodgers, Chairperson
Dr. Devin K. Binder
Dr. William H. Grover

The Dissertation of Christopher Samuel Hale is approved:

Committee Chairperson

University of California, Riverside

ACKNOWLEDGEMENTS

I would like to express my sincerest gratitude and thanks to my advisor Dr. Victor G. J. Rodgers for providing me the opportunity to conduct research and for mentoring me throughout my time in his lab. You provided me with the ability to approach any problem, and I will always appreciate the work you did to develop and strengthen my skills as a researcher. Thank you for all the time you have invested in me over the years, I will always be grateful for it.

I would like to thank Dr. Devin M^{c}Bride for introducing me to research when I was an undergraduate. Much of the research in this dissertation is a continuation of the work you did at UC Riverside, but all of it stems from you asking for research volunteers at the end of a fluid mechanics discussion.

I would like to thank my committee members Dr. Devin Binder and Dr. William Grover for their guidance and for allowing me to work in their lab. I was able to gain knowledge and valuable skills by working with you and your labs. I am also grateful to the researchers who work in your lab for their support of the work in this dissertation. Jennifer Yonan, thank you for working with me on the creation of an osmotic transport device for spinal cord injury, and Heran Bhakta, thank you for the working with me to develop a densimitry method for measuring device extraction rates.

I would like to thank Dr. Dimitrios Morikis for allowing the use of some of his office space to work in, and for allowing me computational time on his laboratory computers. I greatly appreciate you allowing me a place to work when I needed one.

I would like to thank the B2K Group, especially the undergraduates that worked with me: Ramsey N. Batarseh, Roman Chaar, Jordan Hughey, and Kevin Vang, without all of your time and effort this research would not be what it is today.

I would like to thank the staff of the Bioengineering Department and Mrs. Hong Xu for providing support to me throughout my time at UCR. Additionally, I would like to thank my fellow colleagues: Troy Alva, Heran Bhakta, Alexandra Eastes, Dieanira Erudaitius, Zied Gaieb, Reed Harrison, Ryan Kozaka, Rohaine Hsu, Rohith Mohan, Danielle Ornelas, Jennifer Yang, Raymond Yeung, and Nehemiah Zewde for their support and for making my time at UC Riverside enjoyable.

I would like to single out and thank Heran Bhakta and Danielle Ornelas for all of their support over the years. We may not have continued in the same lab, but that did not stop you two from helping me anytime I needed it. You both spent many days and nights helping me, either by talking through a research problem, giving me feedback on a draft, or sharpening my presentation before a talk. I will always be grateful for the help you have provided me.

I would like to thank the funding sources that allowed this research to occur: the Nielsen Foundation, the UCR Graduate Research Mentorship Fellowship (GRMP) and the Jacques S. Yeager Endowment. Additionally, I would like to thank the Mechanical Engineering machine shop for allowing me to use their equipment for the development of osmometer prototypes.

I would like to thank the American Chemical Society (Copyright 2018) for permission to reprint (adapted) the material in Chapter 10 (Hale, C.S., Ornelas, D.N., Yang, J.S., Chang, L., Vang, K., Batarseh, R.N., Ozaki-Felt, N.U. and Rodgers, V.G., 2018. Interrogating the Osmotic Pressure of Self-Crowded Bovine Serum Albumin Solutions: Implications of Specific Monovalent Anion Effects Relative to the Hofmeister Series. *The Journal of Physical Chemistry B.*)

Last, but certainly not least, I would like to thank my family: my parents, Thomas and Mary Hale, my siblings, Michael Hale and Jennifer Perez, my brother-in-law, Christopher Perez, and my nephew Declan Perez, without all of your support I would not have been able to complete this dissertation. I am incredibly grateful for everything you have given me throughout my life. Thank you.

ABSTRACT OF THE DISSERTATION

Application of Direct Osmotherapy for Spinal Cord Injury and
Other Aspects of Crowded Protein Osmotic Pressure

by

Christopher Samuel Hale

Doctor of Philosophy, Graduate Program in Bioengineering
University of California, Riverside, September 2018
Dr. Victor G. J. Rodgers, Chairperson

In the first part of this dissertation, the development of an osmotic transport device (OTD) for the treatment of severe spinal cord injury is presented. Herein, an osmotic transport device was designed for the reduction of edema in spinal cord injury, in which many different limitations and issues needed to be addressed including chemical, physiological, and manufacturing. This design was analyzed by a finite element method to determine device parameter effects on extraction rate, and shown, with later in-vivo parameters, to have an extraction rate exceeding the volume required to reduce tissue water levels to uninjured levels. While this rate was almost an order of magnitude greater than the volume needed to be removed, calculations of physiological restrictions on device effectiveness showed there is a lowest tissue water content that the device could achieve was above uninjured levels.

Fabrication of the device was made with materials that would be biocompatible for the time course of edema. Following fabrication, the extraction rate of the device was analyzed via a densimitry method. Evaluation, with later *in-vivo* parameters, showed good agreement with the extraction rate obtained from finite element analysis. Finally, testing was done *in-vivo* to determine

the device's effect on the water content of an injured spinal cord and shown to significantly reduce water content in injured tissue.

The second part of this dissertation focuses on furthering the understanding of crowded protein osmotic pressure. Herein, a concentrating osmometer was developed and shown to provide concentrated osmotic pressure results similar to literature values in less time and requiring less protein. Additionally, the free-solvent based model was developed for ion hydration and for the determination of association values in protein-protein complexes.

Analysis was done of concentrated osmotic pressure for various sodium salts and showed large variations in osmotic pressure. Also, that smaller ion with large charge densities, like F^-, and Cl^-, bound to proteins in greater numbers and developed systems with higher protein hydrations. Conversely, larger ions with weaker charge densities, like I^- and SCN^-, had lower ion binding and systems with lower protein hydrations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Chapter 3

ρ	Density
u	Velocity
t	Time
p	Pressure
N_i	Total Flux of Species i
D	Diffusion Coefficient
C_i	Concentration of Species i
R_i	Reaction Rate of Species i
u_0	Velocity Entering the System
n	Flux Normal to the Surface
J_v	Water Flux
L_P	Hydraulic Permeability
A_i	Fitting Parameter
V_i	Initial, Uninjured Volume
V_a	Added Volume
L	Segment Length
R_i	Initial Segment Radius
R_n	New, Swollen Segment Radius
WC_i	Initial Water Content of Segment
WC_n	New Water Content of Segment

Greek:

μ	Dynamic Viscosity
σ	Reflection Coefficient
π	Osmotic Pressure

Subscripts:

P	Protein Component of Osmotic Pressure
D	Donnan Component of Osmotic Pressure

1	First fitting Parameter
2	Second Fitting Parameter

Chapter 5

D_i	Density of species i
V_i	Volume of species i
SV	Specific Volume of the Protein
J_v	Water Flux
L_P	Hydraulic Permeability
R_i	Resistance to Flux

Greek:

μ	Dynamic Viscosity
σ	Reflection Coefficient
π	Osmotic Pressure

Subscripts:

P	Protein Volume
S	Solvent Density
$S1$	Initial Solvent Volume
$S2$	Extracted Solvent Volume
$P1$	First Protein Solution Density
$P2$	Second Protein Solution Density

Chapter 9

C_n	Concentration of Solute
V_n	Volume of Solute Chamber
N_i^k	Initial Number of Moles of Species i, in Compartment k
N^k	Initial Total Number of Moles in Compartment k
R	Ideal Gas Constant
T	Temperature
$\overline{V_1}$	Specific Volume of Species 1

Greek:

π	Osmotic Pressure
ν_{ij}	Net Number Of Moles of Species i, Interacting with Species j

Superscripts:

I	Solvent Chamber
II	Protein Chamber

Subscripts (Equation 1):

1	Before Rotation
2	After Rotation

Subscripts (Equation 2):

1	Solvent
$2 \rightarrow (p+1)$	Proteins
$(p+2) \rightarrow N$	Salts
i	Individual Species
j	Individual Species
n	Number of Individual Species
p	Number of Individual Proteins

Chapter 10

N_i^k	Initial Number of Moles of Species i, in Compartment k
N^k	Initial Total Number of Moles in Compartment k
R	Ideal Gas Constant
T	Temperature
$\bar{V}_1$	Specific Volume of Species 1
M	Bulk Ionic Strength of the Solution
$SASA$	Solvent Accessible Surface Area

Greek:

$(\chi_1^k)_{FW}$	Mole Fraction of the Free Water in Compartment k
α	Ionic Strength Ratio
π	Osmotic Pressure
ν_{ij}	Net Number Of Moles of Species i, Interacting with Species j

Superscripts:

I	Solvent Chamber
II	Protein Chamber

Subscripts:

1	Water
2	Protein
3	Salt

Chapter 11

A	Molar Concentration of Species A
AB	Molar Concentration of Species Complex AB
B	Molar Concentration of Species B
K_A	Molar Association Constant
$N_{i,I}^k$	Total Moles of Solvent Species i, in Compartment k, Initially
$N_{i,E}^k$	Total Moles of Solvent Species i, in Compartment k, at Equilibrium
R	Ideal Gas Constant
$SASA$	Solvent Accessible Surface Area
T	Temperature
$\overline{V_1}$	Specific Volume of Species 1

Greek:

χ_i^k	Mole Fraction of Species i, in Compartment k
π	Osmotic Pressure
ν_{ij}	Net Number of Moles of Species i, Interacting with Species j

Superscripts:

I	Solvent Chamber
II	Protein Chamber

Subscripts:

1	Water
2	Protein_1
3	Protein-Protein Complex
4	Protein_2
C	Change
E	Equilibrium
I	Initial
P_1	Protein_2
P_2	Protein_1
PC	Protein-Protein Complex

PART 1.

DEVELOPMENT OF AN OSMOTIC TRANSPORT DEVICE FOR SPINAL CORD INJURY

CHAPTER 1.

INTRODUCTION

1.1 Spinal Cord Injury (SCI)

In the United States, 1.3 million people live with spinal cord injury (SCI), with an estimated 50,000 individuals added each year.[1] A majority of these injuries occur from motor vehicle accidents or accidents working, with the next largest percentage coming from sports related injuries. Depending on the location of the injury the disabling effect will vary.[2] Severe injuries high in the cervical spinal region, have the possibility of paralyzing the muscles responsible for breathing, and usually leads to death. However, even if that is not the case, an injury to cervical region will lead to paralysis of the legs and trunk with the potential to partially or complete paralyze the arms. Injuries lower, in the thoracic section of the spine, cause paralysis below the injury site, while injuries in the lumbar and sacral segments, the bottom of the spine, will lead to leg numbness and weakness. Regardless of the location, a severe injury at any level can cause a loss of bladder and bowel control.

1.2 Phases of Damage

SCI occurs in two phases. The first phase is the initial damage, often caused by a mechanical impact. Following this initial damage, there is a secondary phase consisting of a cascade of vascular, cellular, and biochemical events, including hemorrhage, edema, ischemia, cell death, oxidative stress, cyst formation, and metabolite abnormalities. If the hemorrhaging, metabolite abnormalities, and cell death result in apoptosis, the surrounding area will see increases in edema, the accumulation of water in the tissue that leads to swelling. Swelling is problematic for spinal injuries, as the spinal cord is surrounded by several membranes and bone that prevent tissue

expansion. Since the tissue is not able to expand beyond these restraints, an increase in pressure develops. Once pressure increases significantly, ischemic conditions will develop, where blood vessels collapse and are no longer able to supply nutrients to the surrounding cells. These nutrient losses cause the cells to experience an increased oxidative stress that leads to further cell death and decreased functionality below the injury site.[2]

1.3 Current Treatments

The current protocol for treatment of SCI, is to decompress and stabilize the spine in an effort to prevent further injury. While there is no widely accepted treatment to stop or prevent the secondary phase of SCI damage, methylprednisolone therapy has been used in an effort to reduce the secondary phase of damage. This treatment requires cortical steroids to be introduced systemically through intravenously injection within eight hours of injury. Unfortunately, this therapy has only been shown to have limited effectiveness due to ischemic conditions preventing blood flow of the drug to the injured tissue.

1.4 Proposed Treatment

The goal of our treatment is to prevent the edema by using a novel osmotic transport device (OTD) to remove water through an osmotically driven flux. The OTD can remove water in a controlled fashion, without damaging the underlying tissue, and while maintaining or reestablishing metabolite ratios. Use of this treatment will help to prevent ischemia, oxidative stress, cyst formation, and addition cell death.

1.5 Previous Work

Previously our lab has used OTDs to remove water from the brain.[3–5] Initial experiments focused on water intoxication in mice, where 30% body water of water was injected interperitoneally. These experiments compared untreated mice and OTD treated mice to mice given the gold standard of care, a craniectomy. Treatment began approximately 5 minutes post injection and concluded at 6 hours or when the mouse expired, at which time the tissue was analyzed for water content. While there was no significant difference between the brain water content for the different treatment groups, the time to expiration or sacrifice for the groups was significantly different. The untreated mice expired at 31 minutes, the craniectomy treated mice expired at 48 minutes and the OTD treated mice expired or were sacrificed at 333 minutes. This study showed that OTDs can have a significant impact on the removal of water from the brain.

The next study performed was a test of the OTD's effect on brain water content following severe traumatic brain injury (TBI). Here mice received a severe traumatic brain injury through a controlled cortical impact. Following impact, edema was allowed to form for 3 hours, after which time, treatment was administered for 2 hours. Following treatment, the mice were sacrificed and their brain water content was determined. The results showed an elevated but not significantly different brain water content between uninjured nice and injured mice treated with the OTD. All other treatment groups showed significantly higher water content levels. This study proved the device's effectiveness at reducing water content in the brain following TBI.

The third study also regarded the treatment of TBI, but this time looking at functional outcomes following treatment. After cortical impact injury, treatment followed, consisting of craniectomy for 24 hours and then 2 additional hours of craniectomy or OTD treatment. After which time the injury site was sealed and the animal was allowed to recover. Neurobehavioral testing was administered 1 day post treatment and was compared to testing done 1 day prior to injury. Testing

showed no significant difference between uninjured animals and injured animals treated with an OTD, while craniectomy showed significant differences to uninjured animals in five of the six analytics. This study showed that OTD improved neurobehavioral function when compared to craniectomy treated mice.

1.6 Viability Assessment

Before development of an OTD for the treatment of treat spinal cord injuries, the physiological and potential device parameters need to be evaluated to determine if OTD application would be a viable treatment. For this purpose, the physiology of mice brains to rat spinal cords are compared (Table 1). A mouse brain with a mass of 0.4 g, where half of the brain is injured and the tissue water content increases by 1% would account for an increase of 2 µL of water. The OTD was able to reduce that 0.25% increase over uninjured levels in 2 hours, a reduction of 1.5 µL of water.

Next, these reduction values need to be compared to the expected water increase in rat spinal cords to see if OTDs can be an effective treatment for SCI. A rat spinal cord with a mass of 0.7 g[6], where 17% of the cord is injured and experiencing a 6% increase in water content, would need 7.2 µL of fluid to be removed in order to bring tissue water content back to uninjured levels. Assuming an OTD removal rate similar to the TBI treatment of mice, the treatment time for SCI would be 10.4 hours (Table 1.1). This is reasonable treatment timeframe and therefore makes OTDs a viable therapy for SCI.

1.7 Scope of This Work

The focus of this work is the development of an osmotic transport device for the reduction of edema following spinal cord injury.

Herein, an osmotic transport device is designed and fabricated to be implantable in rats for the time course of severe edema. The device draws the excess fluid, gathering after injury, out of the tissue. The extraction capacity of the device is evaluated computationally to determine extraction rate dependencies on different parameters. Additionally, tissue physiology is evaluated to determine the lowest tissue water content expected to be achievable by the device.

Following computational evaluation, *in-vitro* evaluation, by a novel densimitry method, is used to determine the extraction rate of the device. Similarly, the hydraulic permeability of device components and spinal tissue are evaluated with this method. After *in-vitro* testing, the device is evaluated *in-vivo* and compared to uninjured and untreated animals. Extraction rates between the different evaluation methods are compared and an assessment of clinical viability is made.

Table 1.1: Viability Assessment of OTD Treatment for SCI Injury in Rats

	Mouse Brain	**Rat Spine**
Tissue Mass (g)	0.4[6]	0.7[6]
Tissue Injured (%)	50%[4]	17%*
Reduction in Water Content (%)	0.75%[4]	6%*
Water Mass to Remove (µL)	**1.5**	**7.2**
Treatment Time (h)	**2**[4]	**10.4**

* Data from Dr. Devin Binder's Lab (Jennifer Yonan)

CHAPTER 2.

DEVICE DESIGN

2.1 Function

An osmotic transport device (OTD) works through osmotically driven flux drawing water out of the tissue. To development this flux an osmolyte is restricted to one side of a semi-permeable membrane, creating a chemical potential difference (Figure 2.1: A). In order to reestablish chemical equilibrium, water moves across the membrane toward the osmolytes creating an osmotic flux (Figure 2.1: B). By placing the tissue, in contact to the membrane via hydrogel, on the opposite side of the membrane from the osmolyte solution, water can be drawn out of the tissue and into the device. Together the membrane, hydrogel, and osmotic solution create the fundamental chemistry behind the OTD treatment.

2.1.1. Osmotic Solution

As the osmolyte (Figure 2.2: (1)) determines the magnitude of the osmotic flux (Figure 2.2: (4)), it needs to be well studied to accurately achieve desired pressures. The osmolyte also needs to have a significantly larger molecular weight than the specie(s) targeted for removal from the tissue. By creating a range between the species, the molecular weight cutoff of the membrane can be selected to allow the targeted specie(s) to pass through the membrane unhindered, while the osmolyte is restricted. Selecting a membrane that restricts the osmolyte should prevent the osmotic agent from ever interacting with the tissue, however, in the unlikely case of device failure, the osmotic agent still needs to be biologically inert in order to ensure that the damaged tissue is not negatively affected should the two ever interact. Similarly, the solvent solution, containing the osmolyte, should mimic the uninjured fluid composition of the tissue, with the exception of

specie(s) targeted for removal. Beyond allowing for targeted specie(s) removal, the solvent solution acts as an added buffer to the tissue's solvent, preventing composition fluctuations.

2.1.2 Membrane

Since OTD efficacy is dependent on development of an osmotic gradient, selection of a membrane (Figure 2.2: (3)) with correct molecular weight cutoff and chemical composition is paramount. The molecular weight cutoff determines what species can transport between the two solutions. In order to ensure that the osmolyte is restricted from crossing the membrane and eliminating the osmotic gradient, the molecular weight cutoff needs to be significantly lower than the molecular weight of the osmolyte. Additionally, the chemical structure of the membrane is important as it determines how the membrane interacts with all the species it encounters. For this purpose, a hydrophilic membrane is selected to ensure that water flux toward the osmotic solution is not hindered. A final selection criterion for the membrane is the membrane backing. The backing needs to provide a sturdy support that limits bowing and keeps the membrane flat against the solution channel. If the membrane bows either into or away the solution chamber it changes the flow path of the osmotic solution and could impact the effectiveness of the device.

2.1.3 Hydrogel

For the device to work effectively it needs to be in contact with the tissue. While possible, conforming a membrane to the tissue would be prone to difficulties resulting from animal size variability. Additionally, if the shape of the solution chamber were to change, it could lead to unexpected changes in the solution chamber flow pattern and cause issues with the OTD performance. To ensure that the membrane remains in contact with the tissue, a hydrogel (Figure

2.2: (2)) is placed between the OTD membrane and the injured tissue. This allows for water flux and a maximized treatment area.

As a hydrogel that is made of a majority of water, having a higher water content than uninjured tissue, would potentially deliver water to the tissue, worsening the edema. On the opposite side, a hydrogel with a lower than normal water content, could remove excess water beyond the desired water arriving to cause edema, effectively dehydrating the tissue. Ideally, the hydrogel would match the water content of uninjured tissue, while still retaining the ability to conform to the tissue and be biologically inert to ensure the animal does not react negatively to its composition. Additionally, an ideal hydrogel would have a solution composition similar to that of uninjured tissue, allowing for only targeted removal of select specie(s).

2.2. Physiology

Physiology plays a large role in the design of an OTD, as an OTD works by way of an osmotic flux, it must be in contact with the injured tissue in order to create that osmotic gradient and draw fluid out. For SCI, in order for the OTD to be in contact with the tissue a laminectomy must be performed, where the bony lamina covering the spinal cord is removed.

2.2.1 Time Scale

To properly remove edema, treatment needs to last until edema is gone and does not return after treatment is ended. For SCI, that length is on the order of days. Therefore, any treatment needs to be designed for comparable time periods, as that duration may be required to properly treat edema. Due to the potential length of treatment, the animal is not anesthetized for the entire treatment duration, and has the potential to move around. Therefore the device needs to be fixed in place to ensure that the device-tissue interface is not misaligned by movement. For this purpose

surgical cement is used to fix the device to the lamina above and below the injury site. Additionally, due to the potentially long duration of treatment, keeping the surgical site open to the environment is likely to cause complication with infection or other issues. By designing the device to fit and work under the resealed tissue, environmental complications can be greatly diminished.

2.2.2. Implant

By making the device implantable under the resealed surgical site, addition size and shape limitations, based on the treated animal's size and physiology, are added to the OTD design criteria. These design limitations come from the tissue surrounding the spine, as well as the bone itself. The major limitation generated by the bone are the processes that protrude out of the spine toward the skin and create a proportionally large variation in bone height. To reduce this limitation, these processes can be removed after the laminectomy to allow for a relatively level treatment location and a greater maximum length for the device. The surrounding tissue determines the maximum width and height of the device. The muscle that originally surrounds the spine is pulled away during the laminectomy, but still surrounds the injury site. Having a device with too large of a width would potentially damage this muscle, as well as, being painful for the animal. Additionally, after laminectomy and device placement, the tissue that originally covered the spine needs to be stitched or stapled back together. This limits the device's height to the elasticity of the tissue originally on top of the spine.

2.3. Form Factor

The form of the device contributes greatly to the function of the device. Seemingly small changes to shape, angles, or locations can have a significant impact on the reliability and

effectiveness of the device. Moreover, addition or removal of functional elements can completely change the outcomes from device usage.

2.3.1. Solution Chamber

One issue resulting from osmotically driven flux is that drawing water into the osmotic solution causes dilution (Figure 2.2: (5)). This dilution (Figure 2.2: B) decreases the osmotic gradient and could potentially eliminate the osmotically flux altogether (Figure 2.2: (4*)). Additionally, an unstable osmotic gradient would make the device have an uncontrolled quality to it.

By introducing convective flow (Figure 2.2: C) of new undiluted osmotic solution into the solution chamber (Figure 2.2: (6)), replacing the diluted osmotic solution, a constant, controllable osmotic flow (Figure 2.2: (4)) would be achievable. In order to allow for a convective flow, two solution chamber ports (Figure 2.3: A), an inlet and outlet, need to transverse from below to above the resealed tissue. These ports allow for other equipment to be connected to the device as well as allow for the exchange of osmotic solution. The diameters of the inlet and outlet need to be small in order to allow for the tissue to be resealed, by stitching or stapling, but still large enough to remain sturdy and connect with the inflow and outflow tubes transporting the osmotic solution. These port have the potential to stretch the tissue, leaving gaps that allow for access below the skin which may cause issues in preventing infection. In order to reduce the chance of infection, an antibiotic ointment can be placed around the traversing locations.

2.3.2 Flow Channel

The flow channel is designed to maximize flow of fresh protein solution toward the membrane, but prevent that flow from driving fluid though the membrane toward the tissue. For

this purpose, the flow is redirected from perpendicular to the membrane to a more parallel flow direction, with a gentle slope toward the membrane (Figure 2.3: A). This allows the perpendicular change in flow direction to occur more gradually, while keeping a stagnation zone from developing in the corners of the chamber where the flow transitions occurs.

2.4. Fabrication

Designs can exist in any shape imaginable, however in order to build a design many limitations come into play. Size is often a limiting factor for designs, as the manufacturing processes available are limited, and something too small or too large may not have an applicable method for creation. Assembled designs, containing multiple individual components, often have issues with maintaining alignment or placement during the assembly process, and need to be designed around.

2.4.1. Membrane Holder

Centering the membrane on top of the solution chamber during fabricating can be difficult, as during the application of epoxy the membrane has a tendency to move with the epoxy. In order, to keep the membrane in place, small protrusions are added around the edge of the device (Figure 2.3: (1)). These protrusions act as blocking obstacles that stop the membrane from moving beyond that point.

2.4.2. Epoxy Traps

Another issue that arises during epoxying, is epoxy going under the area of membrane that is available for fluid transport. This reduces the effective surface area and the potential extraction

rate of the device. To prevent this, small cutouts (Figure 2.3: (2) and (3)) are designed into the device to provide a place for epoxy to go without effecting the transport surface area.

2.4.3. Structural Material Properties

The force from the tissue onto the device also needs to be accounted, as a disruption in flow can cause the device to fail. If the flow inlet is closed off, then fresh solution would stop flowing past the membrane, limiting the device's effectiveness. However, a closure at the outlet would cause delivery of water into the tissue, likely leading to a worse prognosis. In order to ensure that the flow is not disrupted by surrounding tissue, the structure around the flow must be rigid and non-deformable. This prevents the potential full or partial closure of the inlet/outlet flow paths. The downside of having a rigid structure is that the shape of the surrounding tissue does not exert an equal force over the device. This can lead to large pressure differentials across the tissue, producing pressure ulcers, and adding additional ailments to the patient. To minimize this potential, a flexible/deformable outer shell incases the rigid/non-deformable solution chamber, except for the inlet/outlet nozzles and the membrane. This shell allows for the device to deform, into the shape least stressful to the tissue, lowering pressure variations across the tissue, and potentially prevent pressure ulcers. Due to the potential for extended usage, both the internal rigid solution chamber and the flexible surrounding shell need to be biocompatible so as to not have a chemical impact on the surrounding tissue.

2.4.4. Construction Method

The size of the OTD restricts the construction of the device to molds and forms of 3D printing. For instance, the machine that 3D prints the rigid structure of the OTD creates layers of material 20 micron thick, with a minimum feature size of 152 microns. Therefore, using this

machine, the minimum horizontal wall/feature thickness of the device is 152 microns. The elastic shell is created from a mold and as such the rigid structure with the elastic shell needs to be removed from the mold, thus limiting the potential shapes of the elastic shell. Additionally, both of these techniques are only available for select materials, thereby limiting the design of the OTD to the materials available to be manufactured by these methods.

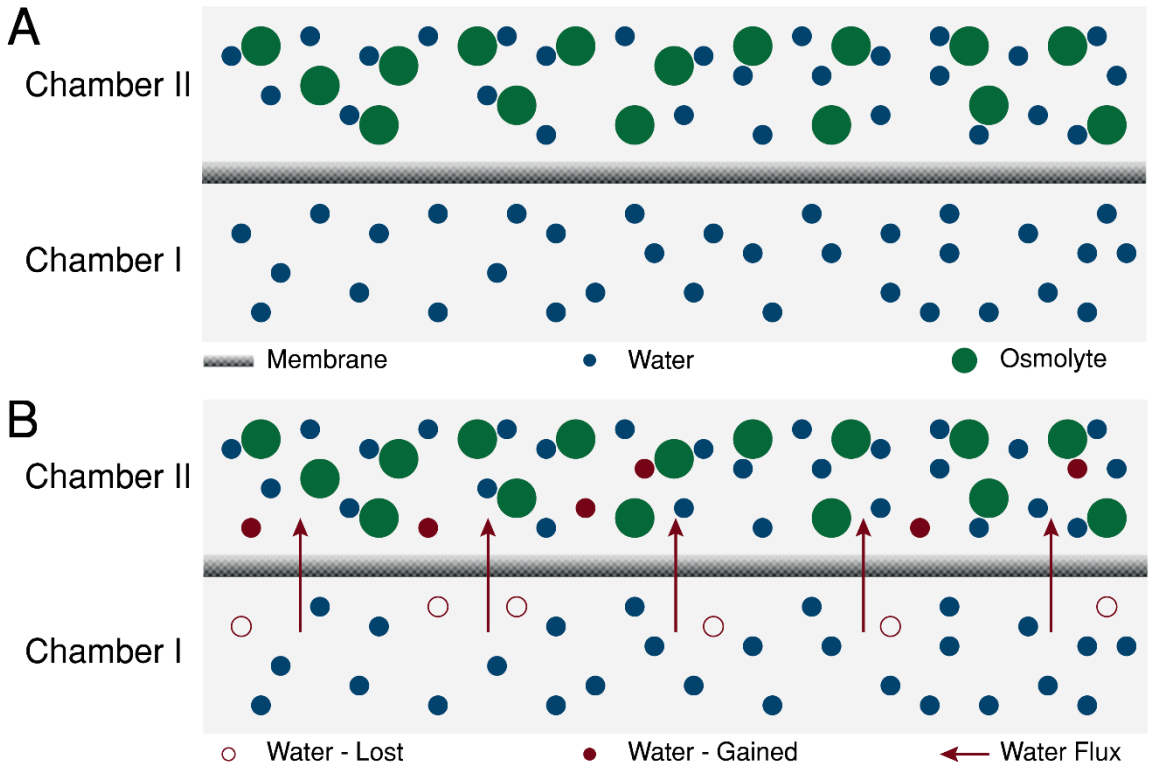


Figure 2.1: A) Unbalanced chemical potential between chamber I, containing only water, and chamber II, containing water and osmolytes. B) Balanced chemical potential between chamber I and chamber II, where some water from chamber I moved across the membrane to chamber II.

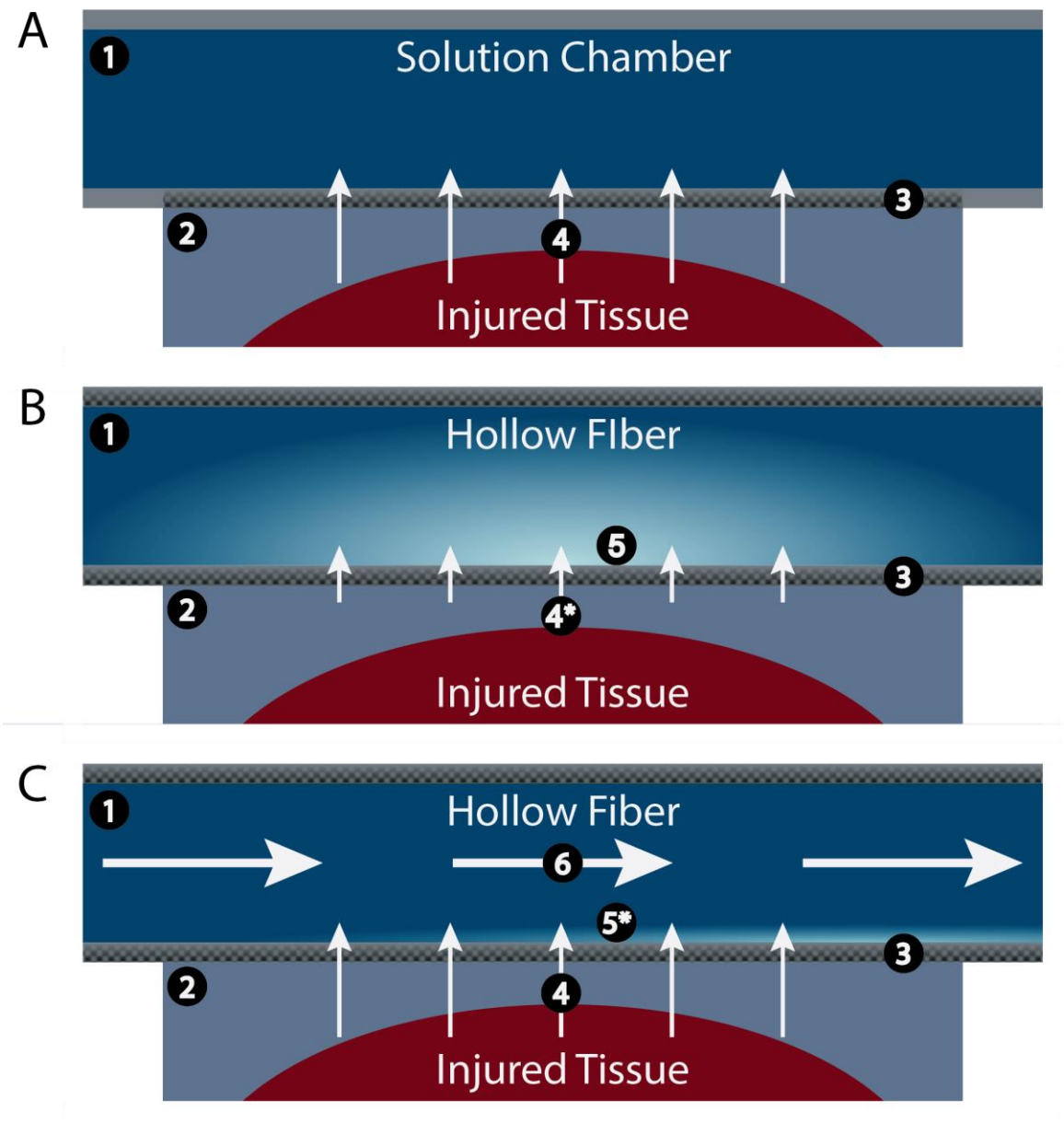


Figure 2.2: A) An osmotic transport device containing the (1) the osmotic solution, (2) the hydrogel, and the (3) membrane. These components combined cause an osmotic pressure gradient that draws fluid out of the injured tissue (4) and into the device. B) As fluid is drawn into the device the protein solution becomes diluted (5) and the osmotic pressure gradient lessens (4*). C) By introducing flow through the solution chamber (6), the protein solution is able to be replenished (5*) and the osmotic gradient returned to undiluted levels (4).

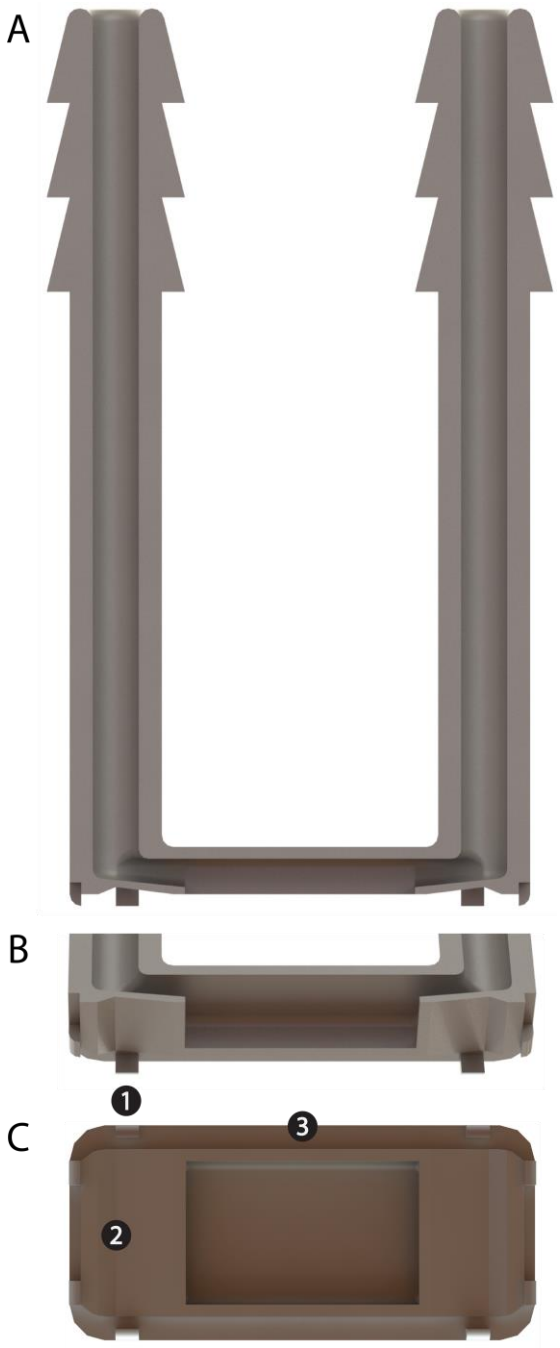


Figure 2.3: A) is the front view of the device, B) is an isometric view of the device, and C) is a view of the bottom of the device. Protrusions around the perimeter of the device (1) keep the membrane seated over the solution chamber during fabrication. A cutout (2) is made next to the solution chamber to allow excess epoxy to flow away from the solution chamber. Cutouts on the side of the device (3) allow for epoxy to seal along the edge and bottom of the membrane.

CHAPTER 3.

MATHEMATICAL ANALYSIS

3.1 Computational

Rationalizations for design criteria are necessary, however, design decisions should be validated before committing to them. A computational model (COMSOL Multiphysics® Version 5.2a, Burlington, MA, U.S.A.) is developed to validate design decisions as well as reduce experimental time when optimizing the device design.

3.1.1 Setup

3.1.1.1 Geometry

The 2-dimensional model is a cross-sectional cutout of the device made to the dimensions of the device (Figure 3.1: A). The inlet is designated at the upper left of the geometry, while the outlet is in the upper right, and the membrane lies at the bottom of the geometry.

The 3-dimensional model (Figure 3.1: C) was bisected symmetrically in order to save on computational time. Similar to the 2-dimensional model, the 3-dimensional model has an inlet at the upper left of the geometry, and an outlet at the upper right. The membrane is along the flat bottom of the geometry.

3.1.1.2 Governing Equations

Fluid velocity profiles were calculated using the Laminar Flow module. The convection and diffusive properties of the osmotic solution were simulated using Transport of Diluted Species module, where the convective terms were coupled with the Laminar Flow velocity profile. Assuming a constant density and viscosity, the system is modelled using Navier-Stokes equation:

$$\rho(\frac{\partial \underline{u}}{\partial t} + \underline{u} \cdot \nabla \underline{u}) = -\nabla p + \mu \nabla^2 \underline{\mathrm{u}} \quad (3.1)$$

where ρ is the density, $\underline{u}$ is the velocity, t is the time, p is the pressure, and μ is the dynamic viscosity. The convection and diffusive properties were governed by:

$$\underline{N_i} = -D_i \nabla c_i + \underline{u} c_i \quad (3.2)$$

$$\frac{\partial c_i}{\partial t} = \nabla \cdot \underline{N_i} - R_i \quad (3.3)$$

where $\underline{N}$ is the total flux of the species i, D is the diffusion coefficient (5.9 x 10^{-11} m^2 s^{-1})[7], c is the concentration of species i, and R_i is the reaction rate of species i.

3.1.1.3 Boundary Conditions and Initial Conditions

The walls of the system were set up as a no slip and no flux boundary condition. The inflow concentration is set to be consistent with the initial concentrations. The outlet of the system was set as an open boundary. The membrane was set to a normal inflow velocity equated as

$$\underline{u} = u_o \underline{n} \quad (3.4)$$

where $\underline{u}$ is the velocity, u_0 is the velocity entering the system, and $\underline{n}$ is the flux normal to the surface. The velocity entering the system is calculated based on a Kedem-Katchalsky model, where the water flux, J_v, at the membrane boundary is related as

$$J_V = L_P(\Delta P - \sigma \Delta \pi) \quad (3.5)$$

where L_P is the membrane hydraulic permeability, ΔP is the pressure difference across the membrane, σ is the reflection coefficient, and Δπ is the difference in osmotic pressure across the membrane. The hydraulic permeability is set at 1 x 10^{-7} m s^{-1} kPa^{-1}. The internal pressure on the membrane is determined by COMSOL, while the pressure and osmotic pressure on the exterior side of the membrane are both set to zero. The reflection coefficient is set to 1, completely reflective. The osmotic pressure for the interior side of the membrane is defined as

$$\pi = \pi_P + \pi_D \tag{3.7}$$

where π_P is the protein contribution to osmotic pressure and π_D is the Donnan effect's contribution to osmotic pressure. However, Donnan contributions can be ignored as they have been shown to be negligible at salt concentrations greater than 0.1 M [11]. The protein contribution to osmotic pressure calculated using the concentration of protein at the membrane boundary in conjunction with an exponential model fitted to osmotic pressure for BSA solution in aCSF, pH 7.4, (Figure 3.2) as shown below

$$\pi_P = exp^{(A_1 + A_2 c)} \tag{3.8}$$

where c is the concentration, and A_i are fitting parameters (-9.68 x 10^{-1} and 1.12 x 10^{-2}, respectively).

3.1.2. Results

The 2-dimensional simulation (workstation computational time courtesy of Dr. Dimitrios Morikis) with static conditions of 25 μL min^{-1} inlet flowrate and a protein concentration of 350 g L^{-1} showed a velocity profile (Figure 3.3: A) with significant convective flux near the membrane interface. Visualization of this region can be seen in the region of low protein concentration near the membrane (Figure 3.4: A). The average flux across the membrane was calculated to be 1.13 x 10^{-6} m s^{-1}. Converting this to flowrate by multiplying by the membrane surface area, gives a 61.0 μL h^{-1} flowrate, which is greater than the 7.2 μL of fluid expected to be needed to return tissue water content to uninjured levels. The transmembrane flux values correspond to 11.6 kPa osmotic pressure greater than the driving force pressure, and an effective protein concentration near the membrane of 124 g L^{-1}.

Three-dimensional analysis showed similar results with an average flux across the membrane of 9.9 x 10^{-7} m s^{-1}. Converted this flux equates to a 53.5 μL h^{-1} flowrate. The velocity

profile (Figure 3.3: B and C) and concentration profile (Figure 3.4 B and C) show dilution near the membrane. The effective concentration of protein near the membrane is 95 g L^{-1}, which correlates to an osmotic pressure of 7.6 kPa.

3.1.2.1 Parameter Optimization

Dependency of device extraction rate on protein concentration is shown in Figure 3.7. For 2-dimensions, 500 g L^{-1} showed an increased extraction rate of 34.7% (74.7 μL h^{-1}) over 350 g L^{-1}, while 3-dimensional analysis showed an increased extraction rate of 22.7% to 65.7 μL h^{-1}. Figure 3.7 also shows that 50 g L^{-1} is able to create an osmotic gradient over hydrostatic pressure with an extraction rate of 19.9 μL h^{-1} for 2-dimension and 17.9 μL h^{-1} for 3-dimensional.

Flowrate dependency is also analyzed for its effect on device extraction rate (Figure 3.8). For 2-dimensions, 500 μL h^{-1} showed an increased extraction rate of 48.9% (90.8 μL h^{-1}) over 25 μL h^{-1}, while 3-dimensional analysis showed an increased extraction rate of 75.7% to 94.1 μL h^{-1}. Additionally, 2.5 μL h^{-1} inlet flowrate shows a 2-dimension extraction rate of 38.1 μL h^{-1} and 3-dimensional extraction rate 29.0 μL h^{-1}.

3.1.4.2 2-D versus 3-D Comparison

The 2-dimensional model shows greater extraction rates than the 3-dimensional model in all cases except for inlet flowrates over 400 μL h^{-1}. This is expected as the volume ratio between the inlet and volume over the membrane is higher in the 2-dimensional case. A higher ratio leads to a faster protein replenishment rate near the membrane. The high concentrations lead to high osmotic pressures and greater extraction rates.

3.1.4.3 Mesh Analysis

The maximum mesh size, differential elements, for the 2-dimension simulation is 1 μm and contained 285973 degrees of freedom with 5444 internal degrees of freedom (Figure 3.1: B). The 3-dimensional simulation had a maximum mesh size of 37 μm, containing 474996 degrees of freedom with 40450 internal degrees of freedom (Figure 3.1: D).

The dependency of extraction rate on maximum mesh size is run to ensure that the calculated extraction rate is independent of mesh size (Figure 3.8). Mesh independence, less than 1% deviation of extraction rate, was determined to occur below 1 μm maximum mesh size in 2-dimensions, but was not achieved in 3-dimensions.

3.2. Quantitative Assessment of Edema in Spinal Cord Injury

One potential issue given the method of this treatment is that the device is not in direct contact with the injured tissue. Instead the device, via the hydrogel, is in contact with the dura that surrounds the spinal cord. Between the dura and spinal cord is the subarachnoid space, a column of cerebral spinal fluid. From Figure 3.9: A, it is estimated the radius of a rat spinal cord is 1.48 mm with the dura estimated to be located 1.62 mm away from the center of the spinal cord[8]. Until the injured spinal cord swells to reach the dura the device is going to be pulling fluid from the cerebral spinal fluid, surrounding the spinal cord, instead of the cord itself. The water content that allows for fluid extraction from the injured tissue is expected to be the minimum water content that the device is able to achieve given the current design and implementation.

This minimum water content that the device can achieve can be calculated. The following equation is evaluated to calculate the initial, uninjured volume, V_i,

$$V_i = \pi L R_i^2 \tag{3.9}$$

where L is the length and R_i is the initial radius of the spinal cord. The additional increase in volume, V_a, caused by swelling can be estimated by

$$V_a = \frac{4\pi}{3}\left(R_n^2 - R_i^2\right)^{1.5} \tag{3.10}$$

where R_n is the new swollen radius of the spinal cord. Given these two volumes and an initial water content, WC_i, the new water content, WC_n, can be calculated by

$$WC_n = WC_i * \frac{V_a + V_i}{V_i}. \tag{3.11}$$

Figure 3.10 shows the radii and water content values for a 5 mm segment of rat spinal cord, given an initial water content of 70%. The radius at which the spinal cord swells to reach the dura calculates to an estimated water content of 72.5%. This water content is expected to be the minimum achievable water content for *in-vivo* experiments in rats.

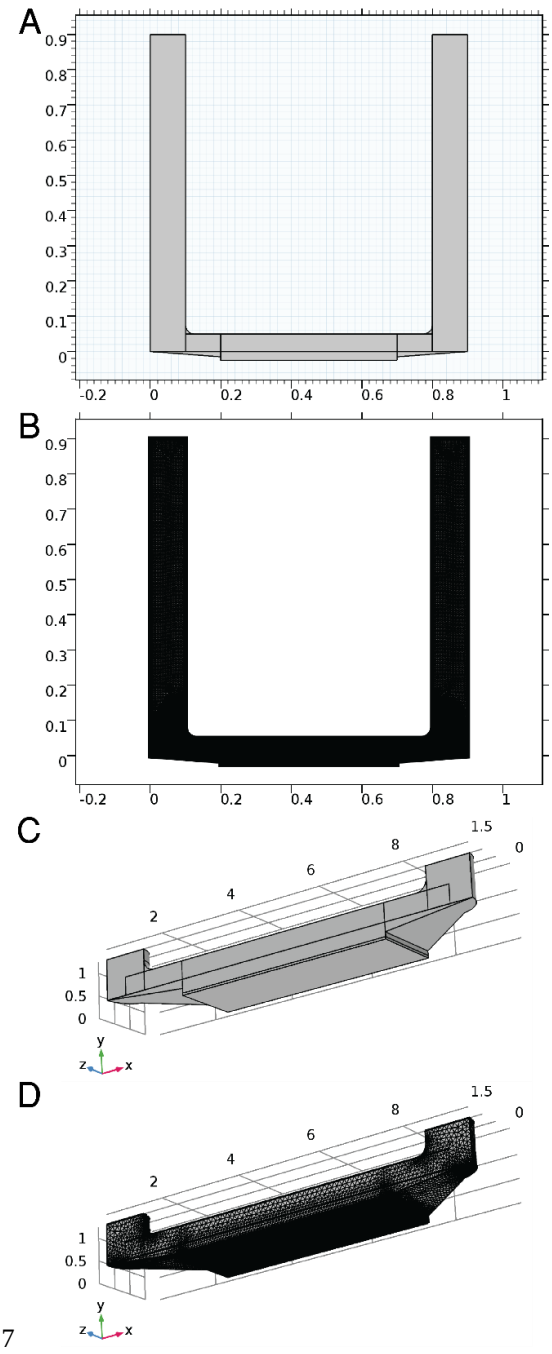


Figure 3.1: The 2-dimensional geometry (A) and the 3-dimensional geometry (C) of the osmotic transport device modelled in COMSOL. The inlet is in the upper left of the geometry and the outlet is in the upper right with the membrane positioned at the middle bottom of the geometry. Meshing for the 2-dimensional geometry is shown in (B) and 3-dimensional in (D).

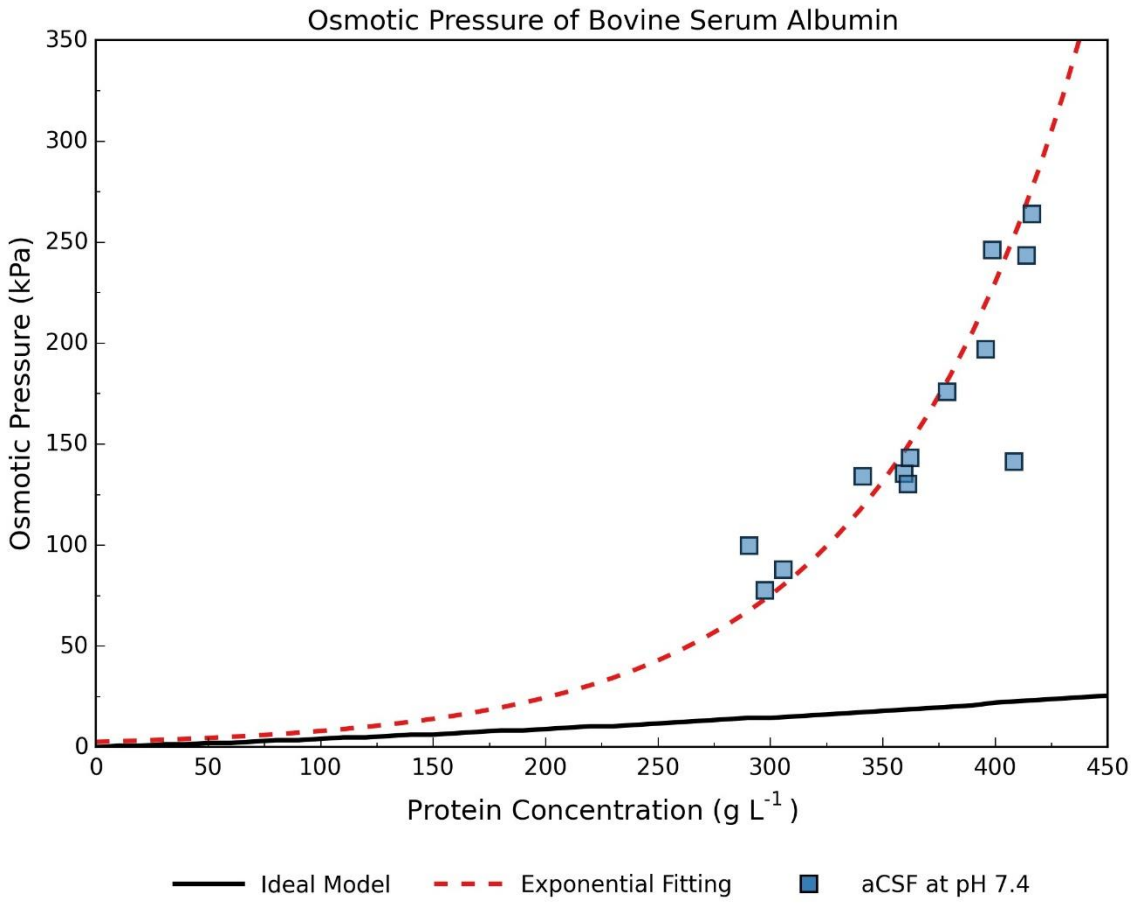


Figure 3.2: The osmotic pressure of bovine serum albumin at pH 7.4 in artificial cerebral spinal fluid. An ideal model is shown along with an exponential model fitted of the data.

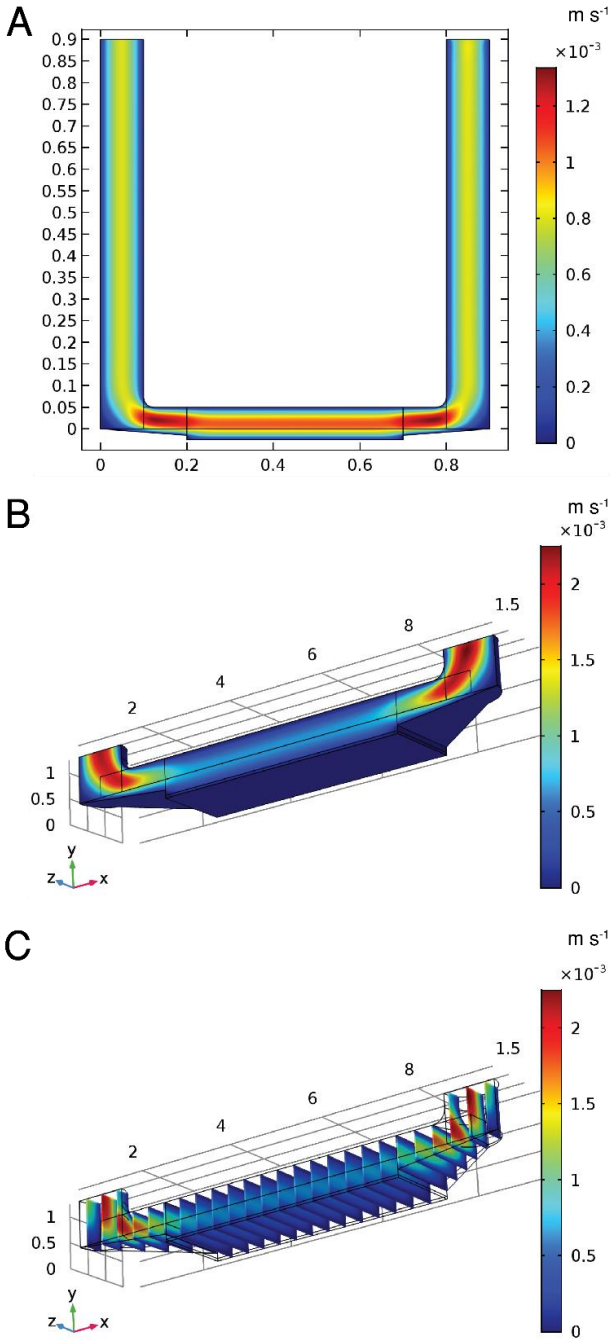


Figure 3.3: 2-dimensional (A) and 3-dimensional (B) velocity profile of the device given an inflow of 25 µL min^{-1} and a protein concentration of 350 g L^{-1}. (C) shows slices of the 3-dimensional velocity profile.

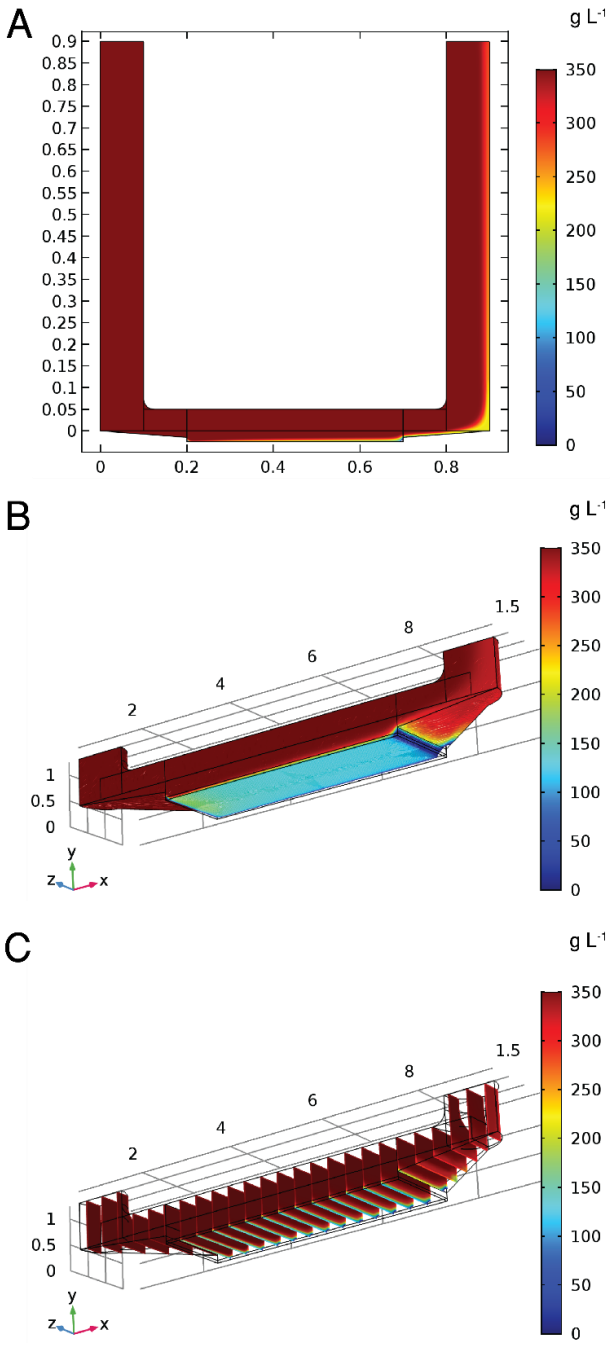


Figure 3.4: 2-dimensional (A) and 3-dimensional (B) concentration profile of the device given an inflow of 25 μL min^{-1} and a protein concentration of 350 g L^{-1}. (C) shows slices of the 3-dimensional concentration profile. Dilution is noticeable in regions near the membrane of the profiles.

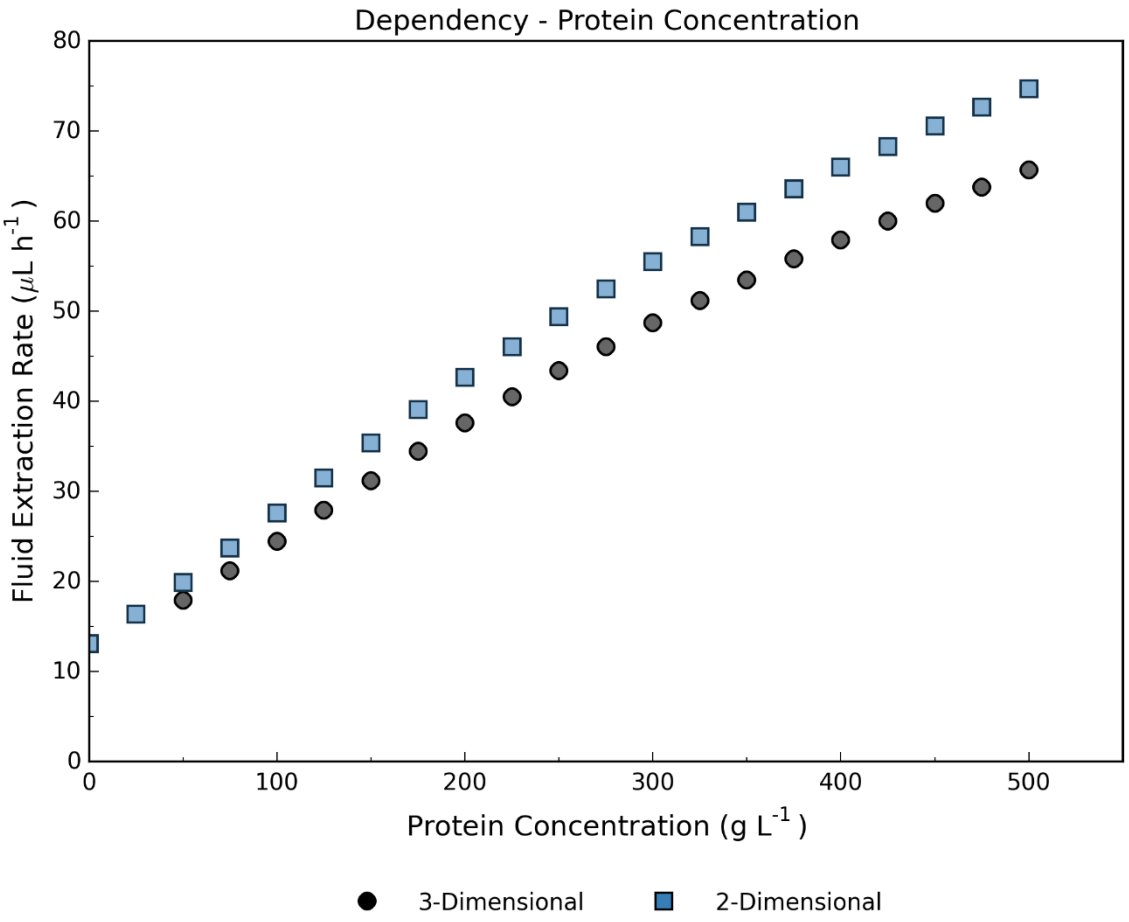


Figure 3.5: Computational analysis of fluid extraction rate dependency on protein concentration at an inlet flow rate 25 μL min^{-1}. The 2-dimensional model shows greater extraction rates than 3-dimensional model. This is expected as the ratio of the volume over the membrane, compared to the inlet and outlet, is greater in the 2-dimensional model than in the 3-dimensional model. This leads to a faster refresh rate of the protein near the membrane surface.

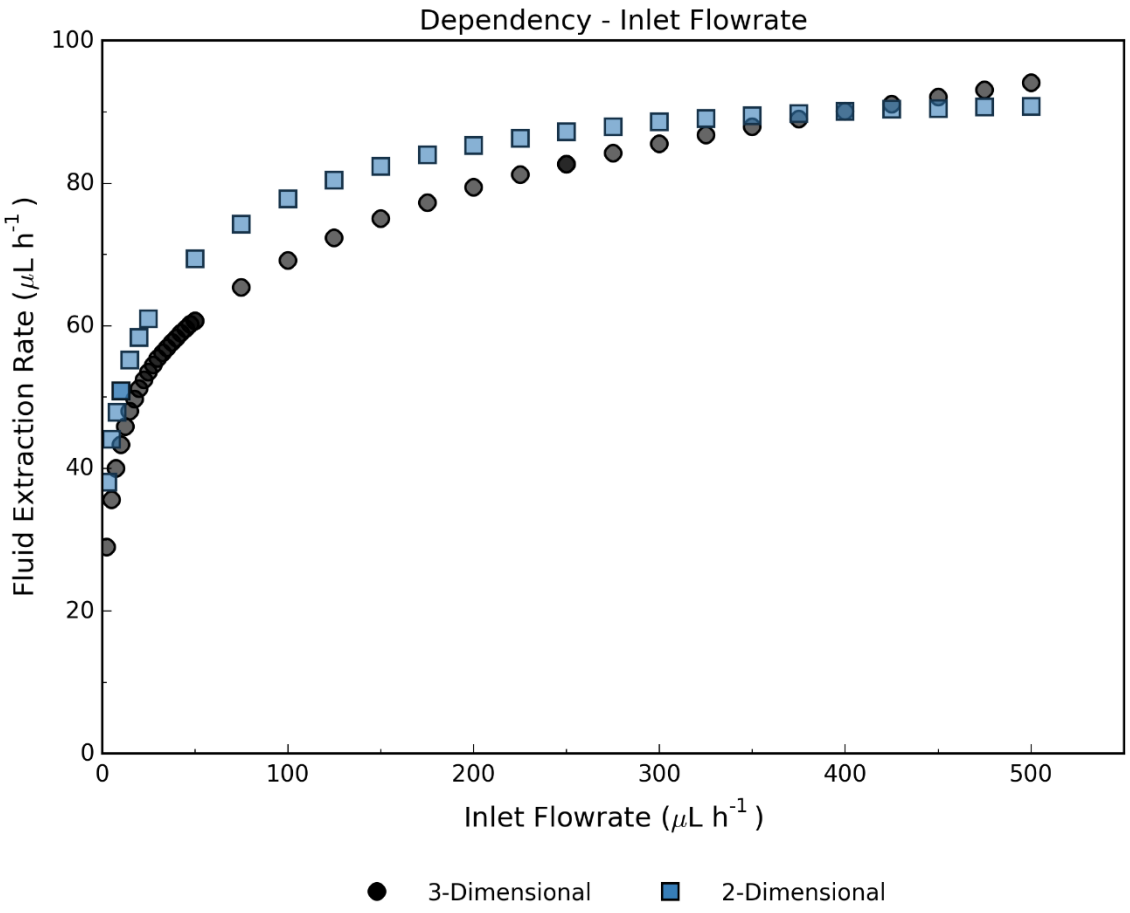


Figure 3.6: Computational analysis of fluid extraction rate dependency on inlet flowrate given protein concentration of 350 g L^{-1}. The 2-dimensional model shows greater extraction rates than 3-dimensional model. This is expected as the ratio of the volume over the membrane, compared to the inlet and outlet, is greater in the 2-dimensional model than in the 3-dimensional model. This leads to a faster refresh rate of the protein near the membrane surface.

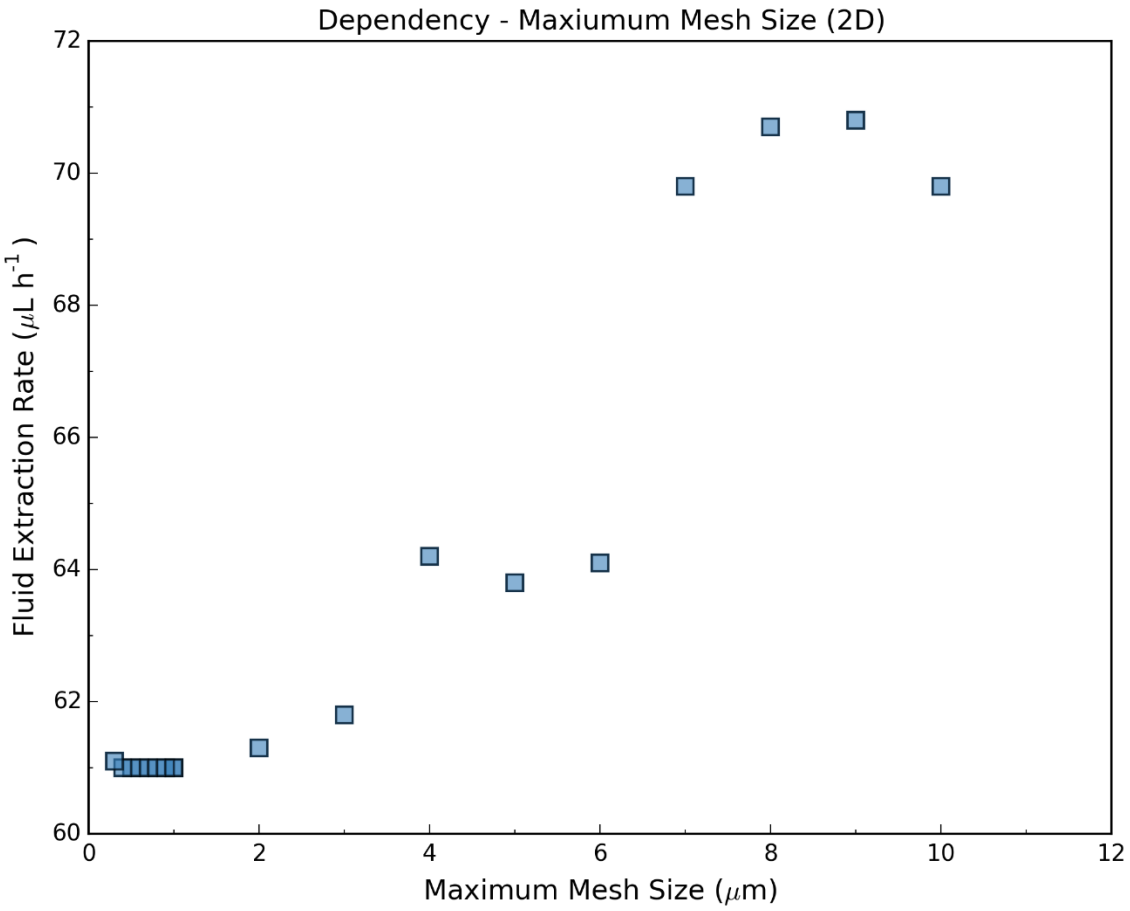


Figure 3.7: 2-dimensional boundary mesh independence at a concentration of 350 gL^{-1} and an inlet flowrate 25 µL min^{-1}. Mesh independence is reached at 1 µm.

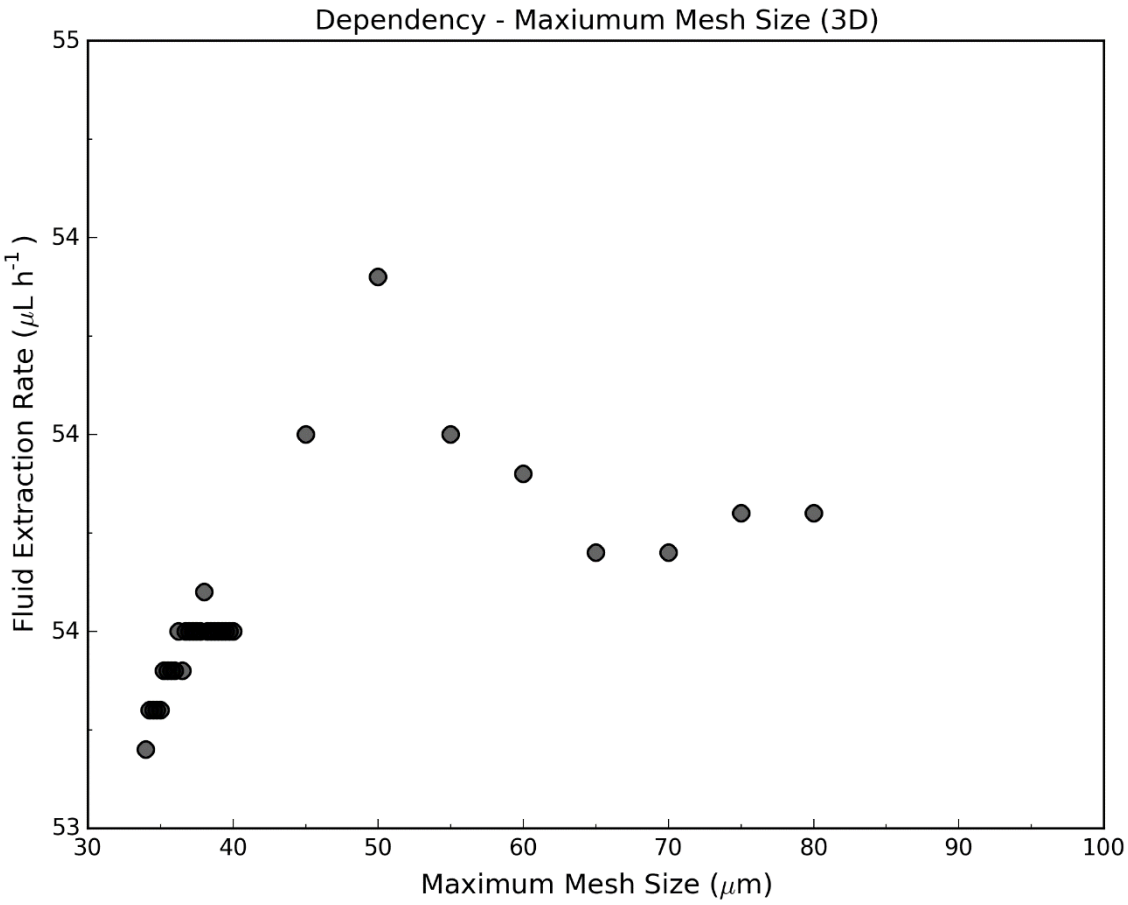


Figure 3.8: 3-dimensional boundary mesh at a concentration of 350 gL^{-1} and an inlet flowrate 25 μL min^{-1}.

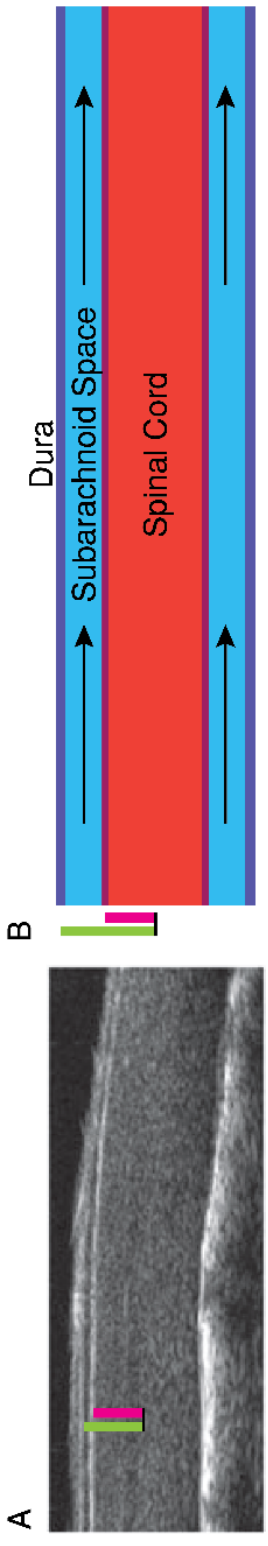


Figure 3.9: A) high resolution ultrasound image[8] of a rat spinal cord. B) drawing of a spinal cord emphasizing the subarachnoid space between the dura and spinal cord, that contains aCSF flowing caudally.

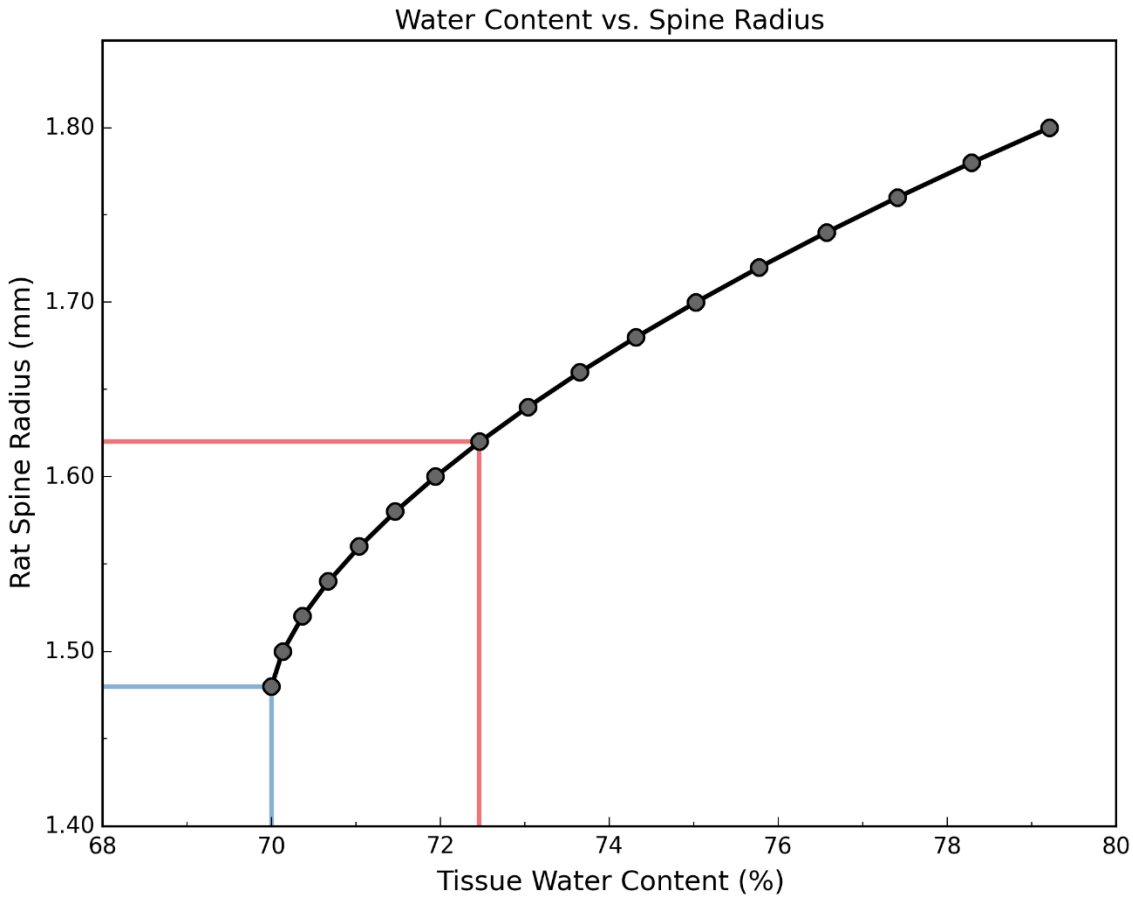


Figure 3.10: Calculation of the radius of a 1.5 mm segment of rat spinal cord given a tissue water content. The blue lines identify an uninjured spinal cord at 70% water content and 1.48 mm radius. The red lines identify the water content, 72.5%, where a segment of swollen spinal cord is in contact with the surrounding dura at 1.62mm.

CHAPTER 4.

DEVICE FABRICATION

4.1. Structure

Many biocompatible materials exist that can be harnessed for implants, however, only some of these materials are available to be fabricated by the required method, for this purpose. These limitations have a great effect on part design. For this application, titanium is utilized for the rigid internal structure and biomedical grade silicone for the flexible shell. Both of these materials have been shown to be biocompatible for extended periods of time.

4.1.1. Internal Structure

Fabrication of this device is a multistep process. The first step of this is the creation of the rigid solution chamber, which is done by the creating a design in SolidWorks, a CAD (Computer-Aided Drafting) program. The design next needs to be fabricated. Due to the small size and intricacy of the design, direct metal laser sintering, an additive process, similar to 3D printing, is employed.

4.1.2 Membrane

After the solution chamber is created, the membrane (NADIR® PM UP010, 10k Da MWCO, Polysulfone, Microdyn Nadir, Germany, Wiesbaden), needs to be attached, sealing off the solution chamber. The membrane is placed in the slotted area on the bottom of the chamber, so that the membrane support is facing away from the device. The membrane was chosen for its hydrophilic nature and its rejection of the osmotic agent.

4.1.3 Epoxy

Epoxy (EP42HT-2ND-2 MED Black, MasterBond, Hackensack, NJ) is then added around the edges of the membrane, to prevent fluid flow around the membrane. The epoxy is sets for 72 hours, at room temperature. By not curing the epoxy under additive heat, the membrane is allowed to dry without potentially altering the properties of the membrane. The epoxy was chosen for its bio-compatibility, and high viscosity, which allows for the epoxy to remain where it's placed and not to bleed into and potentially clog the membrane.

4.1.4 Outer Shell

The outer shell made of silicone (MED-4901, Nusil, Carpinteria, CA) is placed around a non-functional device and inside a mold in order to solidify into the correct shape (Figure 4.1). A non-functional device is used so that the membrane properties are not altered during curing, but still allows for the silicone to take the correct shape. The mold is made into three parts. The bottom part (Figure 4.1: C) keeps silicone from forming over the membrane area (Figure 4.1: (1)). The top section (Figure 4.1: A) has holes to allow for the ports to slide through them (Figure 4.1: (2)), as well as a small cylindrical projection down (Figure 4.1: (3)), toward the device, to ensure the device stays at the correct depth and orientation. By having a middle section (Figure 4.1: B), both the top and bottom sections of the mold can be smaller increasing the ease in which they can be taken off, and decreasing the stress that leads to tearing of the silicone shell. The top and bottom sections have protrusions toward the middle section (Figure 4.1: (4)), that overlap the middle section (Figure 4.1: (5)) and allow for the three sections to be aligned. Each section of the mold has protrusions extending out horizontally (Figure 4.1: (6)) to provide easier manipulation.

The mold is created in the same fashion and out of the same material as the solution chamber, CAD model to direct metal laser sintering of titanium. Once the silicone is poured into

the mold with a non-functional device. The silicone is desiccated for 12 hours to remove any trapped air pockets. After which time, the mold containing the silicone and device are place in an oven at 150°C for 1 hour. Following removal from the oven and cooling, the mold is separated, and the silicone shell is removed from the non-functional device before being placed around a functional device.

4.2. Protein Solution

To prepare a 350 g/L, pH 7.4, BSA solution, a 0.15 M aCSF, pH 7.4, aCSF solvent is used to dissolve a weighed amount of Bovine Serum Albumin, using a stir bar to facilitate mixing. The aCSF solution is prepared by dissolving the proper amount of salts in one liter of nanopure water. The solution pH was then measured by a pH Meter (ThermoScientific Orion 720A+, 13-641-253, Fisher Scientific, Hampton, NH) and adjusted using 1 M HCl (HX0603, EMD Millipore, Burlington, MA) and 1 M NaOH (S318, FisherScientific, Hampton, NH), while undergoing stirring to prevent local denaturation. The amount of acid and base used to adjust pH was considered part of the solution and is considered when determining concentration. The concentration of solutions was determined by dividing the amount of protein or salt by the volume of solvent used to make the solution. The volume of solvent considered the volume of protein or salt in the solution using the specific volume of the protein or the density of the salt.

4.3. Hydrogel

The hydrogel is made from 1% Ager, by weight, dissolved in aCSF solvent. The Agar/aCSF solution is placed in a container to achieve the proper gel height and then heated for 30 sec on high in a microwave. Agar was chosen due to its bio-compatibility and antibacterial properties.[9]

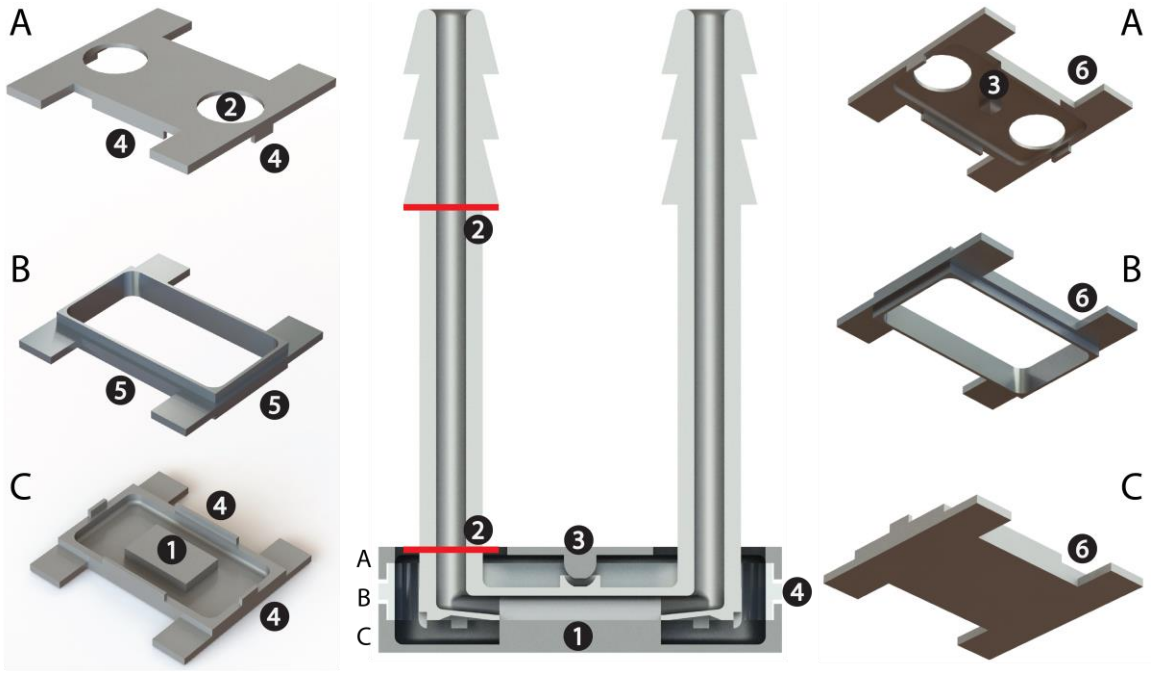


Figure 4.1: Schematic of the silicone mold: A) is the top of the mold, B) is the middle of the mold, and C) is the bottom of the mold. (1) shows where the bottom of the mold protrudes to prevent silicone from covering the area of membrane that fluid transverses. (2) is the gap in hole in the top of the mold that allows the device ports to pass through. (3) is where the top chamber protrudes down to hold the device in place within the mold. (4) identifies the ledges that help to align the bottom and top sections with the overlap in the middle section (5). (6) are the protrusions allowing for the mold to be held easily while being taken apart after curing.

CHAPTER 5.

PARAMETER ASSESSMENT

5.1. Introduction

Before testing the device *in vivo*, the osmotic transport device is tested *in-vitro* to ensure that the devices are functioning as expected. For this purpose, densimitry is employed to detect changes in solution density. These changes would be expected to occur if the device is effectively pulling fluid from outside the device into the device's solution chamber, making the solution less dense. Additionally, this method is able to evaluate, *ex-vivo*, the hydraulic resistances of the tissue.

5.2. Methods

Prior to *in-vitro* testing, each device has its membrane hydraulic permeability determined. Testing is done by placing a device in a pre-weighed test tube, with a reservoir of water connected to one port. A second line to waste is attached to the other device port, with a shut off valve part way down the line. Also, connected to the water reservoir is a pressure line that can pressurize the reservoir, and force fluid into the line, toward the device. Once the pressure has driven fluid past both ports, of the device, the shot off valve is closed leaving only the device membrane as an escape path. Measuring the mass of fluid that accumulates in the test tube over a certain time and pressure allows for calculation of the device membrane's hydraulic permeability.

After hydraulic permeability has been determined, the devices have their extraction rates tested via densimitry. Two densimiters (DMA 35, Anton Paar, Ashland, VA) and two pressure sensors (26PC, Honeywell, Morris Plains, NJ) are place before and after the device. Additionally, solenoids (7889K63, McMaster-Carr Supply Co, Elmhurst, IL), connected to each other, are placed closer to the device than the pressure sensors or densimiters. At set time points, these solenoids

active to change the flow path, either through the device (Figures 5.2: B) or to bypass the device (Figures 5.2: C). By allowing the device to be bypassed, undiluted solution is able to reach the sensor and densimiter after the device. In this way, a new baseline can be established before each parameter change is made.

The device is placed in a beaker of artificial cerebral spinal tissue (aCSF), while protein solution is driven through the lines by syringe pumps (Harvard Aparatus 11 Plus Syringe Pump, Harvard Aparatus, Holliston, Massachusetts), data is acquired by custom LabVIEW code (LabVIEW, National Instruments, Austin, TX) code, and analyzed by custom python code that employs a low-pass filter (Figures 5.2 and 5.3). The signal density differences are corrected by averaging the baseline before and after each parameter change.

For measuring the effect device parameters have on the extraction rate, solution samples were collected after the second densimiter and weighed. Additionally, the samples were diluted by with aCSF and their spectrum measured at 280 nm. The absorbance values were then used to determine the extraction rate of the device. These mass and absorbance calculations for extraction rate were compared to the values determined by densimitry.

For the measurement of tissue resistances, rat spinal cord tissue samples with and without dura (courtesy of Dr. Binder's lab (Carrie Jonak)) were glued (VetBond, 3M, St. Paul, MN) onto the silicon shell of the device. For each test set of tissue both samples, with and without dura, came from the same animal, and were tested with the same device.

5.3. Device Fluid Extraction

In order to determine the rate of fluid extracted by the device, a conversion from density difference needs to be made. By using the following equation for density difference, $(D_1 - D_2)$, a volume extracted, V_{S2}, can be determined

$$(D_{P1} - D_{P2}) = \frac{\frac{V_P}{SV} + V_{S1} * D_S}{V_P + V_{S1}} - \frac{\frac{V_P}{SV} + D_S(V_{S1} + V_{S2})}{V_P + V_{S1} + V_{S2}}, \quad (5.1)$$

where V_P is the volume of the protein, SV is the specific volume of the protein, V_{S1} is the volume of solvent initially in solution, and D_S is the density of the solvent. Solving for the volume extracted gives

$$V_{S2} = \frac{(D_{P1} - D_{P2})\left({V_P}^2 + 2V_P V_{S1} + {V_{S1}}^2\right)}{V_P\left((D_{P1} - D_{P2}) - \left(\frac{1}{SV} - SD\right)\right) + V_{S1}(D_{P1} - D_{P2})}. \quad (5.2)$$

By plugging in

$$V_P = \frac{Concentration * Flowrate * SV}{1000}, \text{ and} \quad (5.3)$$

$$V_{S1} = Flowrate - V_P \quad (5.4)$$

for the volume of protein and volume of initial solvent, respectively, the volume extracted will be converted to an extraction rate.

5.3.1. Flowrate Dependency

Densimitry analysis of the osmotic transport device shows a non-linear relationship between extraction rate and inlet flowrate (Table 5.1). The data shows a positive relationship between flowrate and extraction rate. The average extraction rate, after normalization, for the lowest flowrate, 25 µL min^{-1}, examined was calculated to be 33.7 µL h^{-1}. The highest flowrate, 500 µL min^{-1}, tested had an extraction rate of 97.3 µL h^{-1}.

5.4. Hydraulic Resistance of Device and Tissue Components

Table 5.2 relates the resistances to flow from the osmotic transport device. Beginning at a normalized, average extraction rate of 68.8 µL h^{-1} for an unencumbered device, the extraction rate falls drastically when forced to draw fluid through spinal tissue and hydrogel. The resulting 24.6

μL h^{-1} is a 64% reduction in extraction rate. The dura surrounding the tissue, caused an additional 11% reduction over tissue and hydrogel alone, to an extraction rate of 17.1 μL h^{-1}.

The resistance to water flux for the different components were found by solving for the resistances in the Kedem-Katchalsky model,

$$J_V = \frac{\Delta P - \sigma \Delta \pi}{\mu (R_M + R_{HT} + R_D)} \tag{5.5}$$

where μ is the solvent viscosity, R_M is the membrane resistance, R_{HT} is the combined hydrogel and tissue resistance, and R_D is the flux resistance from the dura. The results (Table 5.2) show the main contributor to flux resistance is the device membrane with a resistance of 1.69 x 10^{-13} m^{-1}. Physiologically, the dura had the greatest resistance at 1.38 x 10^{-13} m^{-1} resistance, compared to 0.94 x 10^{-13} m^{-1} from the tissue and hydrogel.

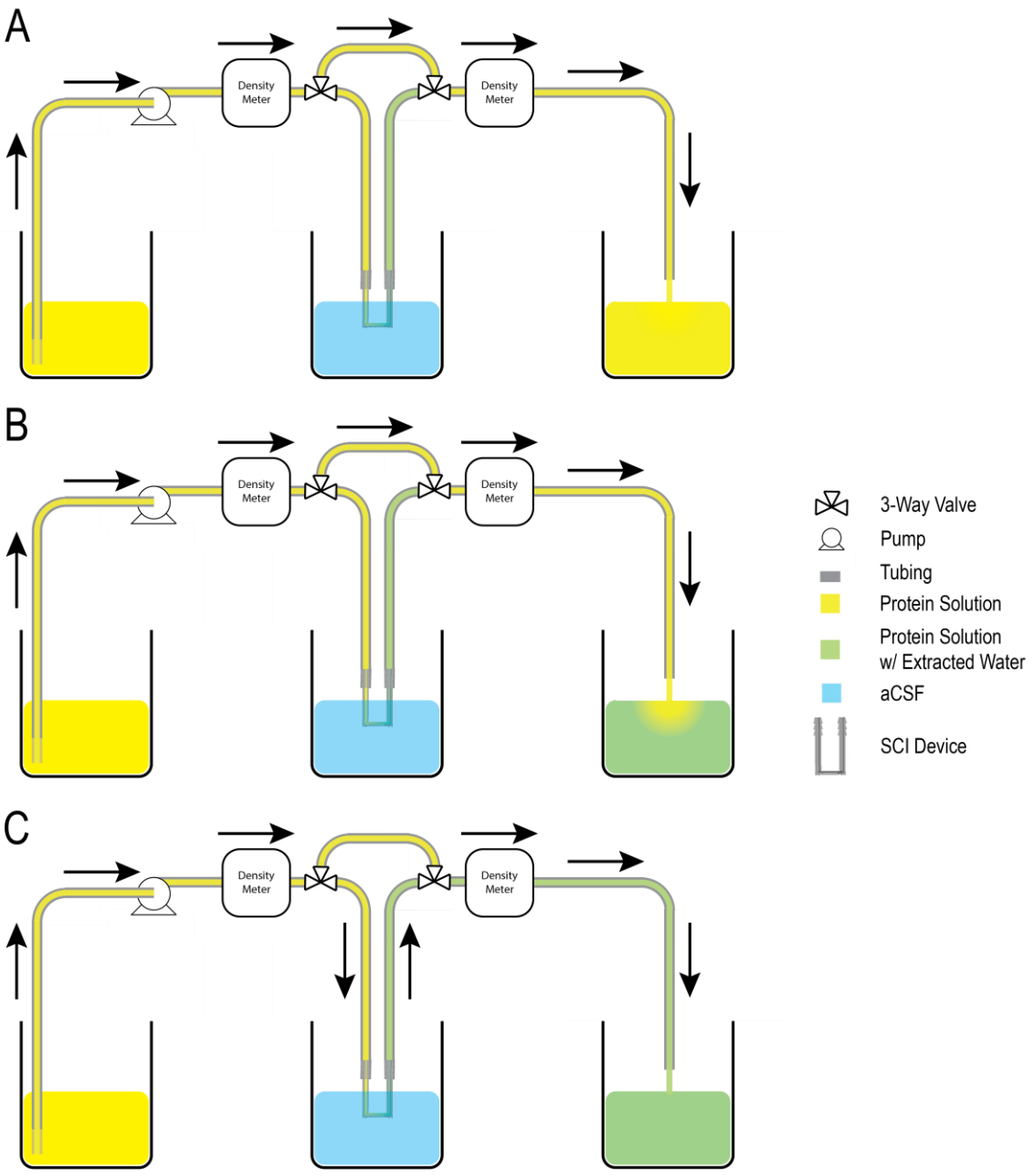


Figure 5.1: Two densimiters, one before and one after the osmotic transport device, measure the density of the protein solution moving through the lines at different flowrates. The difference between these two measurements are calculated for determination of extraction rate. A) shows the flow path at the beginning of the experiment that establishes a baseline reading, B) shows the flow path for measuring the device extraction rate, while C) shows the flow path that measures a new baseline before measuring a new device extraction rate.

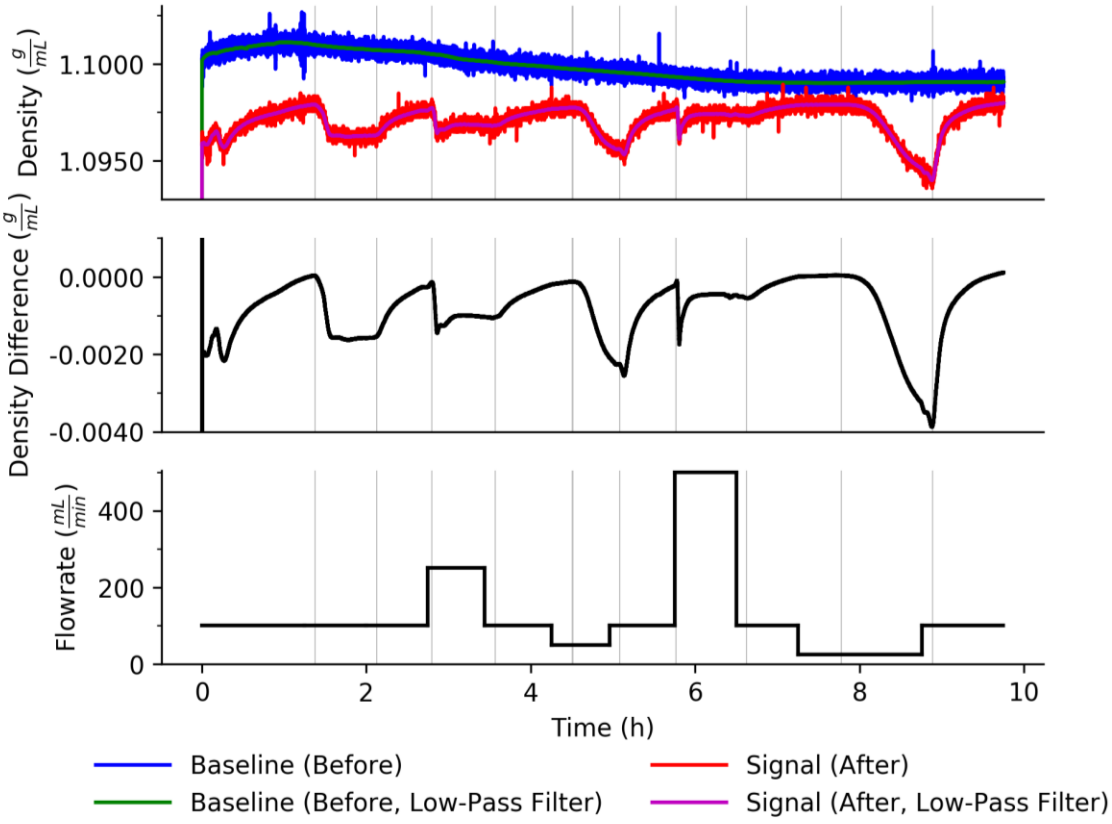


Figure 5.2: Example of densimitry data from a flowrate dependency experiment. The two densimiters, one before and one after the osmotic transport device, measure the density of the protein solution moving through the lines at different flowrates. The difference between these two measurements are calculated for determination of extraction rate. The grey lines indicate the data locations that were used to determine the extraction rate at each flowrate.

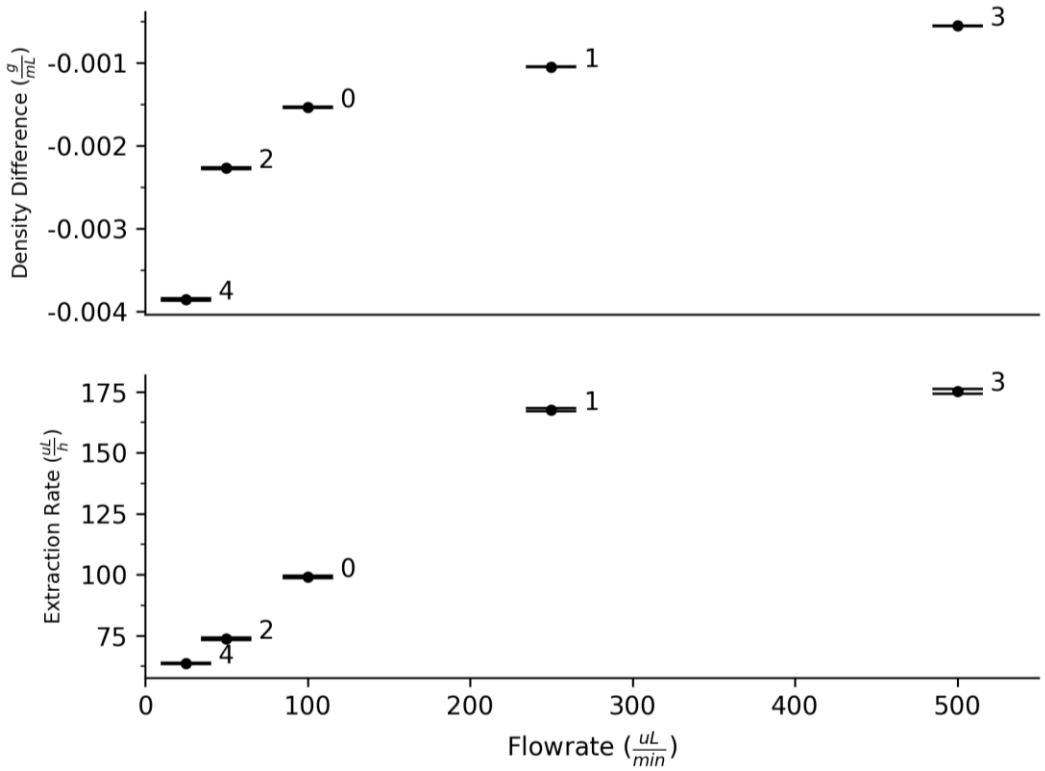


Figure 5.3: Example densimitry data from a flowrate dependency experiment. The extraction rate is a calculation made from the density difference between the densimiter before and after the osmotic transport device considering the flowrate at the time of measurement. The number next to the data point is the order in which that data was obtained.

Table 5.1: Osmotic Transport Device Extraction Rates (µL h^{-1}) at Various Flowrates (µL min^{-1}) and Concentrations

Trial	Hydraulic Permeability (m/s/kPa):	Hydraulic Permeability Error (m/s/kPa):	Flowrate (µL min^{-1}) 25	50	100	250	500
Trial 1	1.4E-07	4.5E-09	52	67.7	101.8	160.3	156
Trial 2	2.1E-07	5.1E-08	63.7	73.7	99	167.6	175.2
Average Rates (µL h^{-1})			**57.9**	**70.7**	**100.4**	**164.0**	**165.6**
Error Rates (µL h^{-1})			5.9	3.0	1.4	3.6	9.6
Trial 1 (Normalized)	1E-07		36.5	47.6	71.5	112.6	109.6
Trial 2 (Normalized)	1E-07		30.9	35.8	48.1	81.4	85.0
Average Rates (µL h^{-1})			**33.7**	**41.7**	**59.8**	**97.0**	**97.3**
Error Rates (µL h^{-1})			2.8	5.9	11.7	15.6	12.3

Table 5.2: Osmotic Transport Device Extraction Rates and Resistances at 350 gL^{-1} and 100 µL min^{-1} Flowrate

Trial	Hydraulic Permeability (m/s/kPa):	Hydraulic Permeability Error (m/s/kPa):	Device	Device + Hydrogel + Tissue	Device + Hydrogel + Tissue + Dura
Trial 1	1.9E-07	6.0E-09	128.7	46.0	32.0

	Resistivity (m^{-1} x 10^{-13})		
	Device	Hydrogel + Tissue	Dura
Trial 1	68.8	24.6	17.1

CHAPTER 6.

in-vivo MODEL

6.1. Treatment

The mathematical modeling and in vitro work have provided insight into the device behavior during clinical application, however, the next step toward clinical validity is determining what effect the device has on reducing edema following spinal cord injury *in vivo*.

A functional osmotic transport device (OTD) would be expected to reduce tissue water content below injured levels. While hydrogel only application, without an osmotic driving force, would not be expected to reduce water content levels, and has the potential to increase the water content level due to the greater water content level causing a gradient into the injured tissue. Additionally, it is possible, although not measured, that the functional OTD and hydrogel with non-functional OTD corrects metabolite abnormalities in the injured tissue.

6.2. Methods

SCI experiments begin with a Sprague Dawley female rat, between 8-10 weeks old, being anesthetized (ketamine/xylazine cocktail (80mg/kg)) before a midline incision is made along the spine. The muscles are then separated to expose the bone before a T8 laminectomy is performed. After the spinal cord has been exposed, a 250 kD contusion is performed (Infinite Horizons Impactor, Precision Systems and Instrumentation, LLC, Fairfax Station, VA) and the T7 and T9 dorsal processes are removed and flattened. The OTD is then placed on top of the injury site and the muscles are sutured together (chromic gut) before the skin is stapled and sutured (silk sutures) around the device. Following site closure and 1-hour post injury, if applicable, the device is connected to 350 g L^{-1} BSA solution in aCSF at pH 7.4 and an 11.5 inch head height flowrate. A

peristaltic pump moves protein solution to the gravity feed location and maintains head height (Cole Palmer, Master Flex L/S, EW-07524-50). A second flowrate setup was employed at 25 μL min^{-1} via a syringe pump (Standard Infusion Only PHD ULTRA™ Syringe Pumps, Harvard Apparatus Holliston, MA).

At the conclusion of treatment the protein solution is removed from the device, if applicable, and the animal is euthanized by Fatal Plus and cardiac puncture. After expiration, a 5 mm segment of the spinal cord tissue is removed at the epicenter of the contusion site. The tissue is weighed before and after processed (85°C for 48 h) to determine tissue water content.

6.3. Results

Uninjured water content at the epicenter averaged 68.3 ± 0.04% (Figure 6.1). Injury (SCI) caused an increase in water content to 73.3 ± 0.28%. SCI treated by hydrogel without a functioning OTD (SCI + Hydrogel (0.3%) did not significantly differ from the injured, untreated case at 73.3 ± 0.17%. The two treatment cases with a functional OTD (SCI + OTD (0.3%) and SCI + OTD (1%)) both showed decreased water content compared to the untreated case. The 0.3% hydrogel with head height flowrate had water content values of 72.4 ± 0.38% compared to 71.7 ± 0.45% of the 25 μL min^{-1} flowrate via the syringe pump group.

6.4. Conclusion

The osmotic transport device is able to significantly reduce tissue water content in injured spinal cord after 3 hours of treatment. Application of hydrogel with a non-functioning device did not have a significant effect on the water content. Both OTD treatment groups are in the range of the calculated minimum water content achievable (Chapter 3.2).

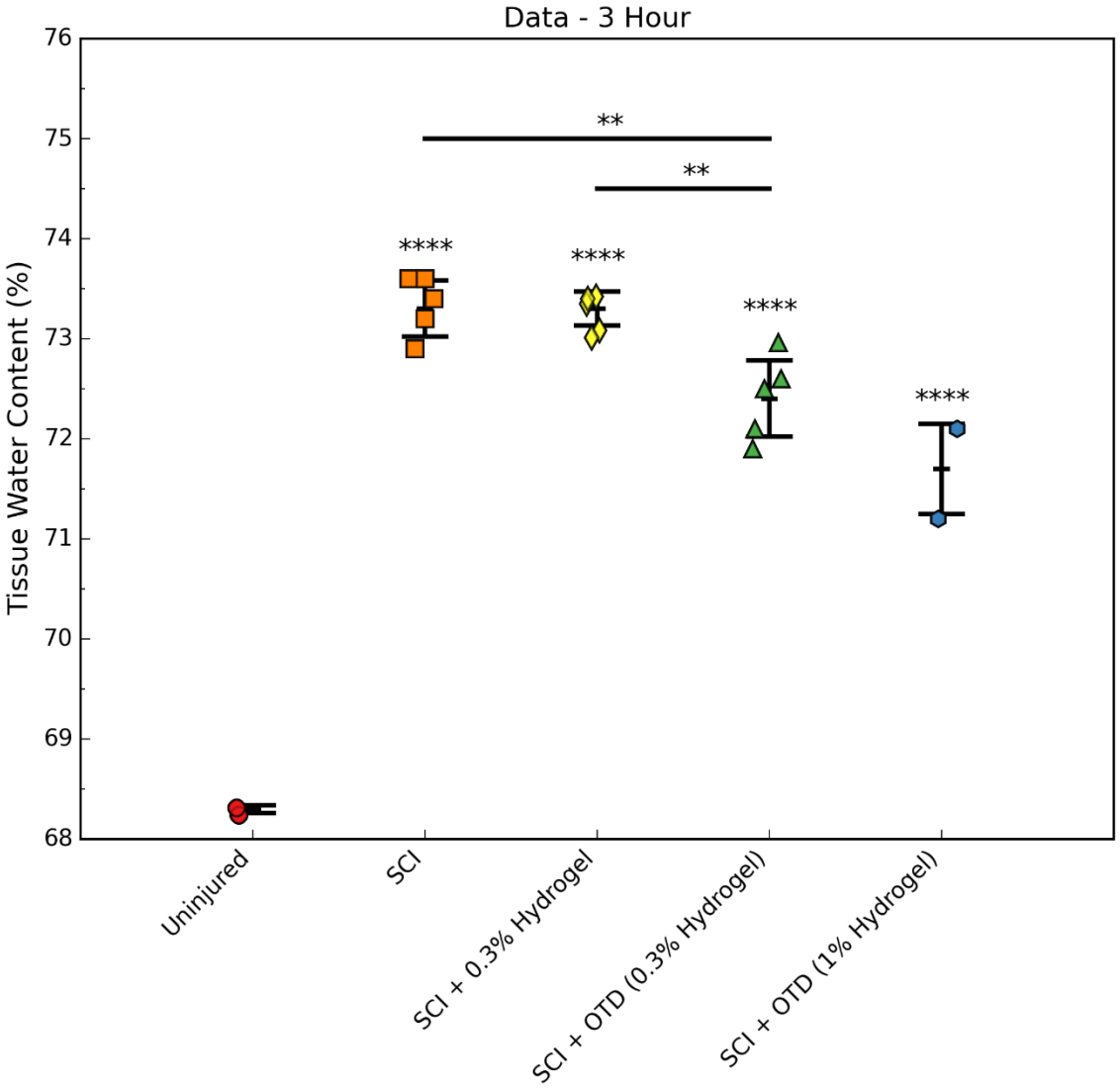


Figure 6.1: 3 hour *in-vivo* data for severe spinal cord injury in rats. All injury cases are significantly different from the uninjured case. There is no significant difference between the injured, untreated case and the injured case treated with hydrogel alone. Both the injured, untreated and injured, treated with hydrogel have significantly higher water content compared to the injured, treated with a function osmotic transport device (OTD) with 0.3% hydrogel.

CHAPTER 7.

CLINICAL ASSESSMENT

7.1. Introduction

Modelling is beneficial for quickly and freely manipulating parameters of a system. However, comparisons need to be made between computation and *in-vitro* experiments to determining the validity of the model. With the accuracy and validity of a model depends on how closely that model compares to real world data.

7.2. Comparison Modeling to Experimental

Finite Element Analysis (Chapter 3) calculated 25 μL min^{-1} flowrate to have an extraction rate of 61.0 μL h^{-1} flowrate. Densimitry experiments, after normalizing for the higher membrane hydraulic permeability, showed an average extraction rate of 33.7 μL h^{-1}. This value and the densimitry data at 50 and 100 μL min^{-1} show lower than expected extraction rates. Flowrates of 250 and 500 μL min^{-1} had extraction rates of 97.0 and 97.3 μL h^{-1}, which is higher than the 87.2 and 90.8 μL h^{-1} expected. However, those rates are within the densimitry error. Moreover, the curves of the data sets are reasonably similar and look to peak at near the same value (Figure 7.1).

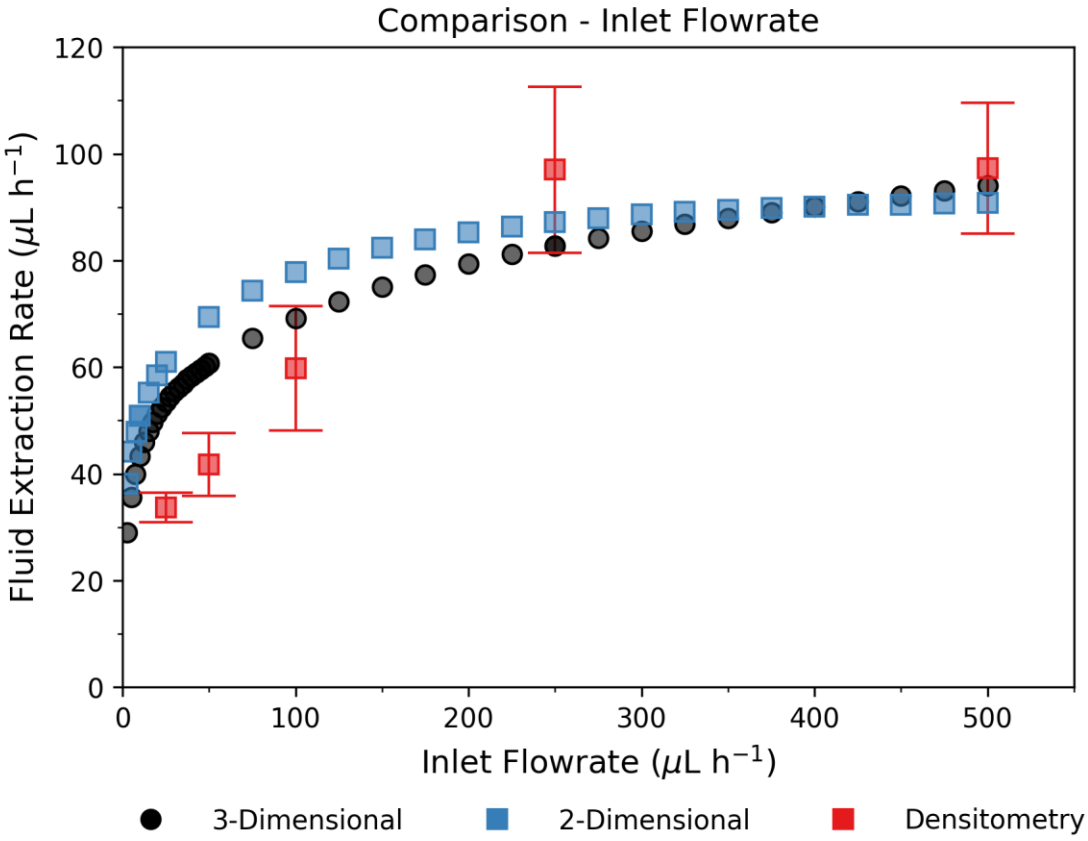


Figure 7.1: Comparison between finite element analysis and *in-vitro* data obtained via densimitry. Densimitry data at flowrates of 100 µL min^{-1} and below show a lower than expected extraction rates. Flowrates at 250 and 500 µL min^{-1} show higher than expected extraction rates but contain those expected extraction rate within their error.

CHAPTER 8.

CONCLUSIONS

8.1. Findings in This Work

Herein, an osmotic transport device was designed for the reduction of edema in spinal cord injury, in which many different limitations and issues need to be addressed including chemical, physiological, and manufacturing. This design was analyzed by a finite element method to determine device parameter effects on extraction rate, and shown, with later *in-vivo* parameters, to have an extraction rate of 61 μL h^{-1}. While this rate was almost an order of magnitude greater than the 7.2 μL needed to be removed, calculations of physiological restrictions on device effectiveness showed the lowest tissue water content that the device could achieve was 72.5%.

Fabrication of the device was made with materials that would be biocompatible for the time course of edema. Following fabrication, the extraction rate of the device was analyzed via a densimitry method. Although, the data curve did not match exactly to the finite element analysis, evaluation showed a 58 μL h^{-1} extraction rate for later *in-vivo* parameters. Finally, testing was done *in-vivo* to determine the device's effect on the water content of an injured spinal cord and showed a greater than 1% reduction in water content in the tissue.

8.2. Future Directions

The next steps for the development of the osmotic transport device for spinal cord injury is testing at longer time points to determine if these reductions in water content (Chapter 6) can be held for the time course of edema. Following that, functional testing of locomotor ability will tell if the reduction in edema has benefited the neurological outcome of the animal.

Once small animal testing has concluded a larger device can be fabricated for human application. Testing of the larger device can be done similarly to the rat version of the device (Chapters 3 and 5). Viability assessment of a 4x4 cm device (Table 8.1), shows an expected 240 µL of fluid that needs to be removed. Extraction rates at the levels seen in this work (Chapter 6) would cause the device to take 12 hours to reduce the water content to 2% above uninjured levels (Chapter 3.2.).

Table 8.1: Viability Assessment of OTD Treatment for SCI Injury in Humans

	Mouse Brain	**Rat Spine**	**Human Spine**
Tissue Mass (g)	0.4[6]	0.7[6]	35[6]
Tissue Injured (%)	50%[4]	17%*	17%[+]
Reduction in Water Content (%)	0.75%[4]	1%	4%[+]
Water Mass to Remove (µL)	1.5	**1.2**	**238**
Treatment Time (h)	2[4]	**3**	**12**

* Data from Dr. Devin Binder's Lab (Jennifer Yonan)

[+] Assumed values are the same as rat spinal cord values

PART 2.

FURTHER UNDERSTANDING OF THE PHYSICAL PHENOMENA OF CROWDED PROTEIN OSMOTIC PRESSURE

CHAPTER 9.

DEVELOPMENT AND APPLICATIONS OF A CONCENTRATING OSMOMETER

9.1. Introduction

Due to its deviation from ideality, concentrated osmotic pressure has fascinated scientists for over a hundred years. Studying near-saturation osmotic pressure has been proven useful with many studies showing that multiple protein properties can be determined by examining proteins in this concentrated region.[10–12] However, the literature is limited due to the quantity of protein required to study the complete osmotic pressure-concentration profile.

A single concentration's osmotic pressure data point, in current osmometers, requires approximately 7 mL of solution.[12,13] Therefore, in order to obtain an entire osmotic pressure-concentration profile, upwards of a hundred grams of the protein is required. For some proteins this is possible, as they are both commercially available and relatively inexpensive. For the majority of proteins that remain, and are either not commercially available or expensive to obtain, obtaining concentrated osmotic pressure data is not feasible.

While the quantity of protein required is often the limiting factor when studying proteins, most of the development in osmometry has been to increase the quantity of measurements that can be obtained in a given time, rather than decreasing the quantity of protein required.[14,15] However, there has been some efforts to allow for a single sample to give multiple data points. One of these methods uses dilution and calculated mass balances to obtain multiple data points from a single sample.[16]

Another approach uses nitrogen to pressurize the protein solution to different concentrations.[17] Unfortunately, this method is not suitable for high concentrations as it requires a spectral analysis of the protein solution at each pressure point. This is not practical, as in addition

to osmotic pressure being non-linear with increased concentration, the viscosity also has a non-linear relationship to concentration. Therefore, after a certain concentration, the protein solution viscosity would prevent the solution from being removed from the chamber via a syringe. The method also uses a capillary head height to determine the steady state of the system. When the pressure in the solution chamber is increased a concentration polarization occurs near the membrane. This concentration polarization causes an elevated concentration of protein to reside next to membrane, causing solvent flow to behave relative to this higher concentration near the membrane, rather than the lower bulk protein concentration. For high concentrations, this would cause difficulty in determining when steady state occurred as concentration polarization can take many hours to dissipate. Removing a sample from the chamber, before concentration polarization dissipated, would give a lower solution concentration than the actual average concentration in the chamber.

Here, we have designed and developed a concentrating osmometer allowing for four orders of magnitude reduction in protein required for measuring osmotic pressure-concentration curve (0.1 g of protein required, assuming a starting concentration of 50 gL^{-1} and initial volume of 2 mL). This is accomplished by beginning with a smaller initial solution volume and after the osmotic pressure reaches steady state, decreasing the volume, and then repeating these steps to obtain additional data points.

This method also allows for a decrease in the time required to obtain a complete osmotic pressure-concentration curve. A conventional osmometer requires that protein solutions be made for each desired concentration, time for a solution to reach steady state in the osmometer, then that solution to be removed, and the osmometer cleaned between experiments. By working continuously with one initial solution and concentrating that solution repeatedly, the time to obtain a profile can decrease from a one to three week time frame, to a multiday time frame. These decreases in protein

required and time for profile acquisition will allow for researchers to study the osmotic pressure of their proteins of interest.

9.2. Designing a concentrating osmometer

9.2.1. Function

The concentrating osmometer works by decreasing the solute chamber volume (Figure 9.1-A, yellow volume) to obtain new solute concentrations (Figure 9.1-B, yellow volume). By rotating the plunger (Figure 9.1-B, pink arrows) the volume in the top chamber (Figure 9.1-B, yellow volume) is reduced, driving transmittable species into the solvent chamber (Figure 9.1-B, blue volume). The difference in plunger height (Figure 9.1, ΔH) between before rotation (Figure 9.1-A, H_{P_1}) and the plunger height after rotation (Figure 9.1-B, H_{P_2}), is equivalent to the change in chamber height (Figure 9.1, ΔH) before (Figure 9.1-A, H_{C_1}) and after rotation (Figure 9.1-B, H_{C_2}). Knowing the dimensions of the solute chamber and the change in chamber height, the change in solute concentration can be calculated by

$$C_1V_1 = C_2V_2, \tag{9.1}$$

where C_1 is the concentration before rotation, V_1 and V_2 are the volumes before and after rotation, and C_2 is the new concentration.

9.2.2. Membrane

An osmometer (Figure 9.2) is created by separating two solutions with a semi-permeable membrane (Figure 9.1 (9)). One of the solutions needs to have a membrane-rejected species in order to generate an osmotic pressure. Selection of the membrane is important to ensure that only desired species are able to transverse the membrane. This is achieved by selecting a membrane pore size that is large enough to freely allow transmittable species through, but small enough to prevent

the restricted species from passing through. The membrane's chemical structure and indeed all surfaces that come in contact with the species are selected to not affect the solutions.

Solute Chamber

The solute chamber (Figure 9.2 (7)) is a straight cylinder allowing for the maximum available membrane surface area, combined with the smallest volume. The solute solution is added directly into the solute chamber allowing for visual inspection and potential removal of any air pockets that may have developed during addition. Once enough solute has been added to form a meniscus above the solute chamber O-ring (Figure 9.2 (8)), the membrane is laid on top of the O-ring in such a way as to prevent air from being trapped in the chamber.

9.2.3. Plunger

In order to change the volume in the solute chamber, a plunger (Figure 9.2 (1-6)) is driven toward the membrane, increasing the pressure in the chamber and forcing transmittable species through the membrane leaving restricted species in a smaller, concentrated, volume.

The plunger is made up of two main sections, the lower (Figure 9.2 (2-6)) and upper components (Figure 9.2 (1)). The plunger bottom (Figure 9.2 (5)) contains the pressure transducer (Figure 9.2 (3)), the most important part of the plunger. To begin with, the pressure transducer provides the data for the experiment and therefore it is essential to get a transducer with the correct pressure range. Too small of a range will be vulnerable to bursting, due to the increased pressure during plunger concentration movements. Too large of a range will lose sensitivity and may lead to useless data. The size of the pressure transducer is also important as it is the limiting factor for the size of the plunger and therefore the solute chamber volume and indeed the entire device.

The pressure transducer needs to seal against the plunger to make sure there are no leaks. To achieve sealing an O-ring (Figure 9.2 (4)) is placed around the sensing side of the transducer

and the inside of the lower plunger. The transducer is secured against the O-ring with enough force to seal the chamber by a threaded cylindrical lock (Figure 9.2 (2)). The lock needs to be screwed into the lower plunger to drive the transducer into the O-ring. The lock has a hole through the center with a slot cut from the center to the edge. This slot allows for the lock to slide over the transducer wires for ease of assembly. Due to the confined area the slot is also the mechanism for screwing the lock in. The slot allows for a flathead screw driver to slide through and then rotate, lowering the lock toward the transducer.

The lower plunger also needs to seal against the walls of the solute chamber. Another O-ring (Figure 9.2 (6)), around the diameter of the lower plunger, is used to ensure that solution does not escape around the plunger during concentrating and at elevated osmotic pressures. This O-ring is located as close to the bottom of the lower plunger as possible, while still allowing for most of the diameter to be enveloped on the top and bottom. The envelopment is important to support the O-ring when the plunger moves. Without the bottom support the O-ring would slide off when the plunger moved up. The same would be true if the top support was removed and the plunger moved down.

The upper component of the plunger threads into the solute chamber allowing the plunger to move up and down. The size of the threads connecting the upper plunger to the solute chamber is important, as the more threads per inch, the greater the plunger's vertical sensitivity is to upper plunger rotation. However, the increase in threads also decreases the strength of the threads. The thread strength needs to be great enough to ensure that they do not strip and potentially lock the plunger in the solute chamber.

The plunger top and bottom are also connected by threading together, but with threads of the opposite direction of those connecting the solute chamber and the plunger top. These connections having opposing thread and a thread gap in the plunger connection, allow the two parts

of the plunger to rotate independently of each other. This thread gap is after the connecting thread on both the plunger bottom and top, so that when the plunger is rotated to lower the plunger assembly deeper into the solute chamber, the threading between the plunger bottom and top are not in contact and so will prevent them from unthreading. Conversely, when the plunger assembly is unscrewed the threads of the plunger top and bottom come in contact but do not unscrew as the plunger top's rotation is in the opposite direction of the connecting threads. By constructing the plunger top and bottom in this way the plunger assembly is only able to become disassembled outside of the osmometer.

The plunger top is slotted similarly to the transducer lock to allow the transducer wires to slide through for easy assembly. The slot has the added benefit of being useful as a measurement of rotation around the solute chamber. Given the plunger threading, initial location, and degree of rotation, the concentration of the solute solution can be determined by change in solute chamber height. In order to rotate the plunger, increasing or decreasing the solute chamber volume, the top of the upper plunger is notched allowing for attachment of a wrench.

9.2.4. Solvent Chamber

The solvent chamber (Figure 9.2 (12)) holds solution containing all transmittable species. In order to ensure that the transmission of species from the solvent to solute chamber does not significantly affect the concentration of that species on the solvent chamber, the quantity of solvent in the solvent chamber needs to be far greater than the quantity of solute solution. Instead of containing the large volume of solvent solution within the osmometer, the solvent chamber has a minimized volume, but uses ports to pump solution past the membrane from a container of desired volume. Using a pump to deliver solvent to the chamber has the added benefit of moving the solvent

past the membrane by convention and does not rely on diffusion to maintain a well-mixed condition.

9.2.5. Reinforced Membrane Support

The osmotic pressure, in addition to the pressure increase when contracting the solute solution, has the potential to bow the membrane toward the solvent chamber, increasing the solute chamber volume, and throwing off the resulting concentration. To minimize this effect, a porous support (Figure 9.2 (10)) is employed on the solvent side of the membrane. The pores allow the transmission of species but do not allow the membrane to bow significantly enough to effect concentration. The membrane support needs to be sealed around its diameter as well as on the solvent chamber side to prevent solution leaks. Sealing around the diameter can be achieved using non-reactive epoxy, while a gasket (Figure 9.1 (11)) can seal the membrane support to the solvent chamber.

9.2.6. Assembly

The solution and solvent chambers need to be screwed together in order to provide the pressure required to seal and secure the gaskets, between the solvent chamber, membrane support, membrane, and the solute chamber. In order to keep the membrane support and the gasket in line with the membrane and solvent chamber, the solvent chamber has a cutout for the gaskets and membrane support to rest, while the solute chamber has an extrusion of a slightly smaller diameter to ensure correct alignment and desired sealing pressure.

Practical considerations

Concentrating the solute chamber can have multiple issues associated with it that should be considered. The first is bursting, either the membrane or the pressure transducer. If the pressure

in the solute chamber grows greater than what the membrane can handle then a hole will develop that allows solute to pass through. The hole may clog and seal itself off, but while it's open, the pressure and solute will leave the solute chamber. This will lead to lower and inaccurate concentration pressure readings. Bursting the pressure transducer is easier to diagnose, as the signal will fail to make sense. While being easier to diagnose, bursting the pressure transducer is far more expensive than bursting the membrane.

Another consideration is concentration polarization, which will occur while concentrating the solute. This concentrating osmometer utilizes an opposing threading design that allows for the plunger to be backed out of the solute chamber, increasing the chamber volume. By backing the plunger out even slightly, the increased chamber volume pulls solvent through the membrane clearing the concentration polarization and potentially clearing any solute clogging membrane pores and hindering transport.

9.2.7. Concentrating Factor

This design has a solute chamber with a maximum volume of 2.6 mL and a minimum volume of 140 µL. A total of 4.8 rotations changes the chamber volume from maximum to minimum values. The height change of each rotation is 1.6 mm and there is a total change in the plunger tip height of 7.6 mm. This allows the solution of interest to be concentrated by a factor of 18.87.

Even given 18x concentrating factor shown in this design iteration, the concentrating osmometer opens up a large amount of previously unavailable proteins that can now be studied in their concentrated region. Where before upwards of 100g of protein was needed to gather a complete osmotic pressure profile, using the concentrating osmometer only needs 55 mg to create 2.6 mL of 21.2 gL^{-1} BSA solution that can be increased to a concentration of 400 gL^{-1}.

Although this design is used herein to validate a concentrating osmometer, alterations can be made to the design to further reduce the minimum and maximum volumes to increase the concentrating factor to over 100x.

9.3. Osmotic Pressure Theory

Many models have been developed for relating the osmotic pressure of a solution at a given concentration to physical parameters. An example of which is the van't Hoff equation, which is used to determine the molecular weight of the solute. While many models only require dilute osmotic pressure data, it is the osmotic pressure of near-saturation concentrations which are the most notable, due to its deviation from ideality. Various models have been developed to explain the phenomena which cause the observed non-idealities in the osmotic pressure as a function of increasing solute concentration, however, the vast majority of the models require fitted parameters since a variant of the virial expansion based on McMillan-Mayer theory is used.[18] One model which does not rely on fitted parameters, and furthermore requires only independently measureable and physically realistic parameters is the free-solvent based model.[12]

9.3.1. Free Solvent Based Model

The free-solvent based model accounts for the hydration and ion binding (when applicable) of the solute. The free-solvent based model has been previously examined for single protein solutions[19], binary protein solutions[20], and sucrose solutions.

Full mathematical development of the free-solvent based model is described elsewhere.[11,12] Briefly, for a two chamber osmometer separated by a semi-permeable membrane in which there are n distinct species, where p proteins (or other rejected solutes) are fully rejected and confined to

chamber II and the remaining species (n - p) are diffusible, the free-solvent based model describes the osmotic pressure, π, as

$$\pi \approx \frac{RT}{\overline{V}_1} \ln\left(\frac{\left(\sum_{i=1}^{n} N_i^{\mathrm{II}} - \sum_{i=1;i\neq 2\rightarrow p+1}^{n} \sum_{j=2}^{p+1} \nu_{ij} N_j^{\mathrm{II}} - \sum_{j=p+2}^{n} \nu_{1j} N_j^{\mathrm{II}}\right)\left(N_1^{\mathrm{I}} - \sum_{j=p+2}^{n} \nu_{1j} N_j^{\mathrm{II}}\right)}{\left(N_1^{\mathrm{II}} - \sum_{j=2}^{n} \nu_{1j} N_j^{\mathrm{II}}\right)\left(\sum_{i=1}^{n} N_i^{\mathrm{I}} - \sum_{j=p+2}^{n} \nu_{1j} N_j^{\mathrm{II}}\right)} \right), \quad (9.2)$$

where N_i^k is the number of moles of species i in compartment k, and ν_{ij} is the net number of moles of species i interacting with species j. The compartment containing the protein solution is denoted as superscript II, while the non-protein compartment is denoted as superscript I.

9.4. Experimental Protocol

Solvent solutions are prepared by dissolving the designated mass of NaCl (No.S9888, Sigma-Aldrich, St. Louis, MO) into one liter of ultrapure water (EASYpure RoDi D13321, Thermo Scientific Barnstead Water System, Thermo Fisher Scientific, Waltham, MA) to produce a 0.15 M solution. This solvent is then used to dissolve a weighed mass of Bovine Serum Albumin (No. A30075, BSA, Research Products International, Mt Prospect, IL), using a stir bar to facilitate mixing. The solution pH of both solutions is measured by a pH Meter (Model 13-641-253, ThermoScientific Orion 720A+, Thermo Fisher Scientific, Waltham, MA) and adjusted, under stirring, using 1 M HCl (No. HX0603, Millipore Sigma, Burlington, MA) and 1 M NaOH (No. S318, Thermo Fisher Scientific, Waltham, MA), to be within 0.05 pH of the desired value. Stirring also allows for the prevention of local denaturation in the protein solutions. The amount of acid and base used to adjust pH is considered part of the solutions and is accounted for when calculating concentrations. The concentration of the solutions was determined by dividing the amount of protein or salt by the volume of solvent used to make the solution. The volume of solvent includes

volume of protein or salt in the solution using the specific volume of the protein or the density of the salt.

The osmotic pressure experiment is setup by filling the top chamber with a measured mass of protein solution, at the desired concentration, until a meniscus is formed above the chamber walls. Any air bubbles present are removed from the solution. A membrane (NADIR® PM UP010, 10k Da MWCO, Polysulfone, Microdyn Nadir, Germany, Wiesbaden), which had been soaked in ultrapure water for at least one hour and is then placed on top of the protein chamber, ensuring no air pocket forms. The protein chamber is sealed and excess solution is expelled. The expelled solution is collected and weighed to allow for an accurate mass of protein to be determined. Next, a membrane support is placed on the opposite side of the membrane to prevent bowing deformation caused by the increased osmotic pressure. A rubber gasket is then placed on the other side of the membrane housing to seal the solvent chamber from leaks.

The osmometer assembly is then screwed together and connected to a beaker of solvent, open to atmosphere. Solvent is then circulated through the solvent chamber using a peristaltic pump (Model EW-07524-50, Master Flex L/S, Cole Palmer, Vernon Hills, IL). Pressure reading are obtained using a pressure transducer (EW-68001-04, Cole-Palmer, Vernon Hills, IL) that is digitally recorded through data acquisition (Model NI SCC-68, National Instruments, Austin, TX). Once the pressure reading stabilized, the pressure and plunger height were recorded. The plunger was then rotated to reduce the top chamber volume and increasing the protein. When the pressure stabilized again, the pressure and plunger height were again recorded and the process was repeated until the desired concentration-pressure curve was obtained.

The parameters for hydration, ν_{12}, and ion binding, ν_{32}, are determined by nonlinear regression of Eq. 2 (TableCurve 2D, Systat Software, San Jose, CA) to best fit the osmotic pressure versus concentration profiles for each trial in the study.

9.5. Results

The osmotic pressure for BSA solutions, in 0.15M NaCl at pH 7.4, 25°C, are shown in Table 9.1 as measured by the concentrating osmometer. Concentrated osmotic pressure for BSA has been previously studied4 and was used for a comparison between the concentrating osmometer and a conventional osmometer (Table 9.1 – Literature). As regression weighs the highest concentrations and pressures the heaviest, only literature concentrations up to concentrations reached using the concentrating osmometer were used. This allows the data sets to be compared to each other in an accurate manner.

Data points for the concentrating osmometer as well as literature data and their corresponding FSB model fittings can be seen in Figure 9.3. The majority of the trials, with the exception of trial 2, are located near their literature counterparts, if slightly higher. Trial 2's osmotic pressure values, at and above the 325 gL^{-1} concentration show significantly elevated values over literature. The literature and aggregate model fittings are nearly identical at concentrations below 300 gL^{-1}. Above that concentration the two fittings begin to diverge with the concentrating osmometer fitting having a higher pressure profile than the literature values. These differences in pressure with the concentrating osmometer data having greater pressures, could be due to a dilution that occurs from high pressures causing membrane bowing during literature data collection. Membrane bowing increases the volume of the protein chamber, allowing solvent to enter into the protein chamber and thereby dilute the protein solution. So the actual concentration may be lower than the recorded initial concentration.

The concentration, pressure curves were then regressed upon using the FSB model (Eq. 2) to determine hydration and ion binding values for BSA (Table 9.2). Comparison of individual trials varied in magnitude relative to the regression of the literature data. The hydration and ion binding

compare similarly between the trials and regression of the literature data. Trials 1 and 5 were almost identical to literature values, while the rest of the trials, with except of trial 3, had overlapping error with literature regressed values. Trial 3's regression values were far below the other trails and literature values.

9.6. Conclusion

Here, a concentrating osmometer was developed and tested. The practical issues and applications were discussed. The concentrating osmometer was tested for concentrating a BSA solution, obtained the osmotic pressure-concentration profile for BSA, and compared to conventional osmometers. There was good agreement between the regressed hydration and ion binding values provided by the FSB model regression.

The concentrating osmometer reduces the required quantity of protein to decrease three to four orders of magnitude and reduces the time from three weeks to three days. Using such an osmometer will now allow for the osmotic pressure of solutions which were previously not feasible to be studied. Furthermore, the use of a concentrating osmometer has uses in chemical, physical, and biological research.

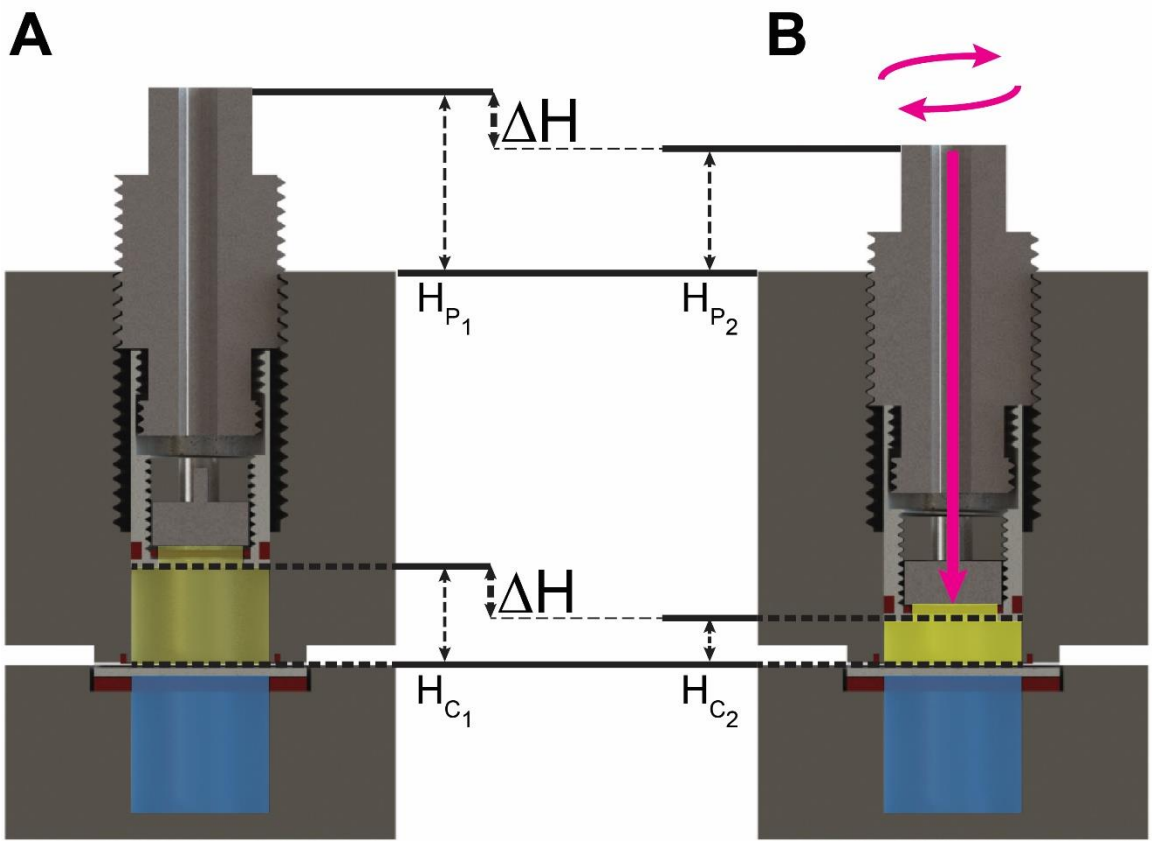


Figure 9.1: A collapsed view of the concentrating osmometer before (A) and after rotation (B). Rotation is donated by curved pink arrows in B), by which the protein chamber volume, denoted in yellow is decreased (pink arrow aimed down). The solvent previously in the protein solution before the rotation (A) is driven into the solvent chamber denoted in blue. The change in plunger height, ΔH, from its initial height, H_{P_1},to its new height, H_{P_2}, is equivalent to the change in chamber height, ΔH, from the initial height, H_{C_1}, to its end height, H_{C_2}. By knowing the change in chamber height and the dimensions of the chamber, the change in solute concentration can be calculated.

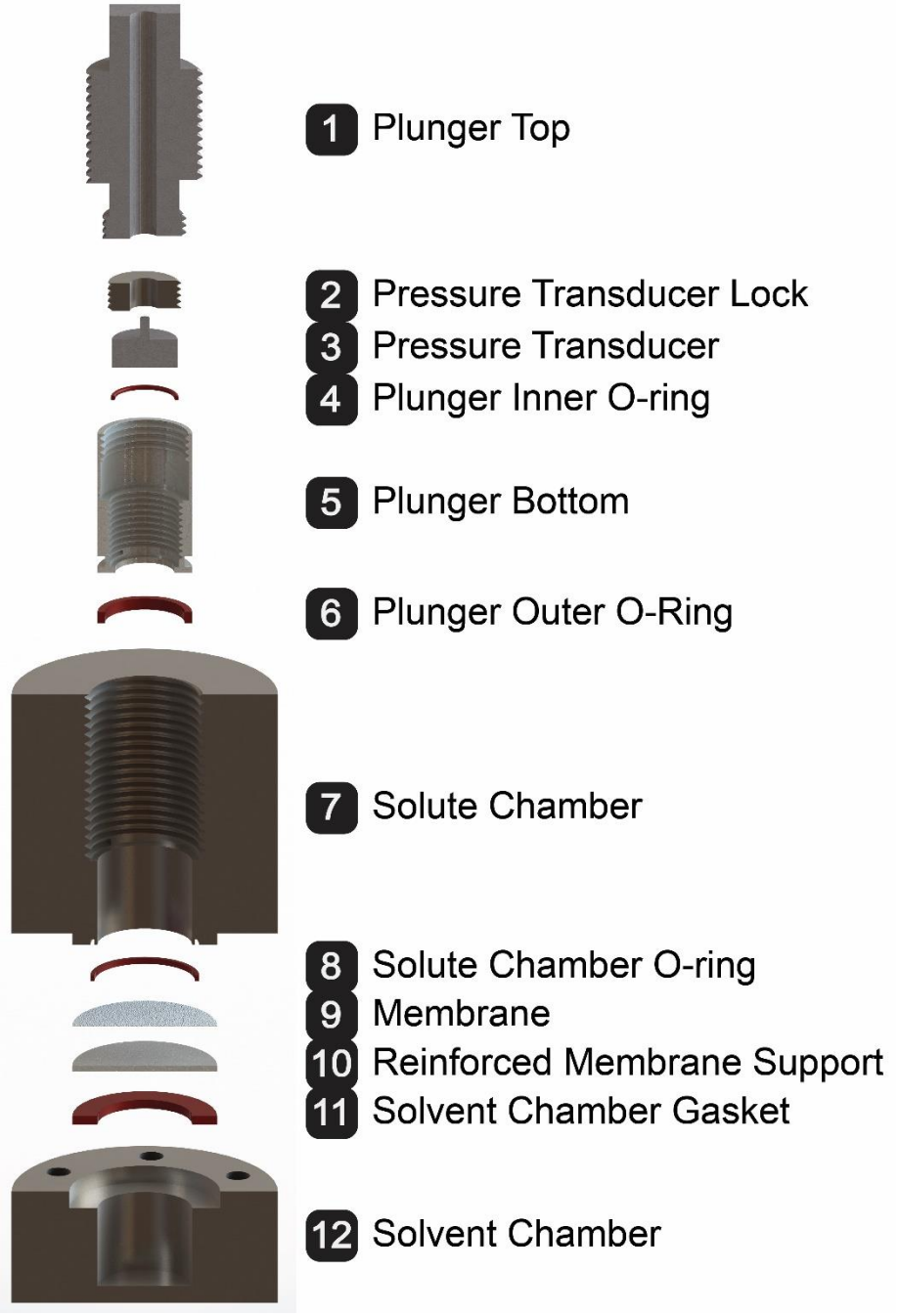


Figure 9.2: Exploded, sectioned view of all the components in the concentrating osmometer. Components 1-6 represent the plunger that changes the solute volume and its O-rings (red – 4, 6) that seal off the solute chamber from going around the plunger. Component 7 and 12 are the solute and solvent chambers, respectively, with components 8-11 containing the components to allow transport between the chambers (9) and ensuring the 2 chambers are otherwise sealed (red - 8, 11).

Table 9.1: Osmotic Pressure of BSA in 0.15M NaCl, pH 7.4, 25°C

Literature*		Trial 1		Trial 2		Trial 3		Trial 4		Trial 5		Trial 6	
[BSA] (g L^{-1} Soln)	Osmotic Pressure (kPa)	[BSA] (g L^{-1} Soln)	Osmotic Pressure (kPa)	[BSA] (g L^{-1} Soln)	Osmotic Pressure (kPa)	[BSA] (g L^{-1} Soln)	Osmotic Pressure (kPa)	[BSA] (g L^{-1} Soln)	Osmotic Pressure (kPa)	[BSA] (g L^{-1} Soln)	Osmotic Pressure (kPa)	[BSA] (g L^{-1} Soln)	Osmotic Pressure (kPa)
84	6.4	312	120.4	220	57.0	191	49.1	246	54.7	310	108.2	324	143.3
91	7.9	317	128.3	227	69.1	201	53.0	250	60.5	316	120.7	328	170.8
211	44.3	324	139.7	268	93.6	207	56.7	252	67.3	321	115.8	333	165.9
211	44.5	332	149.7	301	113.3	224	61.2	255	72.9	328	128.9	335	189.1
289	112.5	335	159.3	305	131.9	228	70.1	259	77.3	330	142.0	339	175.6
325	132.8			325	168.1	253	81.5	262	82.5	335	153.1	345	193.3
325	132.8			325	184.6	255	91.9	269	91.9	337	174.4	350	214.0
354	189.7			330	206.3	271	102.5	273	97.9	343	188.9	353	187.6
357	218.4			332	212.5	278	106.7	288	102.3	347	206.8	356	218.1
				337	235.9	285	113.0	294	107.8	359	216.5		
				342	239.1	299	115.2	294	114.8	363	215.8		
				345	258.0	305	127.9	296	122.4	361	224.1		
				340	296.5	311	137.9	309	135.1	365	233.0		
				354	302.9	345	145.7	312	142.4	367	244.1		
				353	330.2	349	158.7	309	151.6	372	259.2		
						352	184.5	321	160.4				
						369	196.9	323	167.0				
						369	205.2						
						373	212.1						

*Data from Vilker et al.[13]

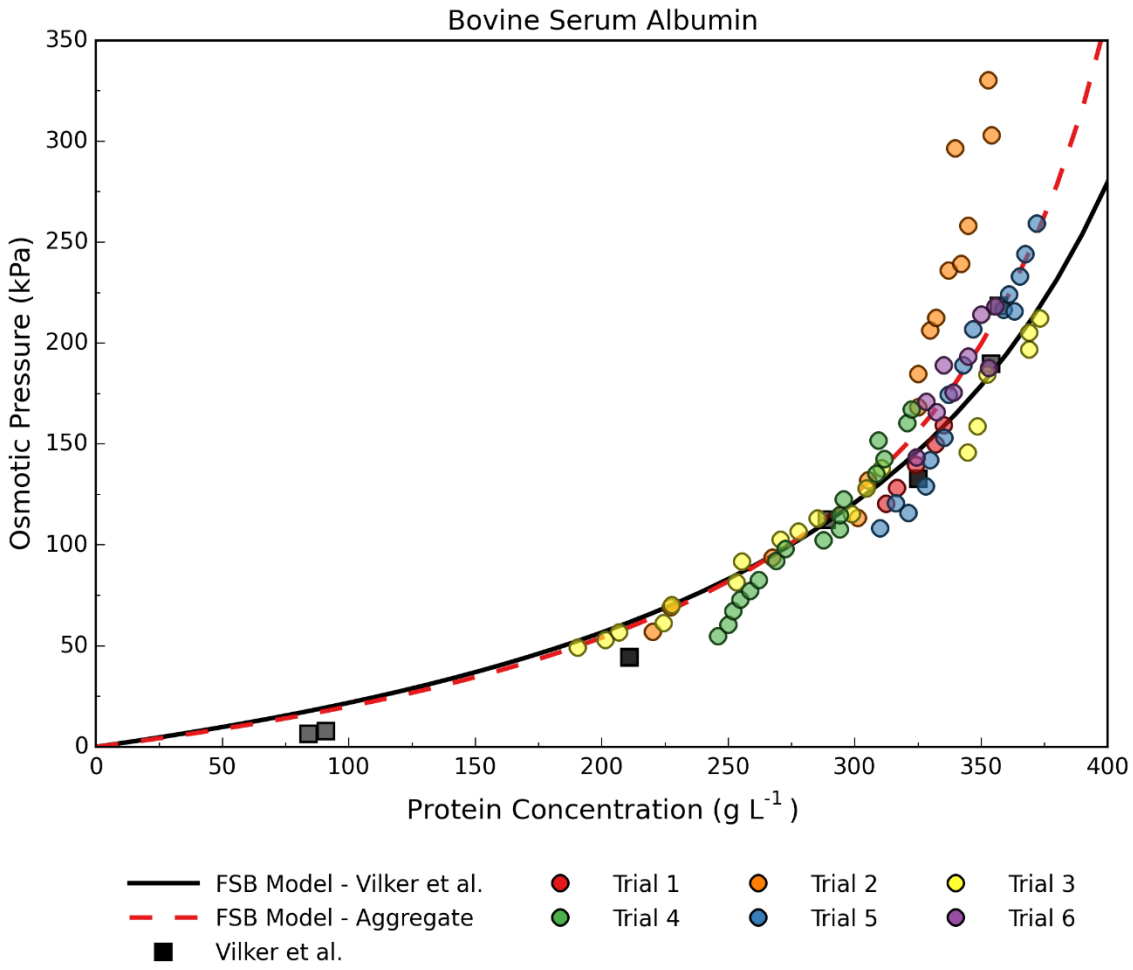


Figure 9.3: Osmotic pressure data, from a concentrating osmometer, for Bovine Serum Albumin in 0.15M NaCl at pH 7.4, 25°C, along with literature osmotic pressure values.

Table 9.2: Regressed Ion Binding and Hydration Parameters from Osmotic Pressure Data

0.15 M NaCl, pH 7.4, 25°C	**Concentration ($g\ L^{-1}$) [Data Points]**	**Hydration ($\frac{mol\ H_2O}{mol\ BSA}$) ν_{12}**	**Ion Binding ($\frac{mol\ Salt}{mol\ BSA}$) ν_{32}**	**Covariance**
Literature*	84 – 357 [9]	5499 ± 557	13.05 ± 2.17	2.08×10^{-5}
Trial 1	312 – 335 [5]	5573 ± 425	13.30 ± 1.73	3.42×10^{-7}
Trial 2	220 – 353 [15]	6356 ± 349	15.46 ± 1.54	8.15×10^{-6}
Trial 3	191 – 373 [19]	3845 ± 419	6.88 ± 1.59	1.06×10^{-5}
Trial 4	246 – 323 [17]	6343 ± 349	15.59 ± 1.25	2.59×10^{-6}
Trial 5	310 – 372 [15]	5563 ± 331	13.30 ± 1.36	3.54×10^{-6}
Trial 6	324 – 356 [9]	4762 ± 1571	9.86 ± 6.26	2.46×10^{-4}
Aggregated Trials	191 – 373 [80]	4827 ± 542	10.10 ± 2.16	3.82×10^{-5}

*Data from Vilker et al.[13]

CHAPTER 10.

INTERROGATING THE OSMOTIC PRESSURE OF SELF-CROWDED BOVINE SERUM ALBUMIN SOLUTIONS: IMPLICATIONS OF SPECIFIC MONOVALENT ANION EFFECTS RELATIVE TO THE HOFMEISTER SERIES

10.1. Introduction

10.1.1 Background

Ion effects on protein solutions have been investigated for more than a century beginning with Hofmeister.[21] Hofmeister and colleges found that ions could be arranged based on their ability to crystallize protein solutions generating the famous Hofmeister series. Figure 1 illustrates the range of anions when in aqueous solutions for their ability to salt out egg white protein.[21,22]

Subsequently, researchers, including Hofmeister, have searched for the reasoning behind this phenomenon.[21,23–25] Historically, the water/ion interaction were analyzed to give rise to the categorization of cations and anions as either chaotropes or kosmotropes based on potential of the ion to order water. Chaotropes are denoted as "water structure breaking" and kosmotropes are "water structure making". The concept of water ordering implies that ions influence water well away from their solvation layer. Recently, researchers have argued that the water ordering concept is overstated.[26–32] However, local interactions of ions with water, such as electrostriction, are generally accepted. Electrostriction is the reduction of volume of a solvent due to the presence of ions.[33] In fact, the rod-like shape of SCN^- naturally induces electrostriction of the local solvent.[33]
A more quantitative approach categorizes ions as chaotropes or kosmotropes based on their Jones-Dole viscosity B coefficient.[34,35] This relationship was based on original observations by Poiseuille that some salts increased or decreased the viscosity of water.[36] Chaotropes are large weakly hydrated monovalent ions of low charge density (e.g., SCN^-, I^-, NH_4^+, K^+). Chaotropes are denoted

as “salting in” because they can result in a decrease in stability and protein crystallization, upon the addition of salt. Kosmotropes, on the other hand, are known for their ability to “salt out” protein solutions and are associated with increased stability and protein crystallization. This grouping consists of small strongly hydrated ions of high charge density (e.g., SO_4^{2-}, HPO_4^{2-}, F^-, Al^{3+}, Ca^{2+}, and Na^+).

Subsequently, the concepts of chaotropes and kosmotropes has morphed into new categorization for ions. Specifically, they are correlated with the size of the ion, electronegativity and charge density.[35,37] Size of ions is assumed to affect the salt ion-binding. An ionic sphere of a large ion results in a steric hindrance when ion-binding sites on protein are not fully exposed or when they are in narrow cavities on the protein surface. Ionic charges are expected to have influence on salt ion-binding since ion-residues interactions are electrostatically-driven. Collins (2004) suggested that proteins with an excess of weakly hydrated positively charged surface amino acids readily crystallize with weakly hydrated ions such as chloride or thiocyanate, but have difficulty with strongly hydrated ions such as sulfate or phosphate.[38] Also, Zangi *et al*. (2006) reviewed the role of simplified hydrophobic surface interfaces in solution and determined charge density was the key to Hofmeister’s ordering.[39] However, Waldron *et al*. (2003) used calorimetry to show that anion binding of their model protein-protein complex was independent of charge.[40] Electronegativity, the tendency of an atom to attract electrons, has been shown to have an impact on the binding affinity to some proteins.[41] Infrared studies have shown that kosmotropic salts can increase ion hydration over chaotropic salts; in addition, protein hydration decreases in the presence of kosmotropic salts when compared to chaotropic salts.[42] Majumdar *et al.* (2013) used size exclusion chromatography to determine the rate of protein aggregation between SO_4^2, Cl^-, and SCN^-. They found that NaSCN accelerated the protein aggregation rate compared to NaCl, while Na_2SO_4 had a small stabilizing effect on the proteins, reducing the rate of aggregation.[43] A comprehensive

summary on the physical properties associated with the Hofmeister series can be found elsewhere.[32,44,45] Table 1 provides a brief historical perspective of proposed justification for anion hierarchy in the Hofmeister series.

We now recognize that, although some generalizations can be made for the association of ions with proteins, no simple relationship captures all of protein/ion/water observed behavior. The classic example is lysozyme; at moderate pH and ionic strength it demonstrates a reverse Hofmeister effects.[46,47] Consequently, more recent work has abandoned the search for a unified global model for the observed Hofmeister effects and have examined the anion-specific surface/water/ion complex. Many have proposed that the anion-protein interaction occurs at the backbone of the protein and have examined this on a molecular scale using molecular dynamics and experimental methods for well-defined peptides.[44,47–52] Algaer and van der Vegt (2011) concluded that I^- interacted primarily with the hydrophobic parts of their model peptide.[50] However, Rembert *et al*. (2012) determined that SCN^- and I^- bind at hybrid amide nitrogen/α-carbon binding sites where Cl^- only weakly binds at the same locations.[51] They also concluded that hydrophobic sites do not contribute significantly to anion binding. More recently, Okur *et al*. (2017) showed that steric arrangements at ion binding sites may dominate the outcome of protein-ion interactions.[44]

Thus, the complexity of protein behavior in aqueous salt solutions warranted continued detail analysis. Overall, however, a central element of interaction is associated with the specific contributions from protein-ion, protein-water and ion-water interactions. In this regard, information relative to ion binding and protein hydration in the presence of various salts provides important clues to the protein-ion-water systems.

With regards to ion-protein interactions, a number of researchers over the century have attempted to quantify ion binding. Linderstrøm-Lang (1924) accomplished the first theoretical treatment of ion binding in native proteins.[53] Linderstrøm-Lang applied the interionic attraction

theory of Debye and Hückel[54] to the analysis of the influence of electrostatic forces on acid-base equilibria in proteins. Tanford (1950) used the theory of Linderstrøm-Lang to provide a computed titration curve for human serum albumin (HSA) and protons.[55] Scatchard *et al.* (1949, 1950) also used the methods of Linderstrøm-Lang to address chloride ion and thiocyanate ion binding to HSA from sodium salts in low concentrations using a membrane dialysis and an electromotive force method.[56–58] They found that thiocyanate ions bind at higher numbers than chloride ions at similar concentrations. Fox *et al* (2015) used ITC and X-ray crystallography to examine the effect of anions on protein solvation in concavities.[59] Others argue that electrostatic contributions to protein-ion interaction are not enough in interpreting interactions and that dispersion interactions, which can be accessed from bulk properties, and ionic quantum fluctuation forces are critical.[22,60,61]

The difficulty in understanding ion interactions with protein solutions is exacerbated in crowded environments. In biological systems, it is recognized that macromolecular crowding may be a significant factor in ion interactions in protein solutions.[62] Zimmerman and Trach showed that interactions of ions on proteins that were observed in dilute in-vitro solutions, did not exists in in-vivo crowded systems where the total protein concentration can be as high as 400 g L^{-1}. Reboiras *et al.* (1986) recognized the significance of solution crowding on ion binding to proteins and extended the work of Scatchard (1950), by studying potassium salt binding to isoionic bovine serum albumin (BSA) at high concentrations (up to 268 g L^{-1}), using the EMF method with ion-exchange membrane electrodes.[57,63] They showed a protein concentration dependency in ion binding.

10.1.2. Relevance of Osmotic Pressure in the Interpretation of Anion Effects

The FSB Model Directly Relates Osmotic Pressure in Crowded Solutions to Protein Hydration and Ion Binding

Osmotic pressure does not give direct crystallization information. Nevertheless, a number of researchers have used osmotic pressure to address observed phenomena of proteins in ionic aqueous solutions via protein-protein interactions.[13,17,64–67] However, we have shown that, via our FSB model, the bulk property, osmotic pressure, of self-crowded proteins and crowded binary globular protein solutions can directly provide ion binding and hydration properties of proteins, specifically in the highly crowded regions where the rate of change of osmotic pressure to protein concentration is highly non-linear.[10,11,19,20,68–73] Because the FSB model focuses on protein hydration and ion-binding, it can provide a fortuitous approach in interrogating anion effects on crowded protein solutions.

Theoretical Development of the FSB Model

The concept of the free solvent model dates back to the twentieth century when Frazer and collaborators addressed non-idealities in sucrose solutions by considering hydration.[74,75] Scatchard (1946) also proposed a 'free-solvent' model that addressed the interaction of diffusible species with the non-diffusible species in solution.[76] Our FSM model considers the physically realizable parameters, salt ion binding and hydration, as dominant in determining non-ideality and uses the mole fraction as the concentration variable.[77] Our FSB model is remarkably successful in predicting the osmotic pressure of albumin in solutions when using the ion binding parameters determined by Scatchard.[57,77] Not only does the predicted ion binding match that of Scatchard, but the hydration, associated with the water deviating from the bulk chemical potential, was associated with a monolayer. Analysis of the FSB model results has also shown that the non-idealities in self-

crowded proteins solutions is dominated by monolayer hydration and ion binding for many globular proteins solutions at moderate ionic strength across a decade of differences in molecular mass.[69] As a result, the FSB model, under the assumption that only a monolayer of water interacts with each protein, provides a means for determining the solvent accessible surface area (SASA) of proteins when protein structure is unavailable.[10]

Our original FSB osmotic pressure model is fully developed elsewhere.[68,77] In summary, the FSB model treats the hydrated protein as a separate macromolecule and all water and salt ions in its 'influence' are absorbed in its definition. The model assumes that counterions (Na^+) to binding anions, are also within the influence of the macromolecule. Scatchard *et al*. (1950) showed that Na^+ do not bind to albumin in dilute solutions.[57] This assumption has been discussed elsewhere.[78] If this assumption is not made, then electroneutrality must be considered.[57,58,79]

With 'influenced' water and ions associated with the protein macromolecule, the FSB model recalculates the mole fraction of the remaining 'free solvent' (water and ions) that have the propensity to diffuse across the semi-permeable membrane. The subsequent predicted osmotic pressure, based on the modified mole fraction of the free solvent, rests on the ideal solution framework where additional macromolecule 'free-solvent' interactions are ignored. Figure 2 illustrates this general concept. A more detailed structural orientation of water about anions can be found elsewhere.[80]

Thus, the FSB model renders the solution ideal with respect to the remaining, diffusible solvent species. The modified mole fraction of the free water, $(x_1)_{\mathrm{FW}}$, considers the hydrated macromolecule as the impermeable solute. The free-solvent model with the mole fraction of the free water, $(x_1)_{\mathrm{FW}}$, as the composition variable is

$$\pi = -\frac{RT}{\overline{V}_1} ln\left(\frac{(x_1^{II})_{FW}}{(x_1^{I})_{FW}}\right) \tag{10.1}$$

where R is the ideal gas constant, T is absolute temperature, $\overline{V}_1$ is the specific volume of species 1 and superscripts I and II represent the solvent and solution chambers. The mole fraction of free water is determined by the remaining moles of water that are not bound to the protein. Assuming the solution is made up of n distinct species and p proteins, and letting species 1 be the solvent, species 2 through $(p+1)$ be the proteins, and species $(p+2)$ through n be the remaining diffusible species, the initial total moles of the solution in compartment II is $N^{\mathrm{II}} = \sum_{i=1}^{n} N_i^{\mathrm{II}}$, where i denotes each species. The final total moles of free-solvent in chamber II, after protein-solvent interactions, is $N_*^{\mathrm{II}} = N^{\mathrm{II}} - \sum_{i=1;i\neq 2\to p+1}^{n} \sum_{j=2}^{p+1} \nu_{ij} N_j^{\mathrm{II}} - \sum_{j=2}^{p+1} N_j^{\mathrm{II}}$, where N_j^{II} denotes the moles of protein j in solution and ν_{ij} is the number of moles of species i interacting with protein j to make the hydrated protein. Then, the mole fraction of free-solvent in chamber II is

$$(x_1^{II})_{FW} = \frac{N_1^{II} - \sum_{j=2}^{p+1} \nu_{1j} N_j^{II}}{N_*^{II} + \sum_{j=2}^{p+1} N_j^{II}} \tag{10.2}$$

while in chamber I, the mole fraction of free-solvent is[52]

$$(x_1^{I})_{FW} = \frac{N_1^{I}}{\sum_{i=1,i\neq 2\to p+1}^{n} N_i^{I}}. \tag{10.3}$$

In this work, we also consider ion hydration and introduce the ion-hydration parameter, ν_{32}. Then, for a single protein species in a monovalent salt aqueous solution, the free-solvent mole fraction in the solvent side of the osmometer, we write

$$(x_1^{I})_{\mathrm{FW}} = \frac{N_1^{I} - \nu_{13} N_3^{I}}{N_1^{I} - \nu_{13} N_3^{I} + N_3^{I}}, \tag{10.4}$$

and for the solution chamber,

$$(x_1^{II})_{\mathrm{FW}} = \frac{N_1^{II} - \nu_{12} N_2^{II} - \nu_{13} N_3^{II}}{N_1^{II} - \nu_{12} N_2^{II} - \nu_{13} N_3^{II} + N_2^{II} + N_3^{II}(1 - \nu_{32} N_2^{II})} \tag{10.5}$$

to obtain

$$\pi = \frac{RT}{\overline{V}_1}\frac{\left(N_1^I - \nu_{13}N_3^I\right)\left(N_1^{II} - \nu_{12}N_2^{II} - \nu_{13}N_3^{II} + N_2^{II} + N_3^{II}\left(1 - \nu_{32}N_2^{II}\right)\right)}{\left(N_1^I + N_3^I\,(1 - \nu_{13})\right)\left(N_1^{II} - \nu_{12}N_2^{II} - \nu_{13}N_3^{II}\right)}. \quad (10.6)$$

The Calculated Hydration and Ion Binding Determined from the FSB Model are Independent

Regression of the protein hydration and protein-ion binding through the FSM model, results in solutions with very low covariance with respect to these two parameters. This independence of parameters is critical and provides additional credibility to model relevance and the validity of the calculated hydration and ion-binding parameters. Moreover, the estimated parameters are dominated by the highest concentration results where deviation of the osmotic pressure from ideality is most prevalent. This is indicative of the strongly non-ideal behavior of the osmotic pressure profile, with respect to concentration, that is characterized by the FSB model.[77]

The FSB Model May Indicate Protein Interaction via Reduction in Hydration

Non-interactive proteins with a prescribed hydration and ion-binding can provide a signature change in the osmotic pressure profile.[54] However, interactive proteins, via aggregation, generate a reduction in hydration or solvation in the FSB model results. Thus, analysis of the FSB model results may also provide indication that specific anions result in significant protein-protein interactions which could be due to aggregation, denaturation or unfolding.[19]

The FSB Model Provides Explanation for Negative Second Virial Coefficient for Non-Attractive Proteins

The FSB model was also used to explain the observed negative second virial coefficient for non-attractive lysozyme solutions. In this work, under the assumptions of the FSB model, that

negative second virial coefficients correspond with a higher local ionic strength around the protein than that in the bulk.[11]

The FSB Model Provides a Theoretical Saturation Limit

Because the FSB model provides a relationship of protein hydration, it also provides a theoretical prediction of the protein saturation concentration. Previously we showed that the predicted saturation concentration for bovine immune-gamma globulin in phosphate-buffered solution of 0.13 M salt (0.12 M NaCl, 0.0027 M KCl and 0.01 M phosphate buffers) at 7.4 pH was 546.7 g/L, which compared well with experimental observation.[68]

10.1.3. Goal of this Work

In this study we address the monovalent anion-specific effects on self-crowded BSA solutions via the calculated hydration and ion binding parameters generated from regression of osmotic pressure data via the FSB model (Eqn (6)). Specifically, we determine the osmotic pressure of BSA in NaF, NaI and NaSCN solutions up to 538 g/L and compared the results to our previously reported results for BSA in NaCl.[77] All studies are done with a 0.15 M salt concentration and at pH 7.4. Using the FSB model, we determined the hydration and ion-binding for each specific monovalent salt solution. Because the free solvent model is predictive at the highest concentrations, these results represent values for protein concentrations at the highest range. The results of this work will be interpreted relative to the Hofmeister series for these anions.

10.2. Materials & Methods

10.2.1. Measurement of Osmotic Pressure

The osmotic pressure experiments were conducted for protein concentrations (No. A30075, BSA, Research Products International, Mt Prospect, IL), up to near-saturation, in 0.15 M sodium salt solutions (sodium fluoride (No. S6776, Sigma-Aldrich, St. Louis, MO); sodium chloride (No. S9888, Sigma-Aldrich, St. Louis, MO); sodium iodine (No. 409286, Sigma-Aldrich, St. Louis, MO); sodium thiocyanate (No. 251410, Sigma-Aldrich, St. Louis, MO)) at pH 7.4 and 25°C.

Osmotic pressure is obtained from an osmometer described elsewhere.[77] The protein chamber is filled with protein solution, at the desired concentration, until a meniscus is formed above the chamber walls. Any air bubbles present are removed from the solution. A membrane (NADIR® PM UP010, 10k Da MWCO, Polysulfone, Microdyn Nadir, Germany, Wiesbaden), is soaked in ultrapure water (EASYpure RoDi D13321, Thermo Scientific Barnstead Water System, Thermo Fisher Scientific, Waltham, MA) for at least one hour and is then placed on top of the protein chamber, ensuring no air pocket forms. The protein chamber is sealed and excess solution is expelled. Next, a membrane support is placed on the opposite side of the membrane to prevent bowing deformation caused by the increased osmotic pressure. A rubber gasket is then placed on the other side of the membrane housing to seal the solvent chamber from leaks. The osmometer assembly is then screwed together and connected to a beaker of solvent, open to atmosphere. Solvent is then circulated through the solvent chamber using a peristaltic pump (Model EW-07524-50, Master Flex L/S, Cole Palmer, Vernon Hills, IL). Pressure reading are obtained using a pressure transducer (Model 7356-51 Cole-Palmer, Vernon Hills, IL) that is digitally recorded through data acquisition (Model NI SCC-68, National Instruments, Austin, TX). Pressure reading stabilize after 5-6 hours.

Solvent solutions are prepared by dissolving the proper amount of sodium salt in one liter of nanopure water to produce a 0.15 M solution. To prepare a BSA solution, this solvent is then used to dissolve a weighed amount of Bovine Serum Albumin, using a stir bar to facilitate mixing. The solution pH is then measured by a pH Meter (Model 13-641-253, ThermoScientific Orion 720A+, Thermo Fisher Scientific, Waltham, MA) and adjusted using 1 M HCl (No. HX0603, Millipore Sigma, Burlington, MA) and 1 M NaOH (No. S318, Thermo Fisher Scientific, Waltham, MA) while undergoing stirring to prevent local denaturation. The amount of acid and base used to adjust pH is considered part of the solvent and is taken into account when determining concentration. Before a solution is used, the pH is checked to be within 0.05 pH of their desired value. The concentration of solutions are determined by dividing the amount of protein or salt by the volume of solvent used to make the solution. The volume of solvent includes volume of protein or salt in the solution using the specific volume of the protein or the density of the salt.

10.2.2. Determining Hydration and Ion Binding

The parameters for hydration, ν_{12}, and ion binding, ν_{32} are determined by nonlinear regression of Eq. 6 (TableCurve 2D, Systat Software, San Jose, CA) to best fit the osmotic pressure versus concentration profiles for each anion solution in the study. The ion hydration parameter, ν_{13}, was estimated from the literature (Table 2). Table 2 also includes the reported crystal radius, apparent dynamic hydration number (ADHN) and kosmotrophicity for each ion. The crystal radius is inversely related to surface charge density.[35] The ADHN for each ion is considered to be a primary factor in ion binding to proteins.[81] Kosmotrophicity is a categorization of whether an ion behaves as a kosmotrope (k) or a chaotrope (c). This evaluation is based on a number of factors including the viscosity B coefficients.[82]

10.3. Results and Discussion

10.3.1. Osmotic Pressure Results

Measured osmotic pressure profiles for protein concentrations up to near saturation for BSA in NaF, NaI, and NaSCN are shown in Tables 3 and plotted in Figure 3 with the previous NaCl data from Yousef *et al.* (1998).[77] All solutions show classic non-linear monotonic increase in rate of change in osmotic pressure versus increased concentration, albeit the data associated with NaSCN is relatively noisy compared to the other solutions. All proteins remained in solution for all measured protein concentrations.

The osmotic pressure data demonstrates a distinct separation in osmotic pressure relative to anion in solution at protein concentrations above approximately 300 g/L. At a representative concentration of about 448 g/L, the osmotic pressure for NaCl is highest, followed by that of NaSCN and NaI, with that from the NaF solution being the lowest. The osmotic pressure for NaF appears to increase in its rate of change relative to NaSCN and NaI at the highest concentrations.

10.3.2. The FSB Model Provides Excellent Fit with Regressed Ion Binding and Hydration Values

As can be seen from Figure 3, the two parameter FSB model successfully captures the physics of the osmotic pressure behavior for increasing protein concentrations for NaCl, NaI, and NaF solutions. The model also well represents the data for NaSCN, given that the data is relatively noisy. The regressed ion binding and hydration values determined for the best fit of the FSB model to each of the data sets are summarized in Table 4.

The ion binding and hydration are well within range of values independently determined by others. Studies using water-^{17}O magnetic resonance showed that the hydration for globular proteins is on the order of 1 g H_2O/g protein.[83] The regressed values for hydration in this study range from 0.64 g H_2O/g BSA to 1.11 g H_2O/g BSA; well within physically realistic values. In

addition, as mentioned earlier, Scatchard reported that ion binding for dilute NaCl solutions of human serum albumin (HSA) was 8 mol NaCl/mol HSA, the same as the regressed ion binding using the FSB model. Thus, given that the fitted FSB model captures the physics of the osmotic pressure curve for concentration and the hydration and ion binding parameters are physically meaningful, this FSB modeling approach has significant credibility and can be used to glean protein hydration and ion binding relationships for crowded solutions within its approximations.[19]

Two distinct behaviors are seen in Figure 3. NaCl and NaF both show an initial gradual change in osmotic pressure with increasing concentration followed by a more rapid change in osmotic pressure at higher concentrations. NaI and NaSCN have a less aggressive increase in osmotic pressure at the higher concentrations relative to the initial rates of change in lower concentrations. Perhaps due to the noisy NaSCN data, but there was no discernable difference in the osmotic pressure rate of change with respect to concentration for NaI and NaSCN.

Confidence intervals and parameter sensitivity for each solution can be found in the Supplement.

10.3.3. Ion Binding

While the ion binding for NaCl was consistent with the work of Scatchard and colleagues, the ion binding for NaI and NaSCN were not. Among the three salts NaCl, NaI and NaSCN, Scatchard *et al.* (1949, 1950) showed, using EMF and titration, that HSA has higher binding with thiocyanate ions, followed by iodide with chloride having the lowest ion binding.[56–58] The FSB model shows that for BSA in self-crowded concentrations, the binding is highest for chloride with no appreciable binding for NaI and NaSCN. This may be due to any number of factors or combination thereof including size of hydrated ions or surface charge density. This may also be a result of variation in ion binding at very high concentrations. Scatchard *et al.* used protein

concentrations up to approximately 300 g/L, much lower than those used in this study. Nevertheless, this observation is consistent with ionic sphere of a large ion that results in a steric hindrance or reduced surface charge density where ion-binding sites on protein are not fully exposed or when they are in narrow cavities on the protein surface or the electrostatic interactions are relatively weak.[35,84]

It should be noted that regression analysis to fit the FSB model to the osmotic pressure data using the ion binding values from Scatchard *et al.* (1950)[57,58] for NaI and NaSCN did not converge. This limitation actually gives further credibility to the independence of ion binding and hydration in fitting the FSB model.

10.3.4. Protein Hydration

Kosmotropes Have a Significantly Higher Protein Hydration than Chaotropes

The ion binding for Cl^- and Fl^- are within experimental error with data associated with Cl^- having a significantly higher hydration. The hydration for NaCl was 1.113 ± 0.030 g H_2O/g BSA while that for NaF was 0.962 ± 0.035 g H_2O/g BSA. The 12% reduction in hydration from solutions of NaCl to NaF had a substantial shift in the osmotic pressure for the highly self-crowded solution; at 446-448 g/L BSA, the osmotic pressure was reduced by nearly half in NaF as compared to NaCl.

Kosmotropes Increase Rate of Change in Osmotic Pressure via Increased Hydration

The NaI and NaSCN results are not significantly different between each other but have significantly lower hydration than NaCl and NaF solution results. However, because of the nearly unobserved ion binding, the osmotic pressure of NaI and NaSCN was over 50 kPa higher than that of NaF. We showed previously that at high self-crowded concentrations, increases in ion binding

decreased osmotic pressure but osmotic pressure rises with increases in hydration, with respect to the FSB model.[73]

Hydration Changes Due to Anion Effects May Correlate with SASA Changes

One note, the correction for ion hydration did not show significant changes for the NaCl regressed parameters.[77] This is due to large relative hydration of protein that is also absorbed in the macromolecule assumption. We have previously reported that a monolayer of water on BSA is approximately 1g H_2O/1g BSA when in moderate NaCl at physiological pH.[69] This relationship allows one to approximate the solvent accessible surface area (SASA) for other globular proteins.[10] The regressed values of 0.64 g H_2O/1g BSA and 0.69 g H_2O/1g BSA for NaI and NaSCN, respectively, may imply that the protein reduced in volume or aggregation may be taking place.

Assuming the radius is that of equivalent spheres and there is a monolayer of water in each case, the radius reduction from 1.111 g H_2O/1g BSA to 0.64 g H_2O/1g BSA corresponds to about 24%. Tanford *et al.* (1955) reported a similar increase in BSA radius due to shifts in pH below its isoelectric point. However, in this case, BSA is assumed to be in a compact form at physiological pH that undergoes reversible expansion at low pH.[85] The observed change may more likely be coupled to protein aggregation.[86–88]

It is generally accepted that proteins at high concentrations partially unfold resulting in non-polar interactions that induce aggregation.[88] Aggregation would reduce the SASA and this would be reflected in the regressed hydration number from the FSB model.

We investigate the amount of SASA reduction that would be associated with BSA aggregation using the rigid-body docking tool, ZDOCK.[89] This algorithm considers electrostatic, statistical potential and shape complementary in determining the change in SASA for protein–protein interactions, $\Delta SASA$. The BSA crystal structure was obtained from the Protein Data Bank

(PDB ID:4F5S, modified to contain only 1 BSA molecule).[90,91] Initially ZDock was used to predict the docking of two monomers. The top reported results were analyzed by using Chimera (UCSF) software to determine the SASA for the multimer.[92] A MatLab script (MathWorks, Natick, MA) summed the UCSF Chimera analysis to determine the multimers total available SASA. The monomer and/or a representative sample of the 10 outputs, determined by average available SASA, was used in determination of larger multimers. Table 5 summarizes these results.

ZDock was able solve the SASA for BSA multimers up to octamers. At the octamer level, this corresponds to a 17% decrease in SASA when comparing eight monomers. Thus, the hydration results from regression of the FSB model remain plausible in reference to potential aggregation of self-crowded BSA in these chaotropic solutions.

Relevance of Hydration and Ion Binding to the Second Virial Coefficient

The second virial coefficient for osmotic pressure have been traditionally associated with solute-solute interactions and have been found to be dependent of solute concentration.[13] The values of the second virial coefficient to not provide direct insight to the phenomena of protein-protein interaction, however, it is well accepted that, larger second virial coefficients account for larger aggregation formation (higher multimers).[93] Previously, we showed the relationship between the second virial coefficient and the ionic strength ratio which is determined from the FSB model parameters.[11] The ionic strength ratio is, α, is

$$\alpha = \frac{\nu_{32}/\nu_{12}}{M} \tag{10.7}$$

where M is the bulk ionic strength of the solution. Our results showed that the decrease in α corresponded to an increased second virial coefficient, implying increased protein aggregation. Since the values for α for both SCN^- and I^- are significantly lower than those for Cl^- and F^-, this may imply that the chaotropic solutions have higher aggregation.

Hydration Shifts May Be a Consequence of Excluded Volume Effects

Tadeo *et al. (*2007) examined the influence of Hofmeister anions on protein stability using the B1 domain of protein L as their model.[86] They determined that effects of anions did not impact the SASA of the protein domain. Assuming the validity of the regressed hydration values, they may also be a quantitative representation of an excluded volume effect.[64,67,94–98] Nevertheless, these results will be useful as researchers continue to elucidate the complex mechanisms of anions on protein hydration in crowded solutions.

10.3.5. Theoretical Saturation

Because the FSB model provides the hydration of proteins in crowded solutions it can be used to determine a theoretical saturation limit. This theoretical saturation limit is based on the assumptions of the FSB model including negligible variation in hydration with respect to protein concentration. We showed previously that despite the low osmotic pressure profile for lysozyme, it salted out well below the predicted theoretical saturation limit.[69] In any case, regardless of its actual significance, it can be used to qualitatively evaluate anion effects on crowded protein solutions.

The theoretical saturation limit for BSA in each of the monovalent sodium salts was determined using regressed hydration and are presented in Table 6. As can be seen, the theoretical saturation limit follows in the order of Cl < F < I and SCN, which is not in the traditional Hofmeister order. The theoretical saturation values for NaF solutions are determined by the highest concentration measured in this study and, from reviewing Figure 3, its value may be substantially lower with osmotic pressure results from higher concentration levels. Nevertheless, it is clear that Cl will have the lowest saturation limit which is inconsistent with a Hofmeister order. The model

also predicts that NaSCN and NaI solutions have the highest theoretical saturation which implies that these chaotropes keep proteins in solution longest among the anions studied. However, in this study, actual saturation values were not determined. These results are consistent with the work of Rembert *et al.* (2012) who indicate that SCN^- and I^- keep proteins in solution.[51] However, we have not extrapolated on to BSA their observation (based on their model polypeptide) that these anions interact with a hybrid binding site between amide nitrogen and adjacent $\alpha-$carbons. Table 6 summarizes the ionic strength ratio for each solution.

10.4. Conclusion

In this work we investigated the FSB osmotic pressure model as a tool to interrogate the anion effects on self-crowded BSA solutions of sodium salts at moderate ionic strength and pH 7.4. The FSB model assumes an ideal relationship between free-solvent and hydrated macromolecules that encompass protein and influenced water and ions. Solutions of NaF, NaCl, NaI and NaSCN were investigated. The results indicate that the NaF and NaCl result in both higher ion binding and hydration than the chaotropic solutions of NaI and NaSCN. The values of hydration for NaI and NaSCN solutions were substantially reduced which may imply increased aggregation or excluded volume. The ion binding for NaI and NaSCN was determined to be negligible compared to solutions of NaF and NaCl. The FSB model continues to require validation. However, it appears to be an effective tool for probing the effects of anions in crowded protein solutions in terms of physically realizable parameters such as ion binding and protein hydration, especially near saturated conditions.

$$HPO_4^{2-} > SO_4^{2-} > F^- > Cl^- > Br^- > NO_3^- > I^- > ClO_4^- > SCN^-$$

Figure 10.1: The Hofmeister series for anions ranged from left to right for their ability to salt out egg white proteins in solution.[2]

Table 10.1: Historical Perspective of Anion Significance in the Hofmeister Series
[4, 6, 13, 16, 34]

$HPO_4^{2-} > SO_4^{2-} > F^- > Cl^- > Br^- > NO_3^- > I^- > ClO_4^- > SCN^-$	
Kosmotropes	**Chaotropes**
Water structure making	Water structure breaking
Salting out	Salting in
Strongly hydrated	Weakly hydrated
Small size, high charge density	Large size, low charge density
Stronger interactions with water than self	Weaker interactions with water than self
Reduces protein aggregation rate	Increases protein aggregation rate

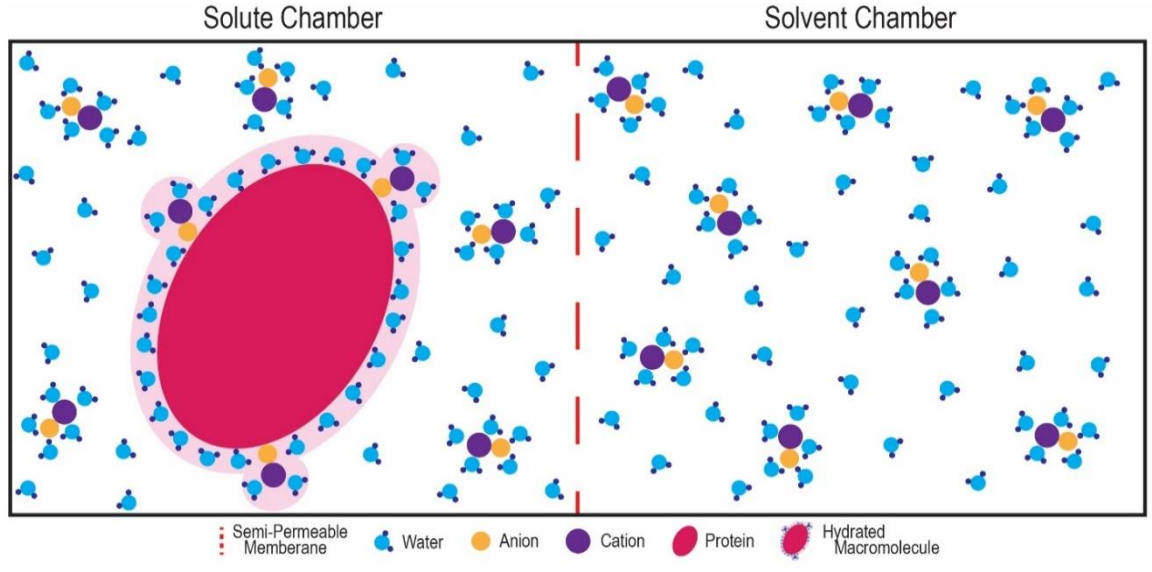


Figure 10.2: Illustration of relationship of species in the FSB model. Water and ions influenced by the presence of the protein are collapsed into the single 'hydrated macromolecule' species. Thus the hydrated water and bound ions are no longer considered part of the continuum. The resulting mole fraction of solvent or water is calculated based on the remaining diffusible species in both chambers. In this study, hydration of the ions is also taken into account. The regressed ion-binding and hydration parameters for self-crowded BSA solutions are determined for 0.15 M sodium salt solutions with the monovalent anions, Cl^-, F^-, I^- and SCN^- and interpreted with respect to their position in the Hofmeister series.

Table 10.2 Properties of Ions in Study

Anion	Bound Water Molecules [71]	Crystal Radius (Å) [72]	Apparent Dynamic Hydration Number [70, 73]	Kosmotrophicity [34]
F^-	7.2	1.33	5	k
Cl^-	5.5	1.81	0	k/c
I^-	6.6	2.20	0	c
SCN^-	8.8	2.13	-	c

k = kosmotrope, c = chaotrope

Table 10.3 Osmotic Pressure of BSA in 0.15M NaF, NaCl, NaI, and NaSCN, pH 7.4, 25°C

NaF		NaCl[a]		NaI		NaSCN	
[BSA] (g/L Soln)	**Osmotic Pressure (kPa)**	**[BSA] (g/L Soln)**	**Osmotic Pressure (kPa)**	**[BSA] (g/L Soln)**	**Osmotic Pressure (kPa)**	**[BSA] (g/L Soln)**	**Osmotic Pressure (kPa)**
297	67.6	84	6.4	296	93.8	299	69.2
314	62.7	91	7.9	343	183.4	299	69.7
343	115.1	211	44.3	397	259.2	322	71.4
372	123.4	211	44.5	421	283.4	346	94.3
396	172.4	289	113.0	447	316.5	397	114.5
422	202.0	325	133.0	471	347.5	343	173.4
446	253.0	354	190.0	481	409.5	394	267.7
470	293.0	357	218.0	487	413.7	423	293.0
480	346.1	413	349.0	507	474.4	468	403.6
		428	374.0	522	513.0	538	484.7
		448	485.0			515	504.0

a. NaCl data from Yousef et al.[66]

Table 10.4: Regressed Ion Binding and Hydration Parameters from Osmotic Pressure Data

0.15 M Salt, pH 7.4	**Ion Binding** $\left(\frac{\text{mol Salt}}{\text{mol BSA}}\right)$ ν_{32}	**Scatchard Ion Binding** $\left(\frac{\text{mol Salt}}{\text{mol HSA}}\right)$ [39]	**Hydration** $\left(\frac{\text{mol } H_2O}{\text{mol BSA}}\right)$ ν_{12}	**Hydration** $\left(\frac{\text{g } H_2O}{\text{g BSA}}\right)$	**Covariance**
NaF	7.4 ± 0.59	N/A	3546 ± 129	0.962 ± 0.035	5.77 x10^{-6}
NaCl	7.7 ± 0.57	8	4102 ± 111	1.113 ± 0.030	6.72 x10^{-6}
NaI	0.8 ± 0.97	11	2407 ± 177	0.643 ± 0.048	2.71 x10^{-5}
NaSCN	2.6 ± 2.11	15	2554 ± 394	0.693 ± 0.107	1.47 x10^{-3}

HSA: human serum albumin

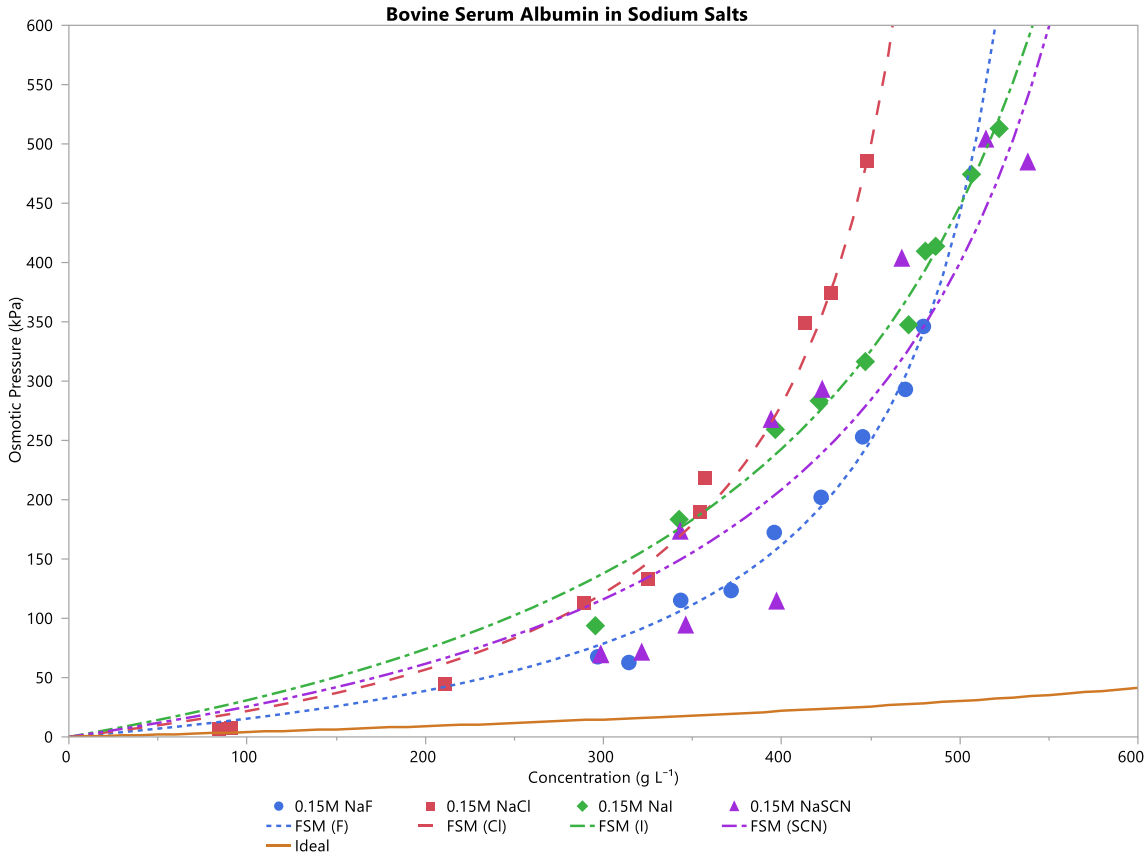


Figure 10.3: Measured osmotic pressure vs. BSA protein concentration in NaF, NaCl, NaI and NaSCN. The dashed lines present the best-fit FSB model from regressed hydration and ion binding for each of the monovalent sodium salts investigated. As can be seen, the FSB model has an excellent fit for NaF, NaI and NaCl and is acceptable for the relatively noisy NaSCN data. The solution for the ideal model for BSA (no hydration or ion binding) is also shown in the solid line as a reference. Data for the NaCl case is from.[66]

Table 10.5: Evaluation of $\Delta SASA$ of BSA Multimers

COMPARISON (Δ)	$\boldsymbol{SASA}$ **(Å²)**	$-\Delta\boldsymbol{SASA}$ **(Å²)**	$\boldsymbol{(-\Delta\%)}$
Monomer Only	28075	-	-
2 Monomer → Dimer	52519	3631	6.47
4 Monomer → Tetramer	75266	12798	11.40
8 Monomer → Octamer	79096	34260	15.25
8 Monomer → Octamer (2 Tetramers)	99502	34260	17.22

Table 10.6: Calculated Properties from Osmotic Pressure Data

0.15 M Salt, pH 7.4	Local Molarity (M)	Ionic Strength Ratio α	Theoretical Saturation Limit (g/L)
NaF	0.116 ± 0.013	0.772 ± 0.086	589 ± 12
NaCl	0.103 ± 0.010	0.690 ± 0.068	543 ± 8
NaI	0.020 ± 0.022	0.131 ± 0.149	725 ± 25
NaSCN	0.056 ± 0.047	0.372 ± 0.313	700 ± 49

CHAPTER 11.

A DEVELOPED FREE-SOLVENT BASED MODEL FOR PROTEIN–PROTEIN BINDING PARAMETERS

11.1 Introduction

Macromolecular crowding occurs in solutions that contain a high concentration of proteins. These solutions of crowded protein are observed in cellular systems. The total concentration range of proteins in these systems is 50-400 gL^{-1}. These systems, as well as concentrated protein solutions in general, exhibit highly non-ideal osmotic pressure at concentrations near-saturation. If a system where ideal it would have a linear pressure-concentration relationship. The non-ideality seen in osmotic pressure data resembles an exponential growth relationship. While various models have attempted to capture these observed non-idealities, the free-solvent based model is the only model which also provides the mechanisms for the non-idealities.[19,20,68,69,73,77]

Protein-protein interactions are a fundamental part of almost all processes occurring in organisms. Information on protein-protein interactions play an integral role in the study of any enzyme, cellular receptor, or other large protein. A fundamental understanding of protein-protein interactions is critical for successful therapeutic drug development and design. When proteins bind there will be a change in SASA when compared to those two species individually. This change will be observable through an osmotic pressure shift. Here the free-solvent based model is developed to use the osmotic pressure change to determine protein-protein binding parameters.

11.2 Background

The free-solvent based model uses the concept of a hydrated macromolecule to correct for these non-idealities. A hydrated macromolecule is the protein containing bound water (protein

hydration) and bound salts. The free-solvent based model treats the hydrated protein as a separate macromolecule and all associated water and salt ions are absorbed in its definition. The remaining water and salts, not interacting with the hydrated species, are considered the "free" solvent.

SASA, first introduced by Lee and Richards, has tremendous applicability in terms of understanding free energies of solvent-protein interaction, protein-protein interactions, protein folding, and many other thermodynamic properties of the protein solution. By definition, the SASA is the accessible area on the surface of a protein available for water interaction. A protein's SASA can be altered through the binding of other proteins, ligands, and/or salts that take up accessible area that would otherwise have been available to water.

Protein-protein binding can have different effects on SASA. This can be seen as the addition of the protein$_1$ SASA ($SASA_{P1}$), protein$_2$ SASA ($SASA_{P2}$) and the subtraction of the change in SASA ($SASA_C$) that is due to the binding area SASA loss to obtain the protein-protein SASA ($SASA_{PC}$) shown as

$$SASA_{PC} = SASA_{P1} + SASA_{P2} - SASA_C. \tag{11.1}$$

The most likely form of change is an increase in SASA from the base protein SASA. A decrease in SASA is also possible. An example of this would be a protein with a concave bowl shape that is filled with a smaller protein. The SASA of the bowl concave is replaced flat top to the bowl with a smaller SASA.

The purpose of this study is to develop the free-solvent based model to account for protein-protein interactions and to be able to regress upon the association constant of the protein-protein binding.

11.3 Free Solvent Based Model for Protein-Protein Binding

The free-solvent based model has been described elsewhere for small ion interacting protein solutions with no protein-protein interactions.[19,20,68,69,77] The following is the development of a protein-protein interaction free-solvent based model which takes into account protein hydration and SASA changes between the protein$_1$, protein$_2$, and the protein-protein complex.

For a two-chamber osmometer, separated by a semi-permeable membrane, with protein solution contained within chamber II and solvent solution contained in chamber I, containing only diffusible species, the free-solvent based model can be expressed as the mole fraction of the two chambers. This mole fraction can describe the osmotic pressure, π, as

$$\pi = -\frac{RT}{\bar{V}_1}\ln\left\{\frac{\chi_1^{II}}{\chi_1^{I}}\right\}, \tag{11.2}$$

where χ_1^k is the mole fraction of the remaining unbound water in chamber *k*.

The mole fraction for each chamber can be described in terms of the moles of each species in that chamber. To account for protein-protein binding, four species must be taken into account, these consist of water, protein$_1$, protein$_2$, and protein-protein complex, labeled 1-4 respectively. The mole fraction of each chamber can therefore be reexpressed as

$$\chi_1^I = \frac{N_{1,E}^I}{N_{1,E}^I + N_{3,E}^I} \tag{11.3}$$

$$\chi_1^{II} = \frac{N_{1,E}^I}{N_{1,E}^I + N_{2,E}^I + N_{3,E}^I + N_{4,E}^I}, \tag{11.4}$$

where $N_{i,E}^k$ is the number of moles of species *i*, at equilibrium, *E*, in chamber *k*.

At equilibrium the free-solvent based model can be described as

$$\pi = \frac{RT}{\bar{V}_1}\ln\left\{\frac{N_{1,E}^I\left(N_{1,E}^{II} + N_{2,E}^{II} + N_{3,E}^{II} + N_{4,E}^{II}\right)}{N_{1,E}^{II}\left(N_{1,E}^I + N_{3,E}^I\right)}\right\}. \tag{11.5}$$

The protein-protein complex concentration variable, N_4, is correlated to the association constant. The association constant, K_A, is stated as

$$K_A = \frac{[AB]}{[A][B]}, \tag{11.6}$$

where $[A]$, $[B]$, and $[AB]$ are the concentration of protein$_1$, protein$_2$, and protein-protein complex, respectively. Solving for the concentration of protein-protein complex gives

$$[AB] = K_A[A][B]. \tag{11.7}$$

Assuming a liter of solution, the solution concentrations $[A]$, $[B]$, and $[AB]$ can be replaced with $N_{2,E}^{II}$, $N_{3,E}^{II}$, $N_{4,E}^{II}$, respectively, as in

$$N_{4,E}^{k} = K_A N_{2,E}^{k} N_{3,E}^{k}. \tag{11.8}$$

For the equilibrium concentration of water to be exchanged for the initial concentration, three interaction terms need to be taken into account, one for each other species the water interacts with as seen in

$$N_{1,E}^{k} = N_{1,I}^{k} - \nu_{12} N_{2,E}^{k} - \nu_{13} N_{3,E}^{k} - \nu_{14} N_{4,E}^{k}. \tag{11.9}$$

$N_{1,I}^{k}$ is the initial, I, number of moles of species i, in chamber k, and ν_{ij} is the binding of species i to species j.

In total, five interaction terms must be accounted for. Three interaction terms consist of water interacting with the other three species and two terms account for protein-protein complex interaction. Solvent-protein$_1$ interaction, ν_{12}, can be determined in experiments without the protein$_2$, by regressing on a simplified free-solvent based model.

$$\pi = \frac{RT}{\bar{V}_1} \ln\left\{\frac{N_{1,I}^{II} + (1-\nu_{12}) N_{2,E}^{II}}{(N_{1,I}^{II} - \nu_{12} N_{2,E}^{II})}\right\}, \tag{11.10}$$

where $N_{2,E}^{II}$ is equivalent to $N_{2,I}^{II}$ because there is no protein species lost to binding. Similarly, solvent-protein$_2$ interaction, ν_{13}, can be determined in experiments and regression without the protein.

$$\pi = \frac{RT}{\bar{V}_1} \ln\left\{\frac{(N_{1,I}^{I} - \nu_{13} N_{3,E}^{I})(N_{1,I}^{II} + (1-\nu_{13}) N_{3,E}^{II})}{(N_{1,I}^{I} + (1-\nu_{13}) N_{3,E}^{I})(N_{1,I}^{II} - \nu_{13} N_{3,E}^{II})}\right\}, \tag{11.11}$$

where $N_{3,E}^{II}$ is also equivalent to $N_{3,I}^{II}$ because there is no protein$_2$ species lost to binding.

As in the case of the protein$_1$ or protein$_2$, if a crystal structure exists, the remaining interaction parameter between solvent and protein-protein complex, ν_{14}, can be determined by molecular modeling. Another option is to use equation 10.1 and insert the solvent-protein$_1$ interaction value, ν_{12}, and the solvent-protein$_2$ interaction value, ν_{13}, to regress on the change in hydration between the protein and protein-protein complex. This can be stated as

$$\nu_{14} = \nu_{12} + \nu_{13} + \nu_C, \tag{11.12}$$

where ν_C is the change in hydration between protein$_1$ and protein-protein complex.

Using equations 10.5, 10.8, 10.9, 10.12, the free-solvent based model, in equation 10.5, can be given as

$$\pi = \frac{RT}{\bar{V}_1}\ln\left\{\frac{N_{1,E}^{I}\left(N_{1,I}^{II}+N_{2,E}^{II}(1-\nu_{12})+N_{3,E}^{II}(1-\nu_{13})+K_A N_{2,E}^{II}N_{3,E}^{II}\left(1-(\nu_{12}+\nu_{13}-\nu_C)\right)\right)}{\left(N_{1,E}^{I}+N_{3,E}^{I}\right)\left(N_{1,I}^{II}-\nu_{12}N_{2,E}^{II}-\nu_{13}N_{3,E}^{II}-K_A N_{2,E}^{II}N_{3,E}^{II}(\nu_{12}+\nu_{13}-\nu_C)\right)}\right\}. \tag{11.13}$$

The free-solvent based model, equation 10.13, can then be used to determine the association constant of protein-protein complex, K_A, as shown in

$$K_A = \frac{e^{\frac{\pi\bar{V}_1}{RT}}\left(N_{1,E}^{I}+N_{3,E}^{I}\right)\left(N_{1,I}^{II}-\nu_{12}N_{2,E}^{II}-\nu_{13}N_{3,E}^{II}\right)-N_{1,E}^{I}\left(N_{1,I}^{II}+N_{2,E}^{II}(1-\nu_{12})+N_{3,E}^{II}(1-\nu_{13})\right)}{N_{2,E}^{II}N_{3,E}^{II}\left(e^{\frac{\pi\bar{V}_1}{RT}}\left(N_{1,E}^{I}+N_{3,E}^{I}\right)(\nu_{12}+\nu_{13}-\nu_C)+N_{1,E}^{I}\left(1-(\nu_{12}+\nu_{13}-\nu_C)\right)\right)}. \tag{11.14}$$

Equation 10.13 can also be used to determine the hydration change from the protein$_1$ and protein$_2$ to the protein-protein complex, ν_C, as shown in

$$\nu_C = \frac{e^{\frac{\pi\bar{V}_1}{RT}}\left(N_{1,E}^{I}+N_{3,E}^{I}\right)\left(N_{1,I}^{II}-\nu_{12}N_{2,E}^{II}-\nu_{13}N_{3,E}^{II}-K_A N_{2,E}^{II}N_{3,E}^{II}(\nu_{12}+\nu_{13})\right)-N_{1,E}^{I}\left(N_{1,I}^{II}+N_{2,E}^{II}(1-\nu_{12})+N_{3,E}^{II}(1-\nu_{13})+K_A N_{2,E}^{II}N_{3,E}^{II}(1+\nu_{12}+\nu_{13})\right)}{K_A N_{2,E}^{II}N_{3,E}^{II}\left(e^{\frac{\pi\bar{V}_1}{RT}}\left(N_{1,E}^{I}+N_{3,E}^{I}\right)-N_{1,E}^{I}\right)}. \tag{11.15}$$

The three interactions terms and the six species concentrations give a total of nine parameters in the developed protein-protein free-solvent based model. All of which are comprised of only physically-relevant and measureable parameters.

11.4. Discussion

Here, we determine theoretical osmotic pressure values for a Trypsin and Trypsin Inhibitor system using the free-solvent based model. The hydrations for Trypsin, Trypsin-Inhibitor and the complex are 1.102, 1.731, and 1.101 g H_2O per g Species, respectively. The hydrations were determined by converting Chimera determined SASAs.[10,92,99,100] The association constant is 1.67 x 10^{13} M.[101] These inputs allow for the osmotic pressure to be determined at different concentrations as seen in Figure 1.

At high concentrations small property changes can cause large differences in osmotic pressures. The sensitivity of FSM with respect to the hydration change between protein$_1$, protein$_2$ and the protein-protein complex is done by determining the derivative of complex hydration change, ν_C.

$$\frac{d\pi}{d\nu_C} = \frac{RT}{\bar{V}_1}\frac{H_1}{H_2 H_3} \tag{11.16}$$

$$H_1 = -K_A N_{2,E}^{II} N_{3,E}^{II}\left(N_{2,E}^{II} + N_{3,E}^{II} + K_A N_{2,E}^{II} N_{3,E}^{II}\right) \tag{11.17}$$

$$H_2 = N_{1,I}^{II} - N_{3,E}^{II}\nu_{13} - N_{2,E}^{II}\left(\nu_{12} + K_A N_{3,E}^{II}\nu_{12} + K_A N_{3,E}^{II}(\nu_{13} - \nu_C)\right) \tag{11.18}$$

$$H_3 = N_{1,I}^{II} - N_{3,E}^{II}(1 - \nu_{13}) + N_{2,E}^{II}\left(1 - \nu_{12} + K_A N_{3,E}^{II}\left(1 - \nu_{12} - \nu_{13} - \nu_C\right)\right) \tag{11.19}$$

In the case of 1:1 Trypsin, and Trypsin-Inhibitor, at 500 gL^{-1}, a 0.003 mol/mol change in complex hydration, corresponding to a 5 water binding shift, causes a 1 psi change in osmotic pressure. For reference, the change between the protein$_1$ and protein$_2$ to the complex is 228 water molecules.

It is also important to look at the sensitivity of the FSM in terms of the association constant, K_A, and can be determined by

$$\frac{d\pi}{dK_A} = \frac{RT}{\bar{V}_1}\frac{K_1}{K_2 K_3} \tag{11.20}$$

$$K_1 = N_{2,E}^{II} N_{3,E}^{II} \left(N_{1,I}^{II} + N_{2,E}^{II} (\nu_{13} - \nu_C) + N_{3,E}^{II} (\nu_{12} - \nu_C) \right) \tag{11.21}$$

$$K_2 = N_{1,I}^{II} + N_{2,E}^{II} (1 - \nu_{13}) + N_{3,E}^{II} (1 - \nu_{12}) + K_A N_{2,E}^{II} N_{3,E}^{II} \left(1 - \nu_{12} - \nu_{13} - \nu_C\right) \tag{11.22}$$

$$K_3 = N_{1,I}^{II} - N_{2,E}^{II} \nu_{13} - N_{3,E}^{II} \nu_{12} - K_A N_{2,E}^{II} N_{3,E}^{II} \left(\nu_{12} + \nu_{13} - \nu_C\right) \tag{11.23}$$

For Trypsin, and Trypsin-Inhibitor, at 500 gL^{-1}, a 1 x 10^5 change in the association constant caused a 1 psi change in osmotic pressure.

11.5. Conclusion

The free-solvent based model is an excellent predictor of osmotic pressure for single protein-ion solutions. Here, a generalized free-solvent based model for protein-protein solutions has been developed. Given that physical parameters are available, experimental data for protein-protein solution no longer needs to be obtained because the free-solvent based model provides excellent predictability. When used with osmotic pressure data the association constant of a protein-protein solution can be determined. It is also possible to obtain the change in hydration between protein and protein-protein complexes, using the free-solvent based model.

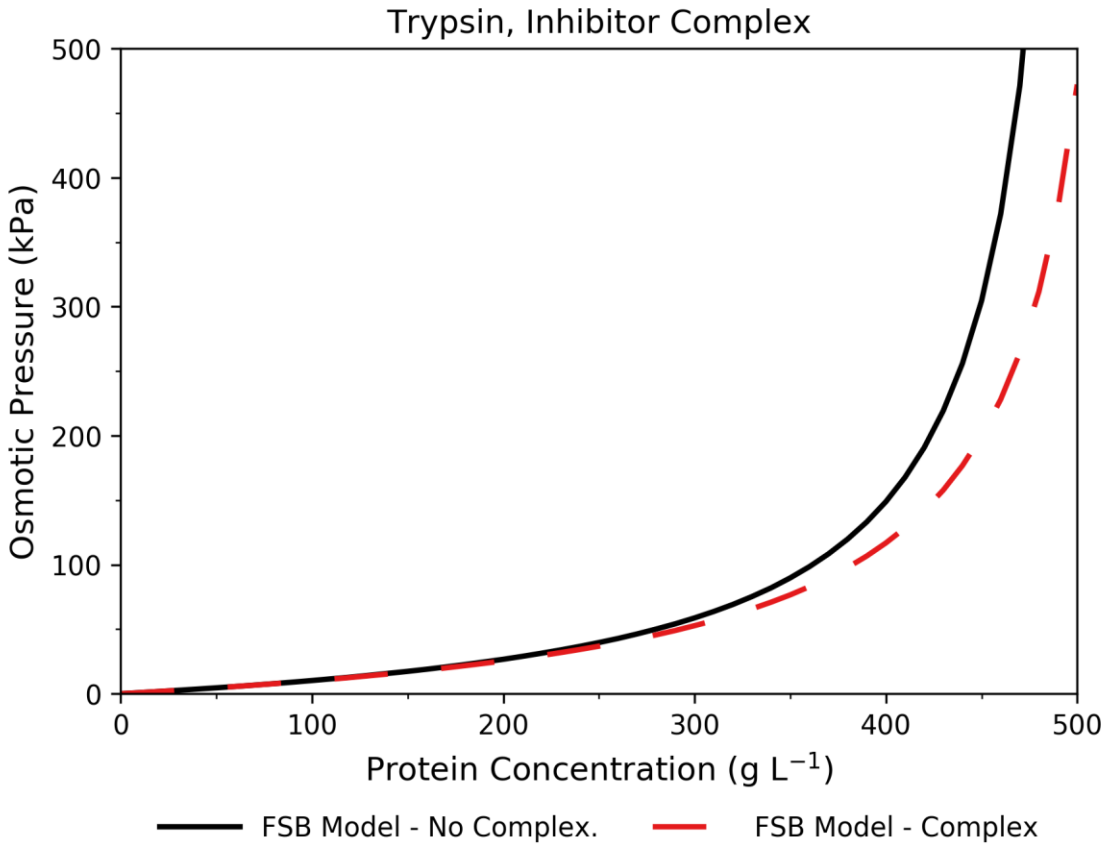


Figure 11.1: Fittings for Trypsin and Trypsin Inhibitor if in a complex and if not in a complex.

CHAPTER 12.

CONCLUSIONS

12.1. Findings in This Work

Herein, a concentrating osmometer was developed and shown to provide concentrated osmotic pressure results similar to literature values in less time and with less protein. This was possible due to the design's ability to concentrate a solution by 18x. Additionally, the free-solvent based model was developed for ion hydration and for the determination of association values in protein-protein complexes.

Analysis of concentrated osmotic pressure for various sodium salts showed large variations in osmotic pressure. Also, that smaller ion with large charge densities, like F^-, and Cl^-, bound to proteins in greater numbers and developed systems with higher protein hydrations. Conversely, larger ions with weaker charge densities, like I^- and SCN^-, had lower ion binding and systems with lower protein hydrations.

12.2. Future Directions

The free-solvent based model has been shown to accurately fit osmotic pressure data for solutions with moderate salt concentrations. However, it still needs to be developed to allow for analysis of osmotic pressure data at low salt concentrations or salt free systems (Figure 12.1). One potential explanation driving the model inaccuracy is an assumption that proteins in the concentrated region are monomers. As shown in chapter 10 (Table 10.5), when monomers bind to each other to create multimers there is a significant reduction in solvent accessible surface area (SASA). This reduction in SASA would have a great effect on both hydration and potentially ion binding, depending where the binding occurred. As can be seen in McBride et al. (2013)[102], there

are significant differences between the hydration and ion binding of Bovine Serum Albumin (BSA) at different pHs. Additionally, in Chapter 10 there are significant differences in the hydration and ion binding of BSA solutions in different sodium solvents. Circular Dichosim Analysis of BSA in different salt solutions (Figure 12.2) and at different pHs (Figure 12.2) show that the structure of BSA is not significantly effected at pHs 5.4, 7.0, and 7.4. Also, that NaCl or only water in the system does not cause BSA to denature. The lack of structural changes to BSA in these systems, while still having significant changes in hydration and ion binding, lends credence to the possibility that multimerization of the protein is causing a concentration dependent effect on hydration and ion binding.

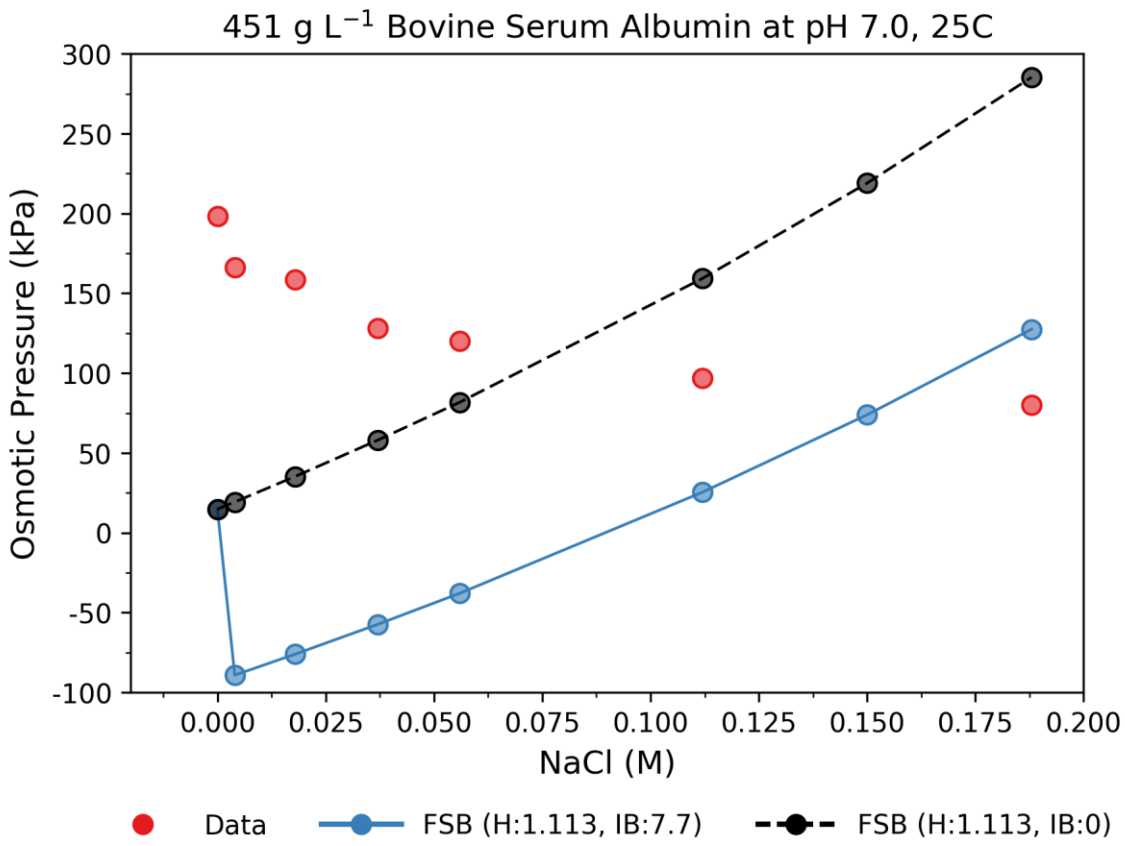


Figure 12.1: Osmotic pressure data for Bovine Serum Albumin at various concentrations of NaCl. Two free-solvent based (FSB) model fittings are shown. Fitting values are regressed upon Bovine Serum Albumin data at pH 7.0.[102] The second FSB Fit has been forced to a protein-ion binding of zero.

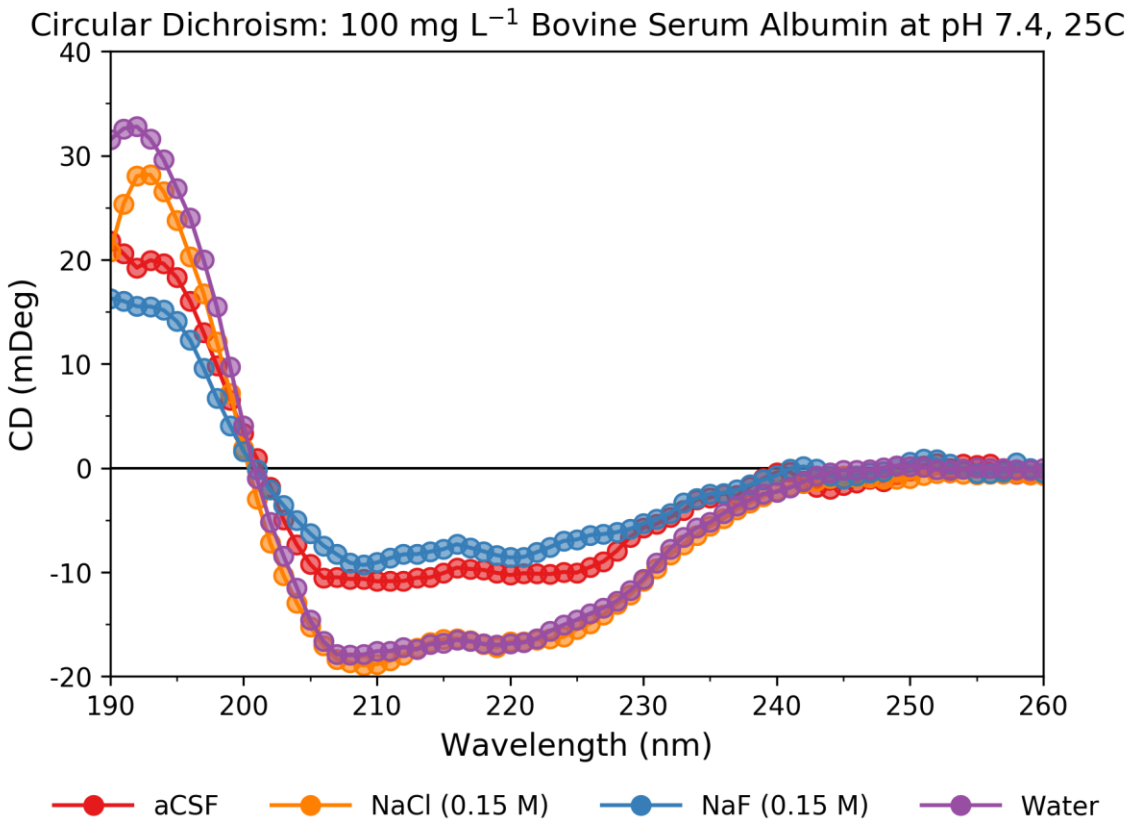


Figure 12.2: Circular Dichroism of Bovine Serum Albumin (BSA) in various salts. NaF and artificial Cerebral Spinal Fluid (aCSF) show BSA denaturation.

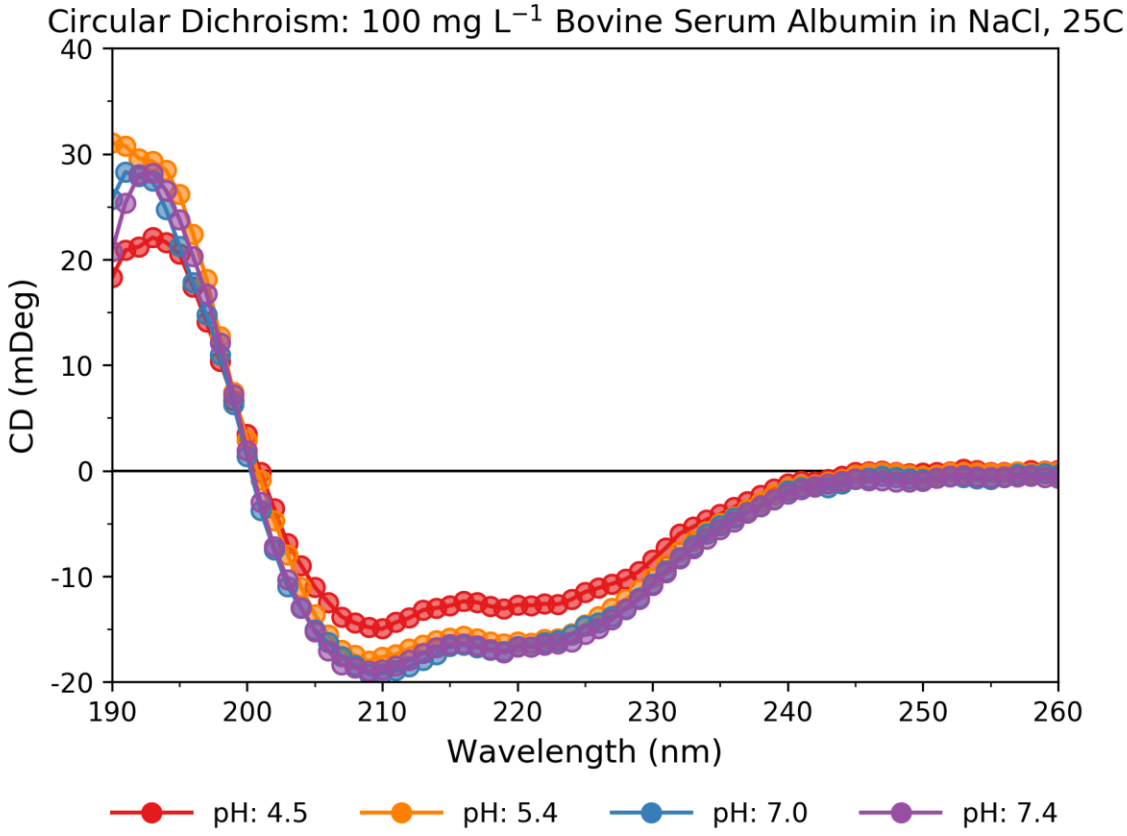


Figure 12.3: Circular Dichroism of Bovine Serum Albumin (BSA) in NaCl at various pHs. pH 4.5 shows BSA denaturation.

APPENDIX A.

SUPPLEMENTAL DATA

Table A.1. Bovine Serum Albumin in Artificial Cerebral Spinal Fluid at pH 7.4, 25°C

[BSA] (gL^{-1} Soln)	Osmotic Pressure (kPa)
290	99.9
298	77.5
306	87.8
341	134.1
359	135.4
361	130.2
362	143.1
378	176.0
396	196.9
399	246.2
408	141.4
414	243.4
416	264.0
436	390.1

Table A.2. 451 gL^{-1} Bovine Serum Albumin in Various Sodium Chloride Molarities at pH 7.0, 25°C

[NaCl] (M Soln)	Osmotic Pressure (kPa)
0.000	198.4
0.004	166.4
0.018	158.6
0.037	128.3
0.056	120.0
0.112	96.7
0.188	80.1

Table A.3. Circular Dichroism: 100 mg L^{-1} Bovine Serum Albumin at pH 7.0, 25°C

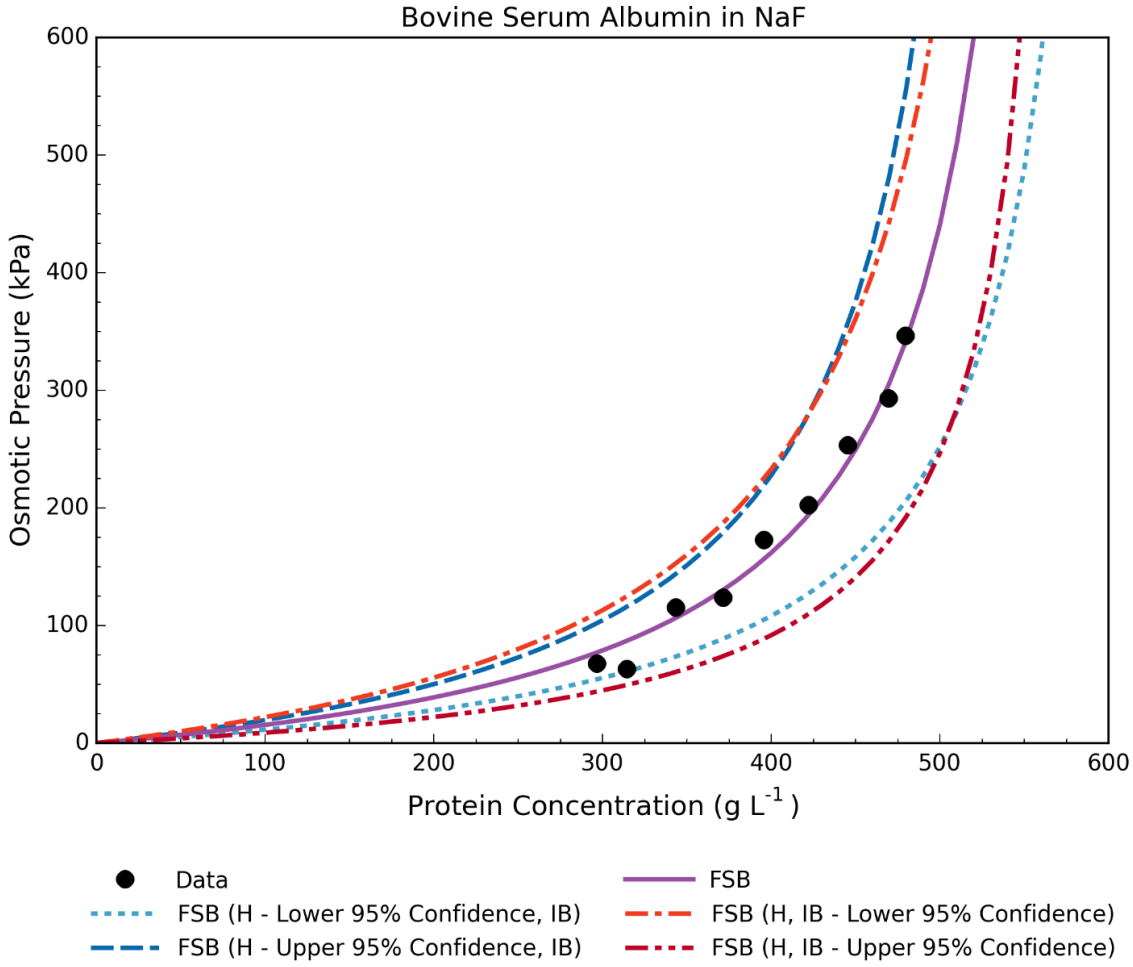
Wavelength (nm)	Water	NaF	aCSF	NaCl (pH: 4.5)	NaCl (pH: 5.4)	NaCl (pH: 7)	NaCl (pH: 7.4)
260	0.0	-0.5	-0.6	0.0	-0.1	-0.4	-0.7
259	-0.2	0.0	-0.6	0.0	-0.1	-0.3	-0.6
258	-0.2	0.5	-0.6	0.0	-0.2	-0.4	-0.5
257	0.0	-0.4	-0.3	-0.2	-0.1	-0.4	-0.6
256	0.0	-0.5	0.4	-0.2	-0.2	-0.6	-0.6
255	0.0	-0.5	0.2	-0.1	-0.2	-0.8	-0.6
254	-0.1	-0.1	0.4	0.0	-0.1	-0.7	-0.6
253	-0.2	0.4	0.0	0.1	-0.2	-0.6	-0.5
252	-0.2	0.8	0.4	-0.1	-0.2	-0.6	-0.5
251	0.1	0.9	0.2	-0.1	-0.4	-0.7	-0.7
250	0.2	0.6	-0.3	-0.2	-0.5	-0.8	-1.0
249	0.2	0.2	-0.6	-0.3	-0.4	-0.8	-1.1
248	0.0	-0.3	-1.3	-0.1	-0.2	-0.7	-1.1
247	-0.2	-0.4	-1.1	0.0	-0.1	-0.5	-0.9
246	-0.2	-0.7	-1.4	-0.1	-0.2	-0.7	-0.9
245	-0.2	-1.0	-1.7	-0.1	-0.4	-0.8	-0.8
244	-0.4	-0.7	-2.0	-0.5	-0.8	-1.3	-1.0
243	-0.7	-0.1	-1.8	-0.8	-1.1	-1.6	-1.2
242	-1.3	0.1	-1.2	-1.0	-1.4	-1.5	-1.5
241	-1.9	-0.1	-0.4	-1.0	-1.4	-1.6	-1.8
240	-2.3	-0.7	-0.4	-1.2	-1.6	-1.9	-2.2
239	-2.5	-1.1	-1.0	-1.7	-2.3	-2.8	-2.7
238	-3.0	-1.6	-1.8	-2.3	-3.0	-3.4	-3.3
237	-3.6	-2.1	-2.6	-2.9	-3.7	-3.9	-4.0
236	-4.3	-2.4	-2.5	-3.5	-4.4	-4.5	-4.9
235	-5.2	-2.5	-2.8	-4.1	-5.0	-5.2	-5.6
234	-5.7	-2.9	-3.0	-4.6	-5.7	-6.0	-6.4
233	-6.6	-3.3	-4.1	-5.3	-6.7	-7.1	-7.3
232	-7.8	-4.4	-4.7	-6.0	-7.6	-8.2	-8.3
231	-9.1	-4.9	-5.4	-7.3	-8.9	-9.5	-9.7
230	-10.6	-5.3	-5.7	-8.4	-10.1	-10.7	-10.8
229	-11.8	-5.9	-6.6	-9.6	-11.3	-12.1	-12.2
228	-12.8	-6.2	-8.0	-10.3	-12.2	-13.1	-13.1
227	-13.4	-6.3	-9.0	-10.7	-13.0	-13.8	-14.1
226	-14.0	-6.4	-9.5	-11.1	-13.8	-14.4	-14.9
225	-14.6	-6.9	-10.0	-11.5	-14.6	-14.7	-15.5
224	-15.1	-7.0	-10.0	-12.2	-15.4	-15.6	-16.2
223	-15.6	-7.6	-10.2	-12.6	-15.9	-16.1	-16.4
222	-16.4	-8.1	-10.2	-12.6	-16.0	-16.3	-16.5
221	-16.8	-8.5	-10.1	-12.8	-16.3	-16.7	-16.7
220	-16.9	-8.6	-10.3	-12.7	-16.2	-16.6	-16.7
219	-17.0	-8.4	-10.1	-13.1	-16.4	-17.1	-17.2
218	-16.9	-8.0	-9.8	-12.8	-16.2	-17.0	-16.9
217	-16.7	-7.7	-9.7	-12.5	-15.9	-16.7	-16.6
216	-16.5	-7.3	-9.6	-12.4	-15.7	-16.5	-16.4
215	-16.8	-7.8	-10.1	-12.8	-15.9	-16.6	-16.5
214	-16.9	-8.0	-10.5	-13.0	-16.1	-17.4	-16.8
213	-17.4	-8.3	-10.5	-13.2	-16.5	-18.0	-17.3
212	-17.2	-8.3	-10.8	-13.9	-16.8	-18.5	-18.0

211	-17.6	-8.6	-10.9	-14.4	-17.4	-18.9	-18.5
210	-17.6	-9.1	-10.9	-15.0	-17.6	-19.2	-18.8
209	-17.9	-9.3	-10.7	-14.8	-18.0	-19.3	-19.0
208	-18.0	-9.1	-10.7	-14.4	-17.5	-18.4	-18.7
207	-17.9	-8.3	-10.5	-13.8	-17.0	-17.6	-18.4
206	-16.6	-7.5	-10.5	-12.5	-15.6	-16.3	-17.1
205	-14.6	-6.3	-9.2	-11.0	-13.6	-15.1	-15.2
204	-11.5	-5.0	-7.4	-9.0	-11.0	-13.0	-13.0
203	-8.4	-3.6	-5.0	-6.9	-7.9	-11.0	-10.3
202	-5.2	-2.1	-1.8	-3.6	-4.7	-7.5	-7.2
201	-1.0	-0.1	0.9	-0.2	-0.9	-3.8	-3.0
200	4.1	1.6	3.4	3.4	2.9	1.4	1.9
199	9.7	4.0	6.5	6.6	7.4	6.3	7.2
198	15.5	6.7	9.8	10.4	12.7	11.0	12.1
197	20.0	9.6	13.0	14.1	18.1	14.8	16.7
196	24.0	12.3	16.0	17.4	22.4	17.8	20.3
195	26.8	14.1	18.3	20.5	26.2	21.3	23.8
194	29.6	15.2	19.6	21.6	28.5	24.7	26.5
193	31.6	15.5	20.0	22.1	29.3	27.5	28.2
192	32.8	15.5	19.2	21.2	29.6	27.9	28.0
191	32.6	16.0	20.6	20.9	30.7	28.3	25.4
190	31.5	16.3	21.8	18.3	31.1	25.7	20.8

APPENDIX B.

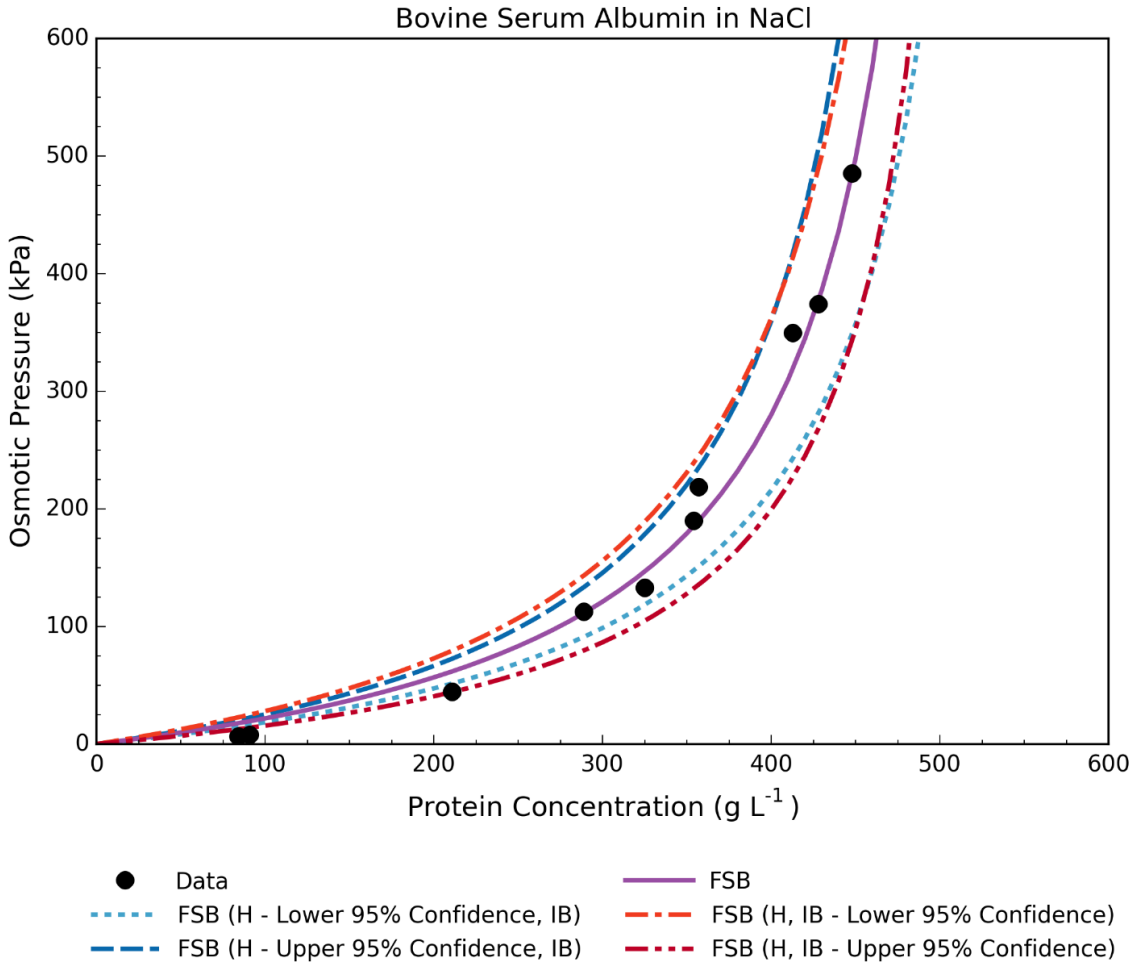
CHAPTER 10 SUPPLEMENTAL

Figure B.1. Confidence Interval Curve: Bovine Serum Albumin in 0.15M NaF at pH 7.4 and 25°C



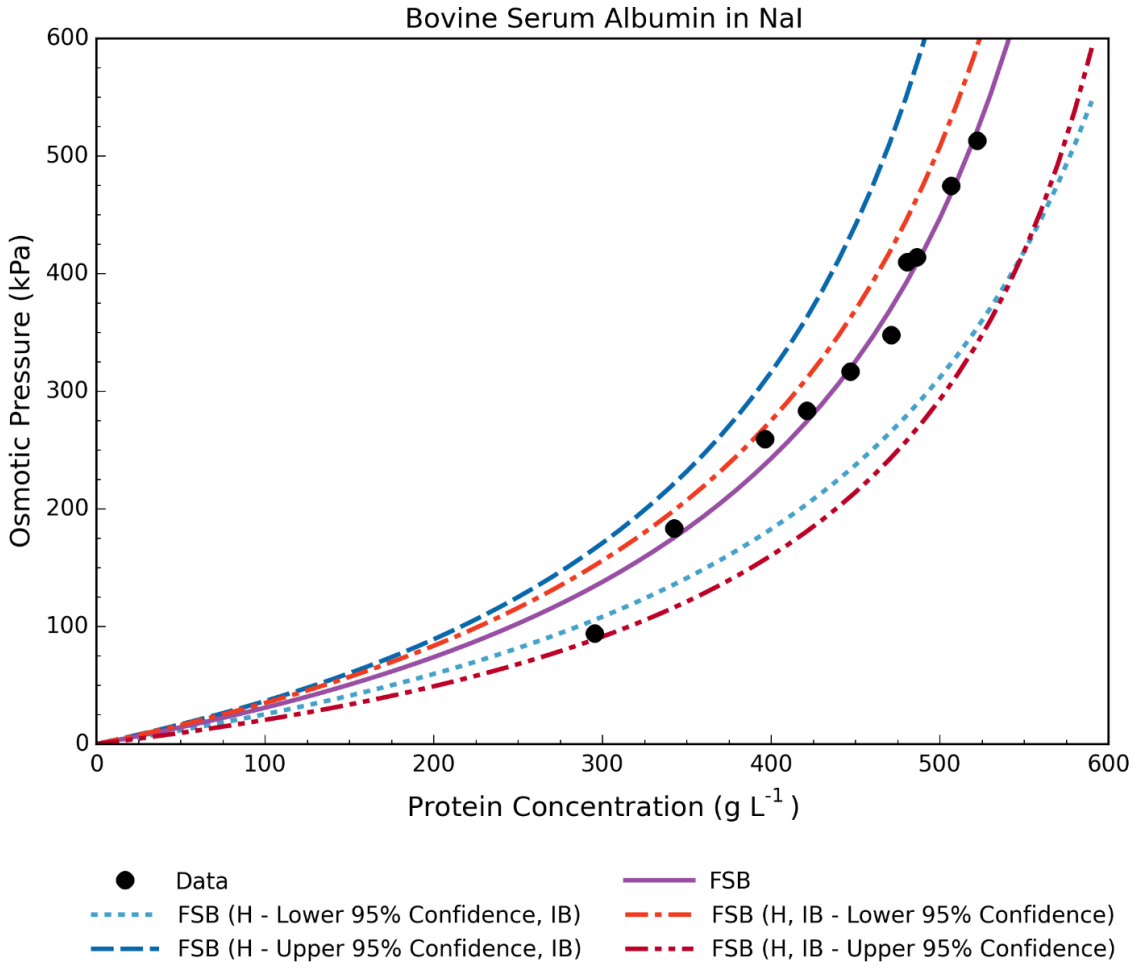
H: hydration value parameter adjustment IB: ion binding parameter adjustment

Figure B.2. Confidence Interval Curve: Bovine Serum Albumin in 0.15M NaCl at pH 7.4 and 25°C



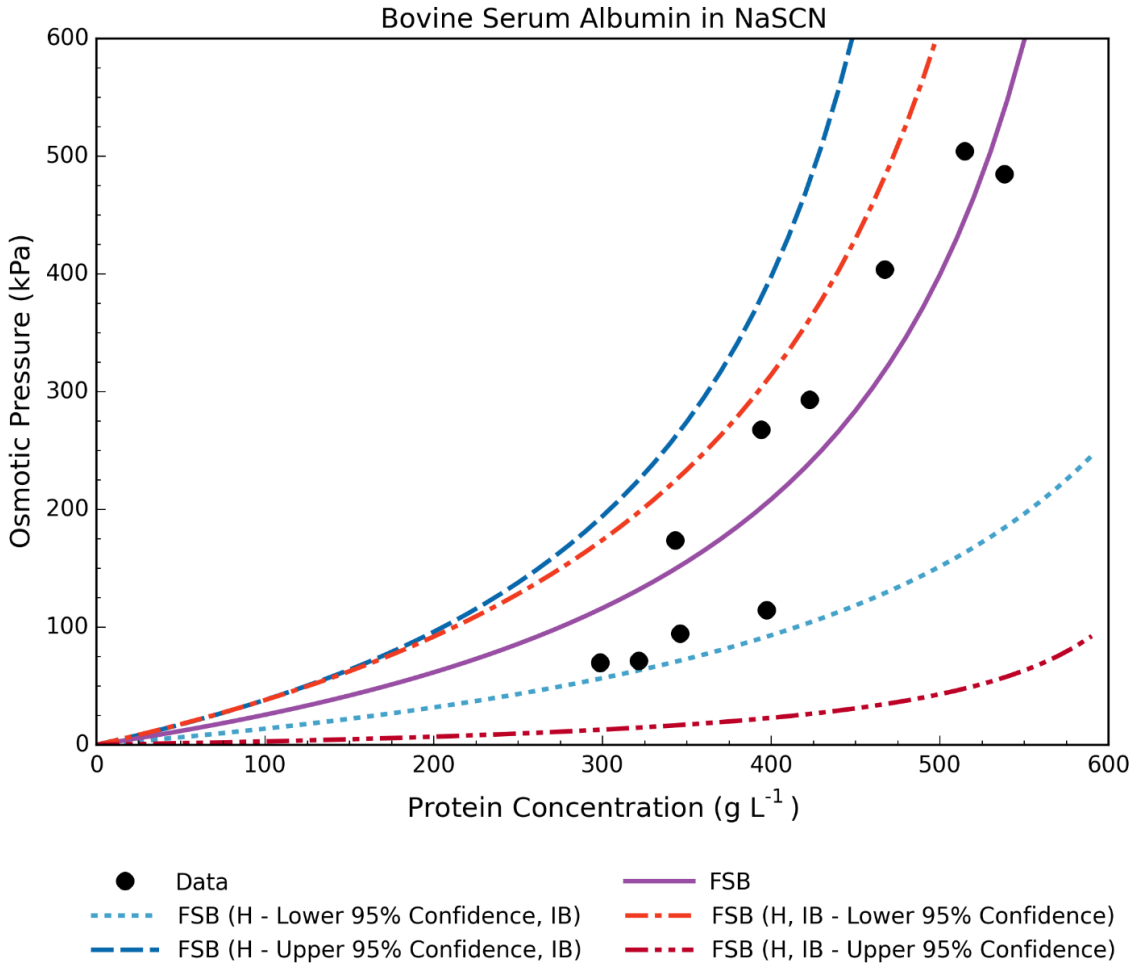
H: hydration value parameter adjustment IB: ion binding parameter adjustment

Figure B.3. Confidence Interval Curve: Bovine Serum Albumin in 0.15M NaI at pH 7.4 and 25°C



H: hydration value parameter adjustment IB: ion binding parameter adjustment

Figure B.4. Confidence Interval Curve: Bovine Serum Albumin in 0.15M NaSCN at pH 7.4 and 25°C



H: hydration value parameter adjustment IB: ion binding parameter adjustment

Table B.1. Parameter Sensitivity for Bovine Serum Albumin in Sodium Salts
(Changes in parameters required for 1 kPa pressure change in osmotic pressure at 400 gL^{-1})

0.15 M Salt, pH 7.4	Ion Binding $\left(\frac{\text{mol Salt}}{\text{mol BSA}}\right)$	Hydration $\left(\frac{\text{g } H_2O}{\text{g BSA}}\right)$
NaF	-0.132	0.009
	0.132	-0.010
NaCl	-0.109	0.007
	0.109	-0.008
NaCl (Scatchard)	-0.103	0.006
	0.103	-0.006
NaI	-0.182	0.011
	0.183	-0.012
NaSCN	-0.172	0.011
	0.172	-0.012

APPENDIX C.

CHAPTER 11 SUPPLEMENTAL

Determining Initial Concentrations

In order for the free-solvent model to contain only be comprised of only experimentally measureable parameters, concentrations need to be expressed in terms of the experimentally determinable initial concentration, rather than experimentally indeterminable equilibrium concentration. The mole fractions, equations 10.3, 10.4, can therefore be stated as

$$\chi_1^I = \frac{N_{1,I}^I - \nu_{13} N_{3,I}^I}{N_{1,I}^I + N_{3,I}^I(1-\nu_{13})} \tag{E.24}$$

$$\chi_1^{II} = \frac{N_{1,I}^{II} - \nu_{12} N_{2,E}^{II} - \nu_{13} N_{3,E}^{II} - \nu_{14} N_{4,E}^{II}}{N_{1,I}^I + N_{2,E}^{II}(1-\nu_{12}) + N_{3,E}^{II}(1-\nu_{13}) + N_{4,E}^{II}(1-\nu_{14})}, \tag{E.25}$$

where the mole of protein$_2$ in chamber I is equivalent, initially and at equilibrium.

The free-solvent model can then be expressed as

$$\pi = \frac{RT}{\bar{V}_1} \ln\left\{\frac{(N_{1,I}^I - \nu_{13} N_{3,I}^I)\left(N_{1,I}^{II} + N_{2,E}^{II}(1-\nu_{12}) + N_{3,E}^{II}(1-\nu_{13}) + N_{4,E}^{II}(1-\nu_{14})\right)}{\left(N_{1,I}^I + N_{3,I}^I(1-\nu_{13})\right)\left(N_{1,I}^{II} - \nu_{12} N_{2,E}^{II} - \nu_{13} N_{3,E}^{II} - \nu_{14} N_{4,E}^{II}\right)}\right\}. \tag{E.26}$$

Protein$_2$ in chamber II binding to protein$_1$ will cause a decrease in the molarity of protein$_2$ in chamber II. This will cause a flux of protein$_2$ between the chambers to reach a new free protein$_2$ equilibrium concentration. The protein$_2$ flux also causes a flux of water in the opposite direction. Both of these fluxes need to be accounted for in the free-solvent model as shown in

$$\pi = \frac{RT}{\bar{V}_1} \ln\left\{\frac{\left(N_{1,I}^I - \nu_{13} N_{3,FE}^I + N_{1,FA}^I\right)\left(N_{1,I}^{II} + N_{2,E}^{II}(1-\nu_{12}) + N_{3,E}^{II}(1-\nu_{13}) + N_{4,E}^{II}(1-\nu_{14}) + N_{1,FA}^I\right)}{\left(N_{1,I}^I + N_{3,\mathrm{FE}}^I(1-\nu_{13}) + N_{1,FA}^I\right)\left(N_{1,I}^{II} - \nu_{12} N_{2,E}^{II} - \nu_{13} N_{3,\mathrm{FE}}^{II} - \nu_{14} N_{4,E}^{II} + N_{1,FA}^I\right)}\right\}, \tag{E.27}$$

where the new protein$_2$ flux equilibrium, $N_{3,FE}^k$, can be shown as

$$N_{3,FE}^k = \alpha\left(N_{1,E}^k\right) \frac{N_{3,\mathrm{E}}^I + N_{3,E}^{II}}{\alpha(N_{1,\mathrm{E}}^I + N_{1,E}^{II}) + \beta\left(N_{3,\mathrm{E}}^I + N_{3,E}^{II}\right)}, \tag{E.28}$$

where α and β are the conversions of moles of water/protein$_2$ to liters of water/protein$_2$, respectively. The water flux adjustment can be described as

$$N_{1,FE}^{k} = \chi\left(N_{1,E}^{k} - N_{3,FE}^{k}\right), \tag{E.29}$$

where χ is the conversion from moles of protein$_2$ to moles of water.

In order for $N_{4,E}^{II}$ to be put into terms of $N_{2,E}^{II}$, and $N_{3,E}^{II}$, two interaction terms need to be used. These two terms account for the number of proteins binding in the protein-protein complex, υ_{24}, and conversely the number of protein$_2$ binding in the protein-protein complex, υ_{34}. A number of techniques including mass-spectrometry, x-ray crystallography, and HPLC analysis can be used to determine the quantities of protein$_1$ and protein$_2$ in the protein-protein complex.

The concentration, at equilibrium, of protein$_1$ and protein$_2$ in terms of the protein-protein complex concentration can be stated as

$$N_{2,E}^{k} = N_{2,I}^{k} - K_A N_{2,E}^{k} N_{3,E}^{k} \upsilon_{24} \tag{E.30}$$

$$N_{3,E}^{k} = N_{3,I}^{k} - K_A N_{2,E}^{k} N_{3,E}^{k} \upsilon_{34}. \tag{E.31}$$

Equations 10.8, E.30, and E.31 in terms of only initial concentration values, can be expressed as

$$N_{2,E}^{k} = \pm\sqrt{\frac{N_{2,I}^{k}}{K_A\upsilon_{34}} + \left(\frac{1}{2}\left(\frac{1}{K_A\upsilon_{34}} - N_{2,I}^{k} + \frac{N_{3,I}^{k}\upsilon_{24}}{\upsilon_{34}}\right)\right)^2} - \frac{1}{2}\left(\frac{1}{K_A\upsilon_{34}} - N_{2,I}^{k} + \frac{N_{3,I}^{k}\upsilon_{24}}{\upsilon_{34}}\right) \tag{E.32}$$

$$N_{3,E}^{k} = \pm\sqrt{\frac{N_{3,I}^{k}}{K_A\upsilon_{24}} + \left(\frac{1}{2}\left(\frac{1}{K_A\upsilon_{24}} - N_{3,I}^{k} + \frac{N_{2,I}^{k}\upsilon_{34}}{\upsilon_{24}}\right)\right)^2} - \frac{1}{2}\left(\frac{1}{K_A\upsilon_{24}} - N_{3,I}^{k} + \frac{N_{2,I}^{k}\upsilon_{34}}{\upsilon_{24}}\right) \tag{E.33}$$

$$N_{4,E}^{k} = \frac{1}{\upsilon_{24}}\left(N_{2,I}^{k} - \left(\pm\sqrt{\frac{N_{2,I}^{k}}{K_A\upsilon_{34}} + \left(\frac{1}{2}\left(\frac{1}{K_A\upsilon_{34}} - N_{2,I}^{k} + \frac{N_{3,I}^{k}\upsilon_{24}}{\upsilon_{34}}\right)\right)^2} - \frac{1}{2}\left(\frac{1}{K_A\upsilon_{34}} - N_{2,I}^{k} + \frac{N_{3,I}^{k}\upsilon_{24}}{\upsilon_{34}}\right)\right)\right). \tag{E.34}$$

Combining equations E.27, E.28, E.29, E.32, E.33, and E.34 sets up the free solvent model in terms of all initial concentrations.

APPENDIX D.

DETAILED DESIGNS OF AN OSMOTIC TRANSPORT DEVICE

FOR SPINAL CORD INJURY

Figure D.1. Detailed Design of Osmotic Transport Device

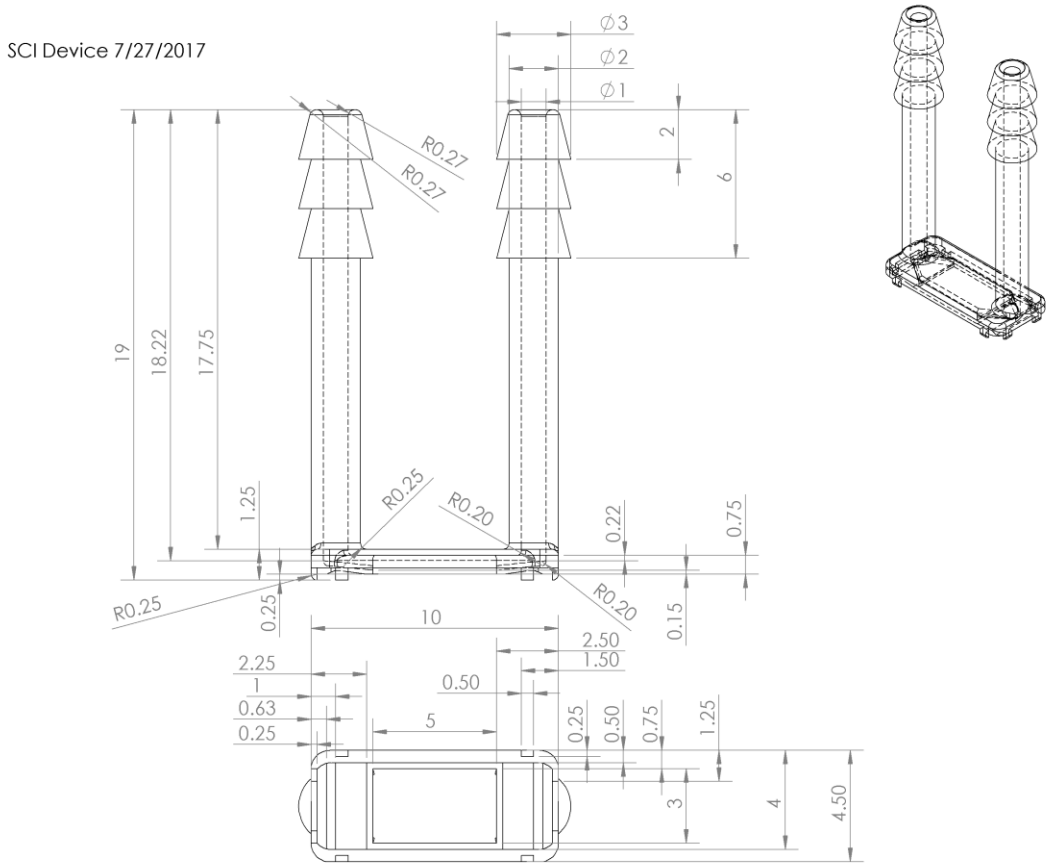


Figure D.2. Detailed Design of Osmotic Transport Device: Mold Top

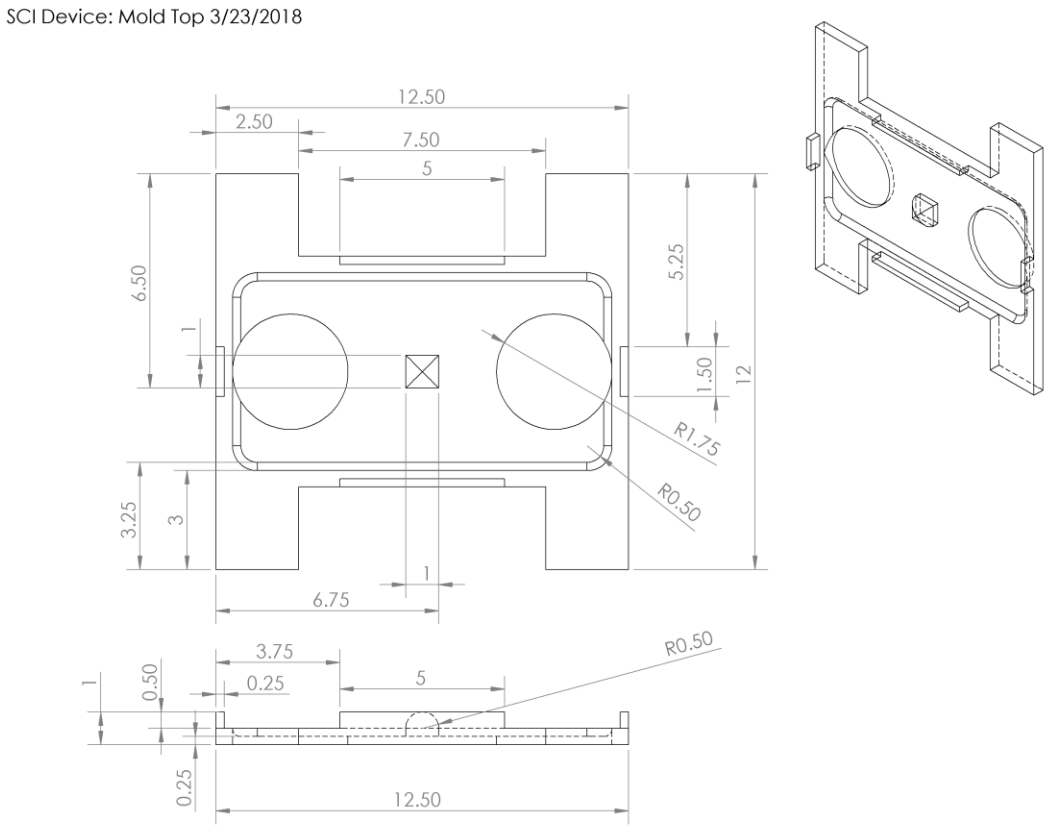


Figure D.3. Detailed Design of Osmotic Transport Device: Mold Middle

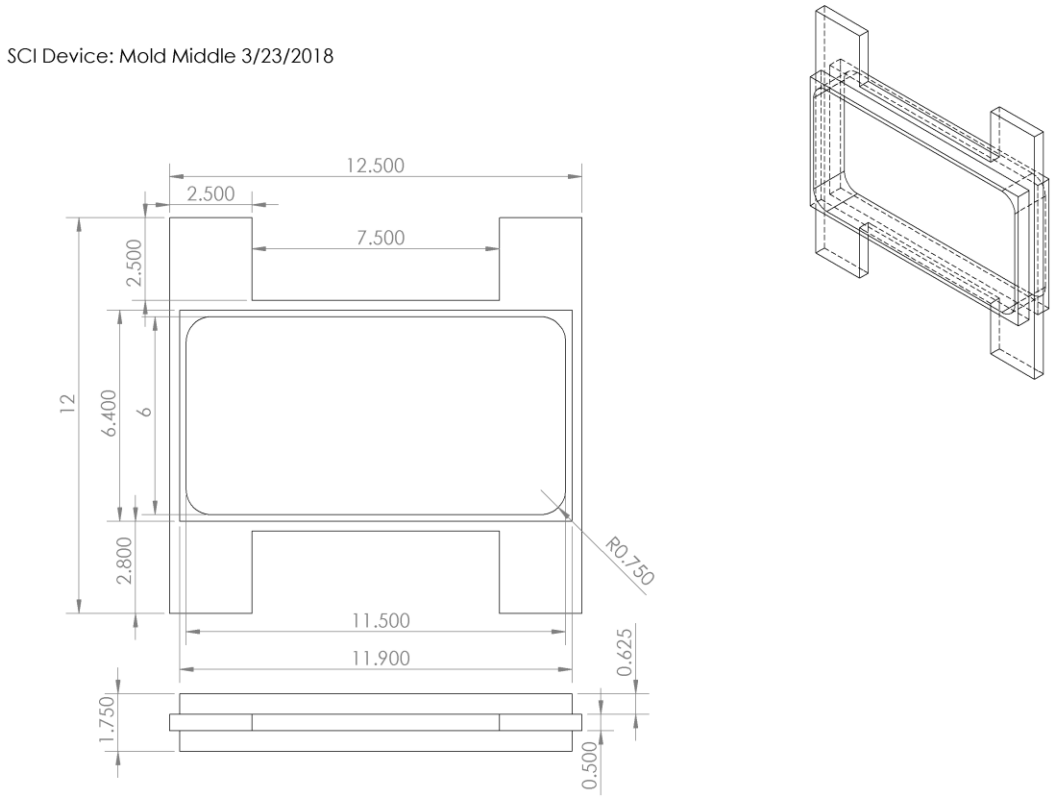
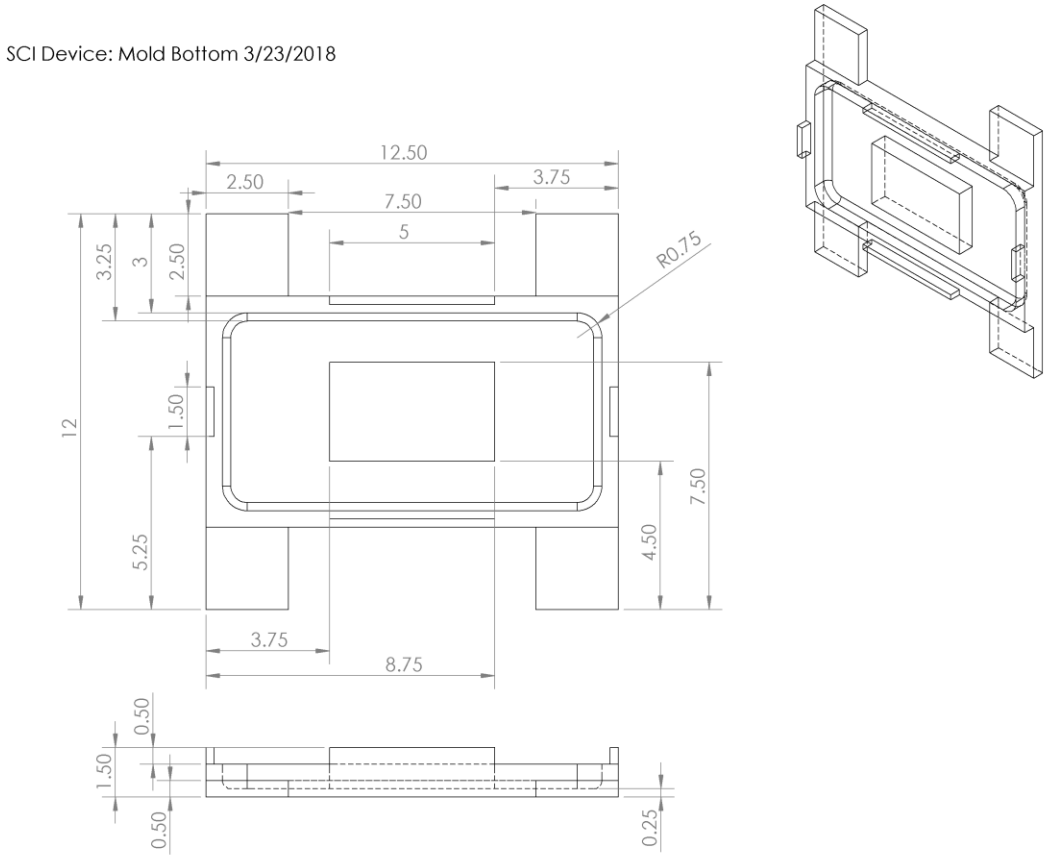


Figure D.4. Detailed Design of Osmotic Transport Device: Mold Bottom



APPENDIX E.

DETAILED DESIGN OF CONCENTRATING OSMOMETER

Figure E.1. Detailed Design of Concentrating Osmometer: Plunger Top

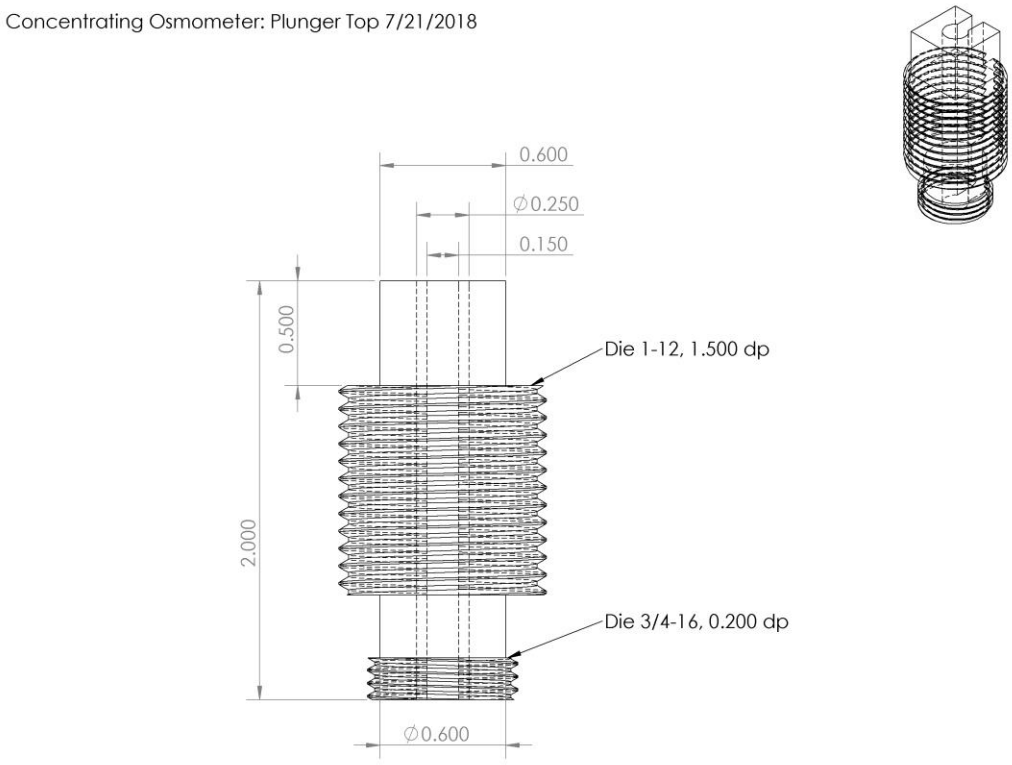


Figure E.2. Detailed Design of Concentrating Osmometer: Plunger Bottom

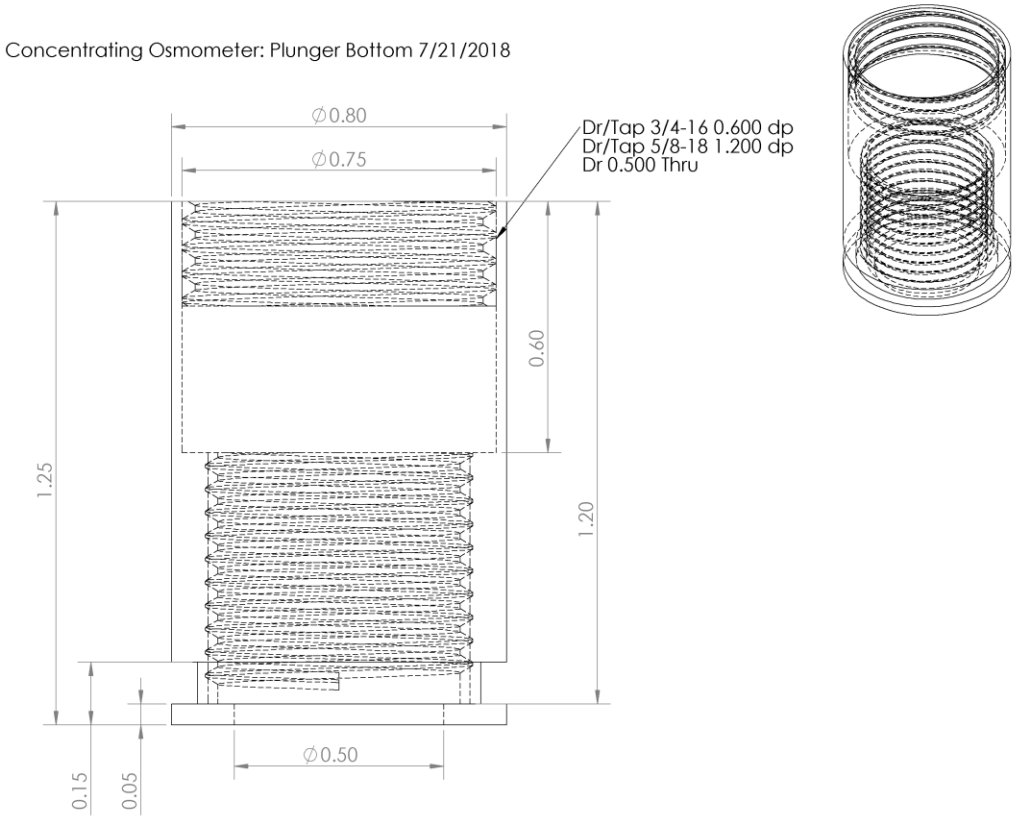


Figure E.3. Detailed Design of Concentrating Osmometer: Plunger Pressure Transducer Holder

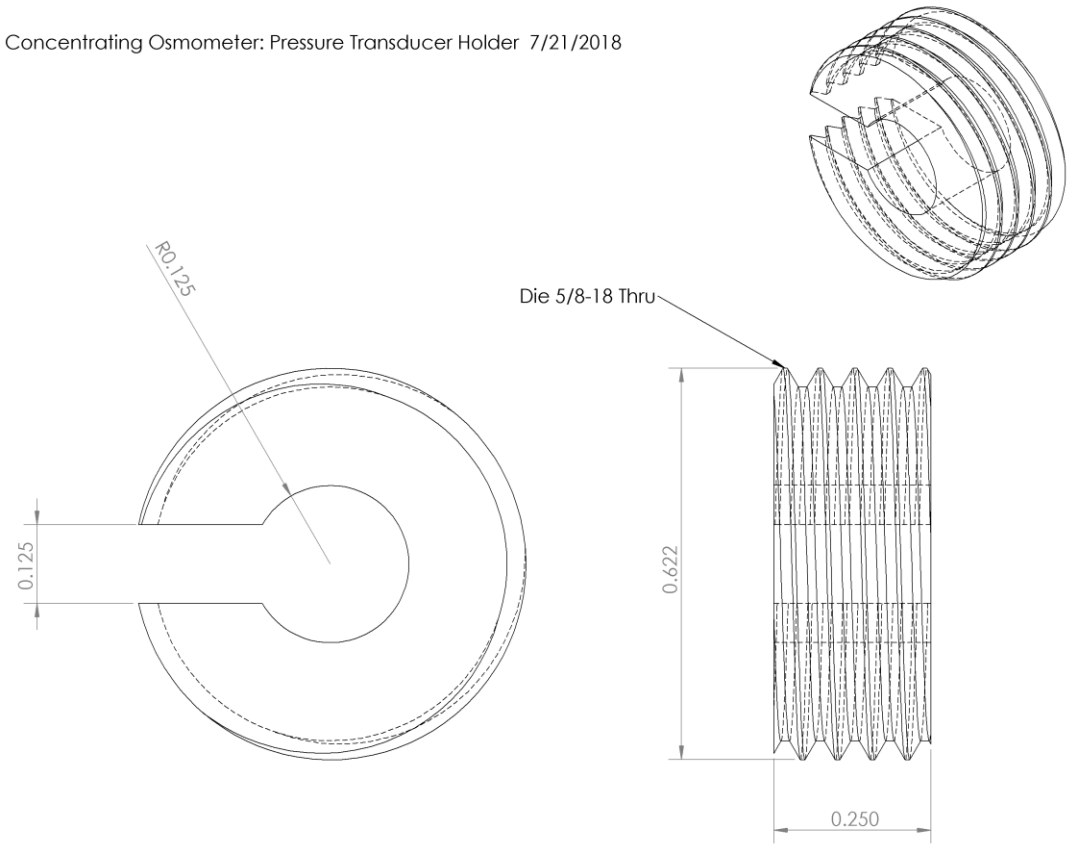


Figure E.4. Detailed Design of Concentrating Osmometer: Solute Chamber

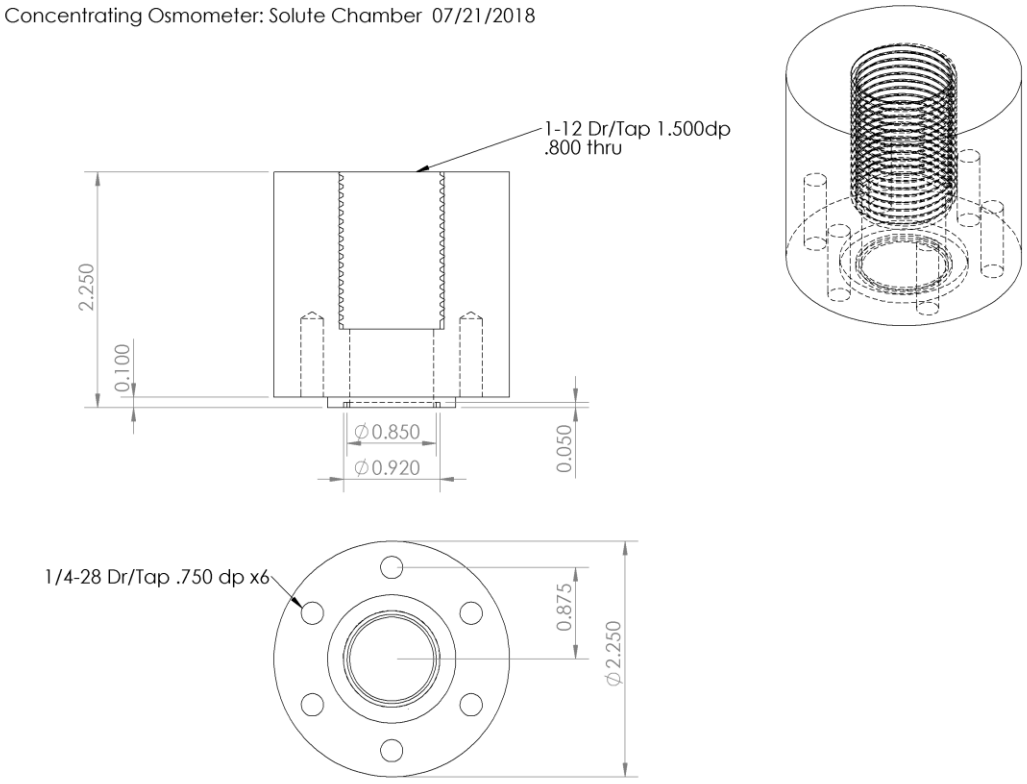
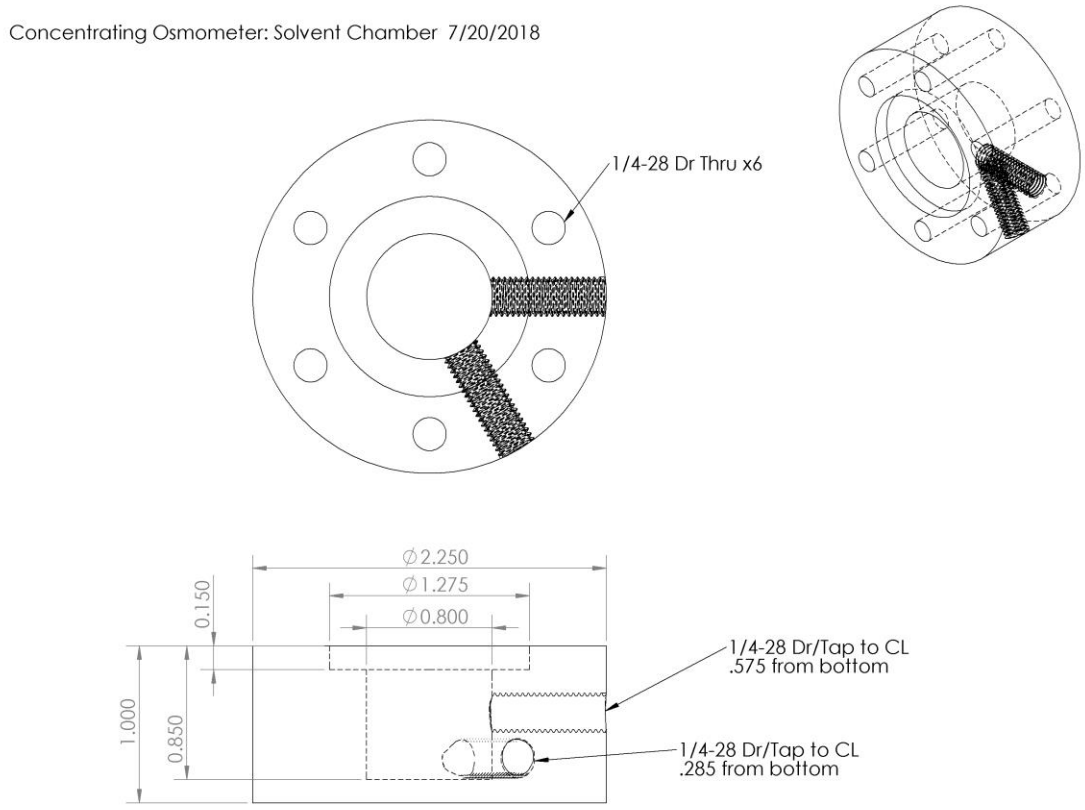


Figure E.5. Detailed Design of Concentrating Osmometer: Solvent Chamber



APPENDIX F.

DETAILED DESIGN OF CONCENTRATING OSMOMETER: VERSION 2

The first version of the concentrating osmometer has been shown to be able to concentrate protein solutions to obtain concentration pressure profile (Chapter 9). However, there is a draw back in the design that occurs when loading the sample. After sample loading, when the osmometer chambers are tightened together, to secure the membrane between the chambers, the o-ring between the membrane and the solute chamber compresses, decreasing the volume in the solute chamber and concentrating the solution. This concentrating event is accounted for by measuring the absorbance after the experiment is over and back calculating the initial concentration from the changes in solute chamber volume that occurred during the experiment. To remove the need for an absorbance measurement, the addition of valves to the side of the solute chamber allow for solution to escape, if needed, when the osmometer is being tightened together. Additionally, by moving the pressure transducer to the side of the solute chamber and out of the plunger, the concentrating factor can be easily increased by changing the height of the solute chamber and plunger.

Figure F.1. Concentrating Osmometer Version 2: A) Exploded, B) Collapsed

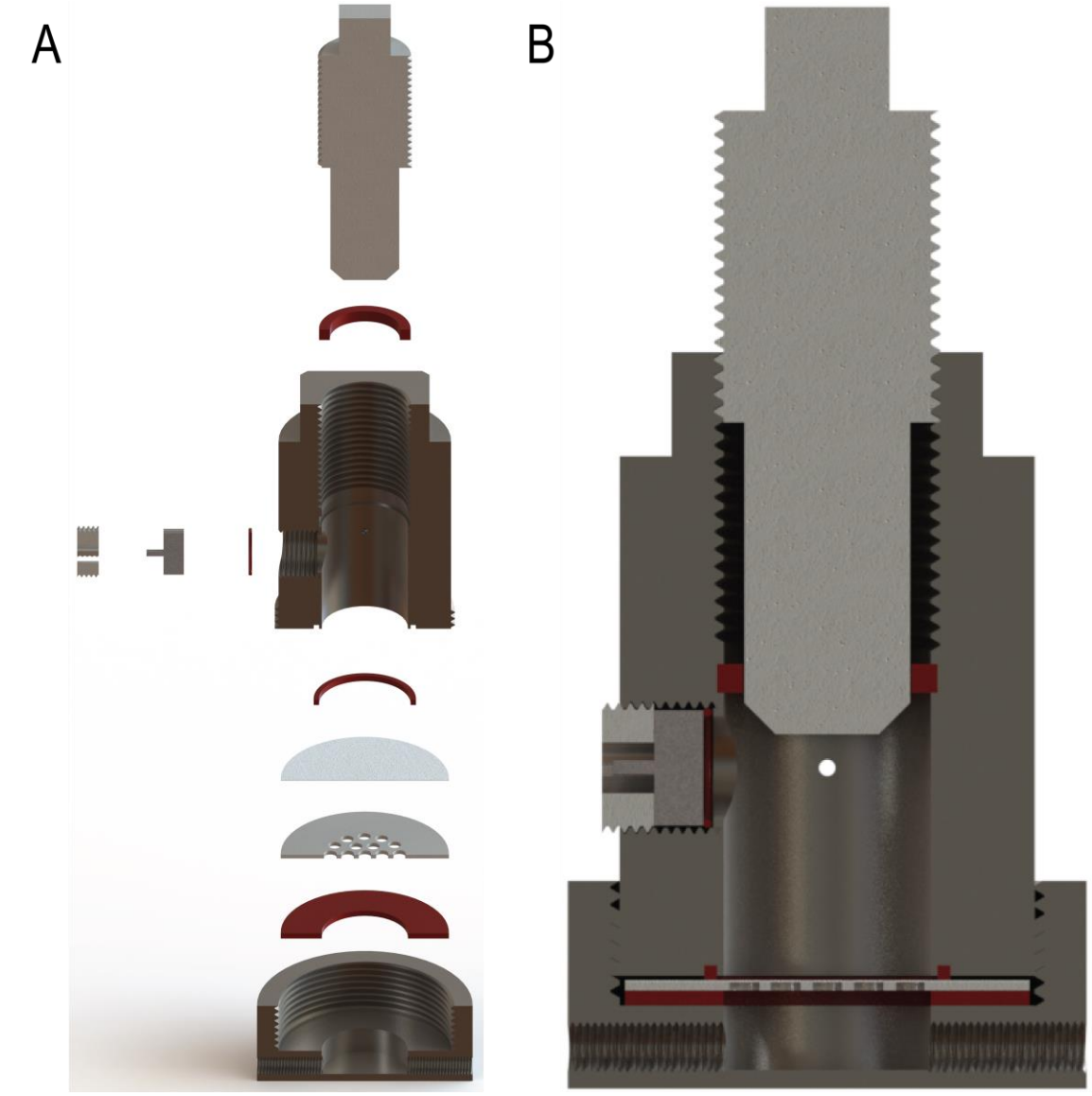


Figure F.2. Detailed Design of Concentrating Osmometer Version 2: Plunger

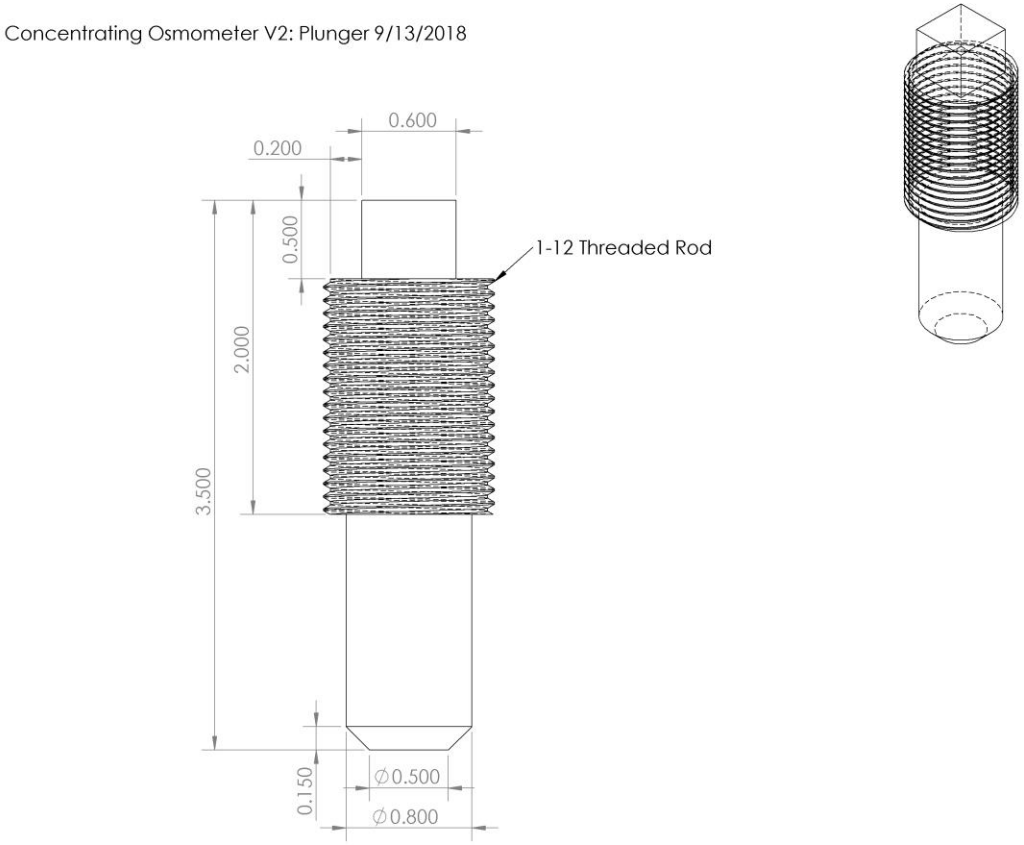


Figure F.3. Detailed Design of Concentrating Osmometer Version 2: Solute Chamber

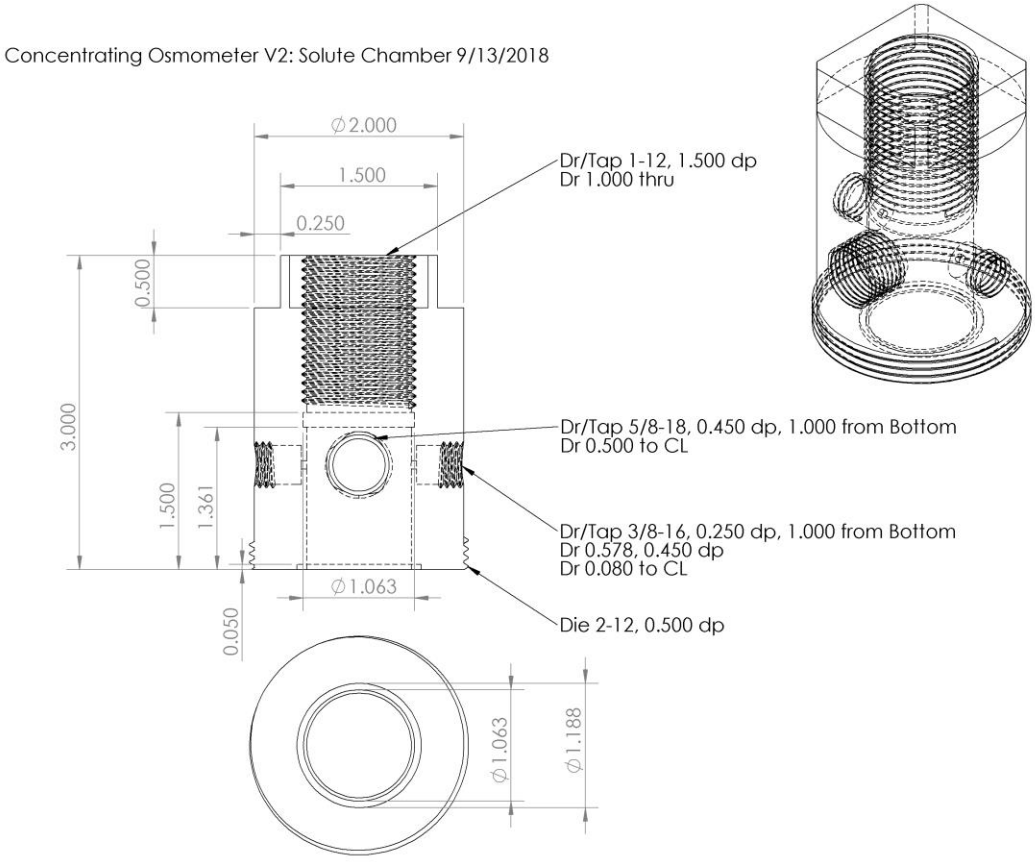


Figure F.4. Detailed Design of Concentrating Osmometer Version 2: Membrane Support

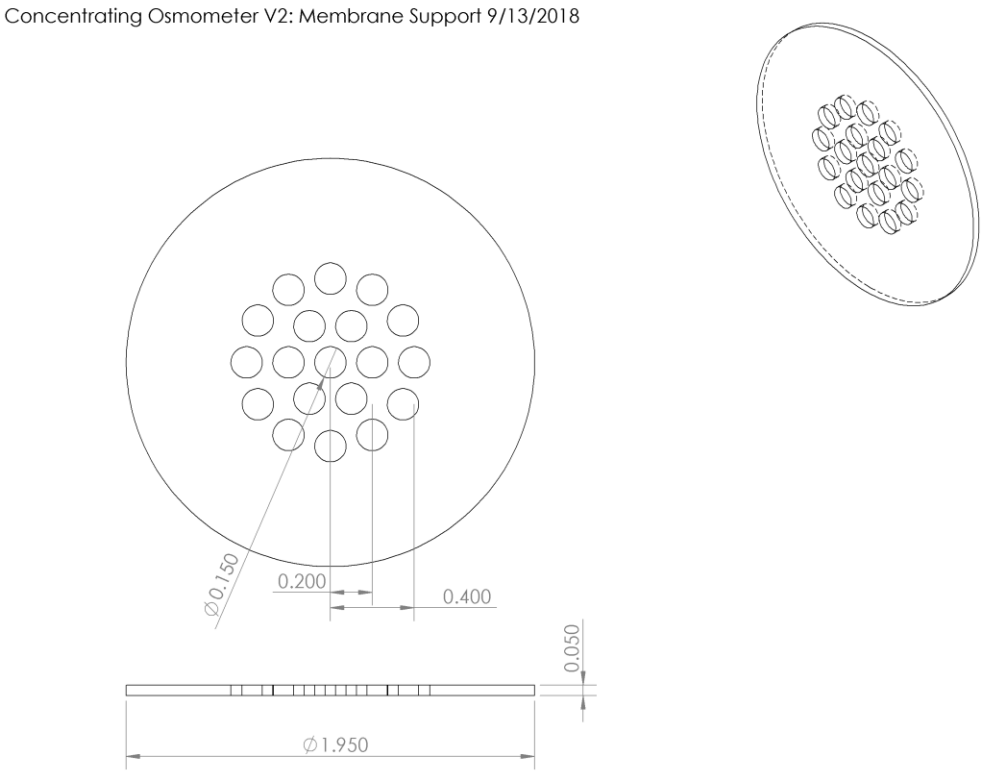
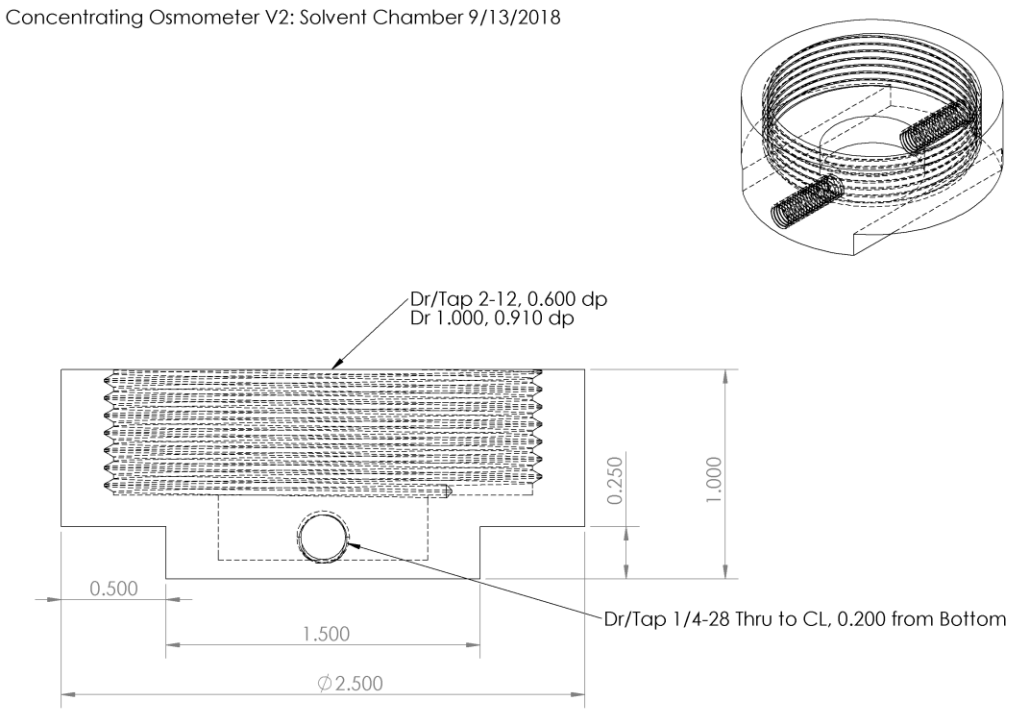


Figure F.5. Detailed Design of Concentrating Osmometer Version 2: Solvent Chamber



APPENDIX G.

DETAILED DESIGN OF TWO AXIS CLAMP

A clamp system was developed to allow for the OTD to be placed with 3-dimensional accuracy without resorting to adhesive attachment of the device to embedding screws in the spine. The hooks at the bottom of the clamps allow for positioning of the device in the foramen of the spine. This allows for a stable y-dimensional position of the device along the spine. The top bolt hole allows for the two clamp pieces to be secured together and thereby allowing for the OTD to be secured across the spine (x-dimension). The bottom hole completes the 3-dimnesional alignment by allowing a bolt to be screwed into contact with the top of the device, securing the final dimension (z-dimension).

Figure G.1. Function of Two Axis Clamp: A) Separated, B) Closed, C) Separated - Sliced, and D) Closed – Sliced

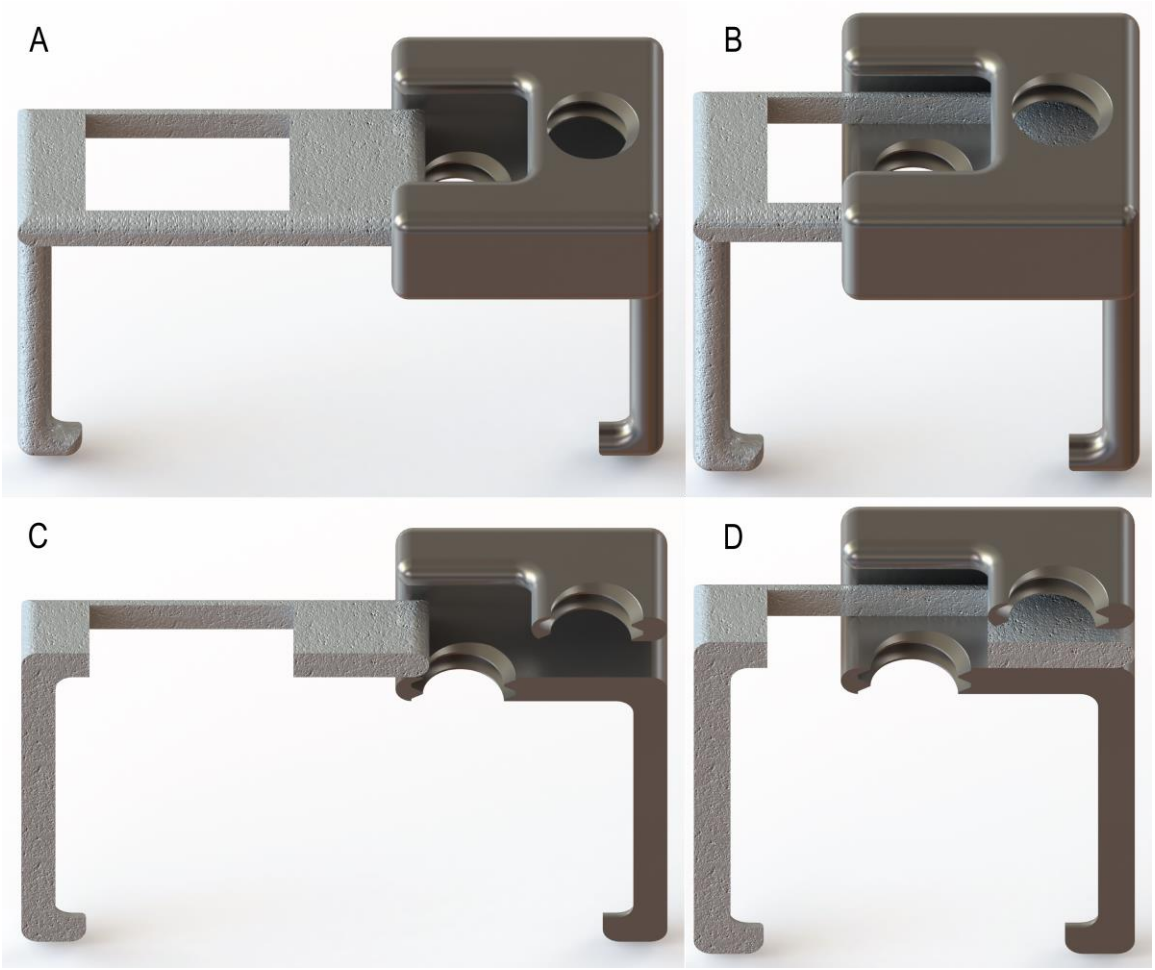


Figure G.2. Detailed Design of Two Axis Clamp: Large Side

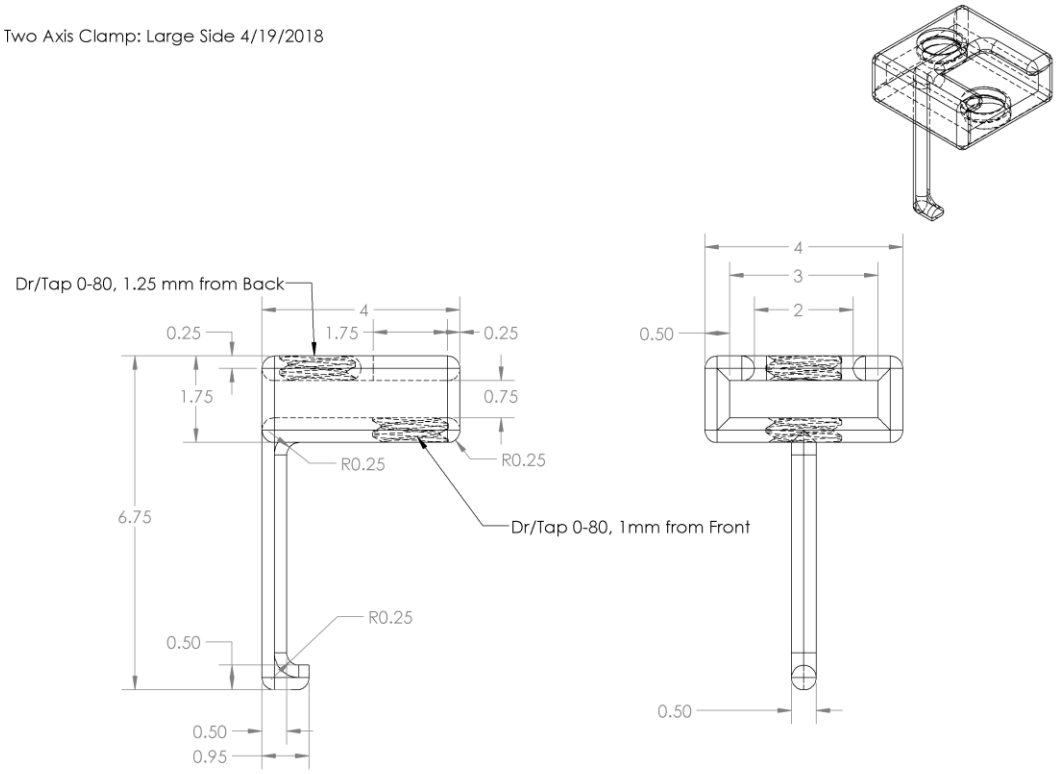
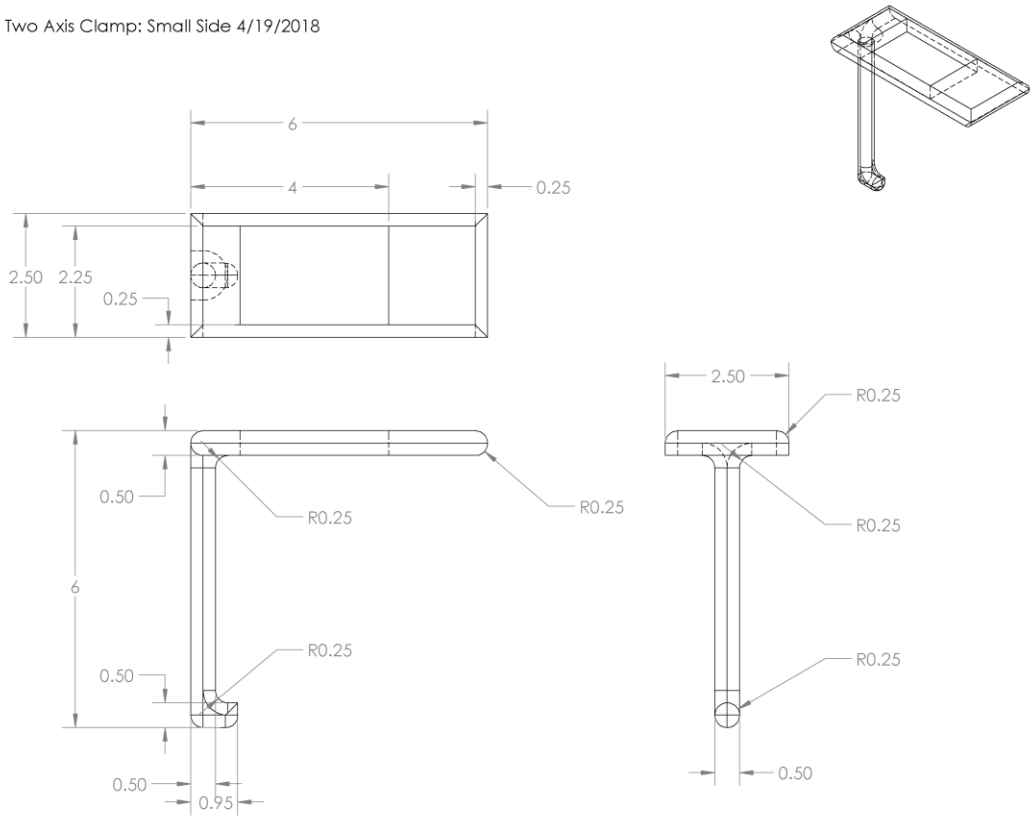


Figure G.3. Detailed Design of Two Axis Clamp: Small Side



APPENDIX H.

MULTI-DAY RAT SETUP

A counter balance is employed to remove the weight of the tubing, to ensure that there is not further injury due to the weight pressing down on the rat's spine. The counter balance is attached to the cage, with one end weighted to provide suspension and the other attached to a spring, which is connected to the rat via a vest and 3-d printed vest-to-spring adapter. The adapter has an extruded cylinder with a hole in it to allow for the spring to attach. The underside of the adapter has looped Velcro that connects with the hooked Velcro on the rat vest. Connecting all of these pieces together allows for all of the weight from the tubing to be supported by the counter balance as well as ensuring that the OTD stays upright on the spine.

Figure H.1. Multi-Day Rat Setup - Illustration

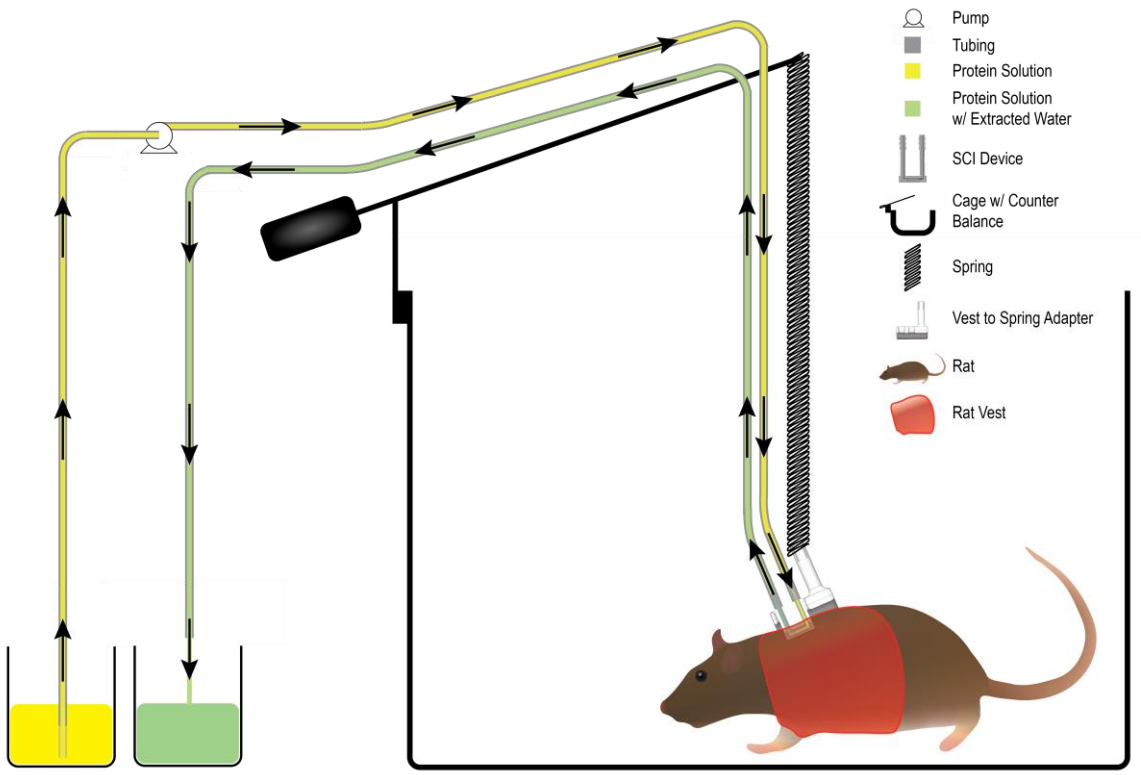


Figure H.2. Multi-Day Rat Setup: A) Top View, and B) Side View

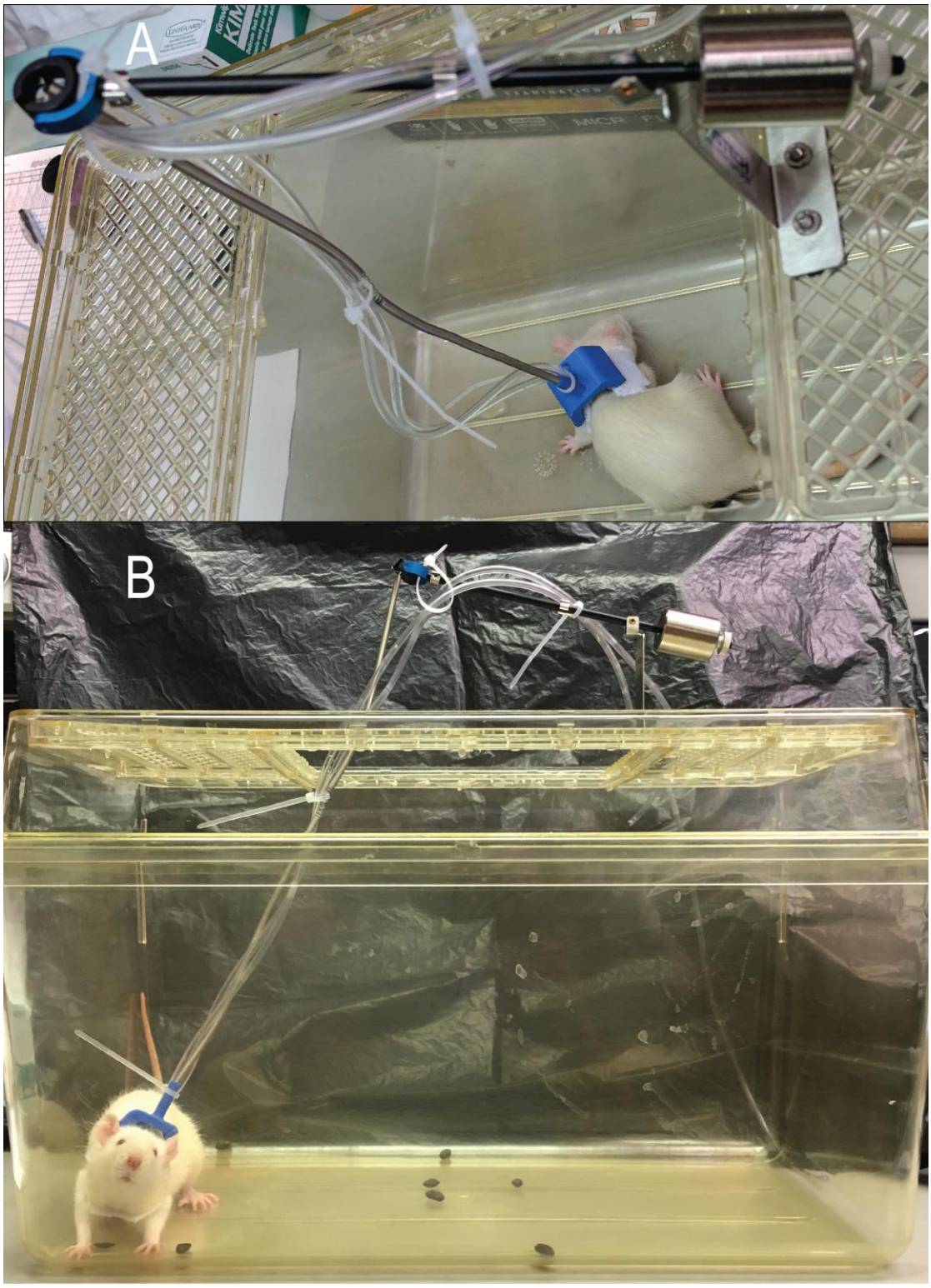
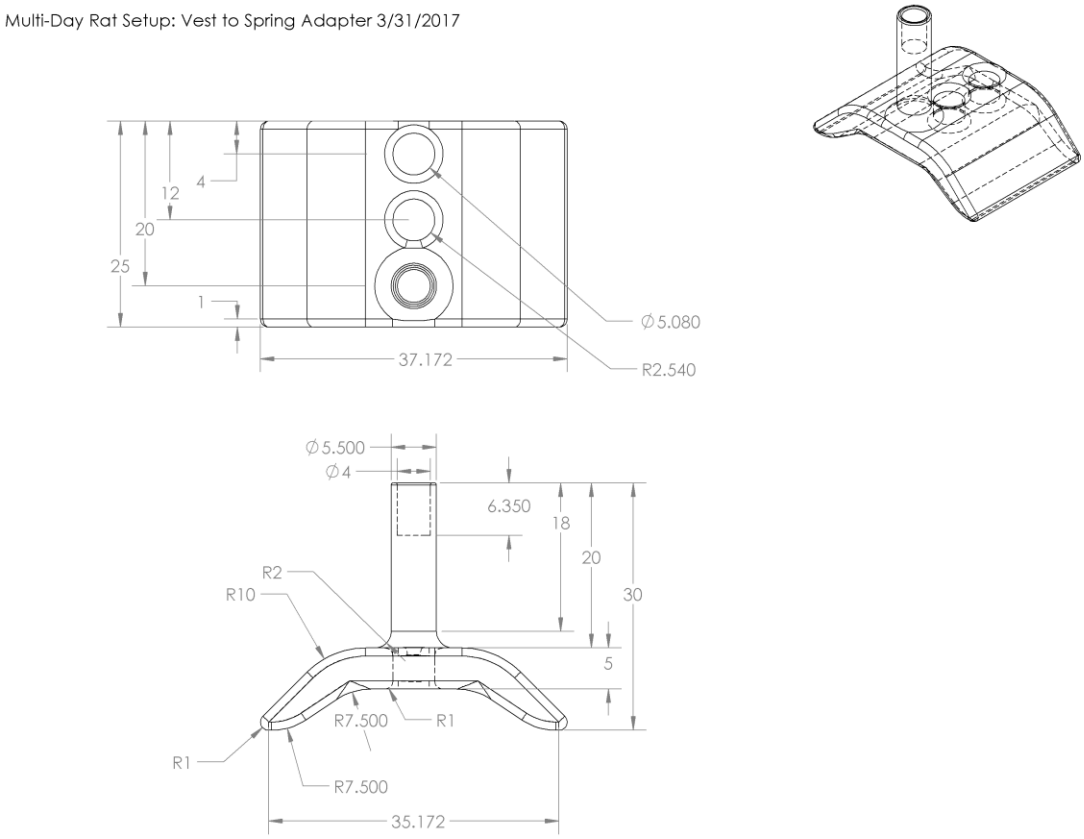


Figure H.3. Vest to Spring Adapter



APPENDIX I.

SCI DEVICE FABRICATION PROCEDURE

A. Weigh epoxy part A to be a 100:40 weigh ratio to epoxy part B.

B. Mix parts A and B thoroughly for 1 minute.

C. Hold OTD with membrane area facing up.

D. Place mixed epoxy on left and right of the membrane area.

E. Place pre-cut membrane with shiny side down on the membrane area. Apply slight downward pressure on membrane to ensure there is no gaps between the membrane and the mixed epoxy.

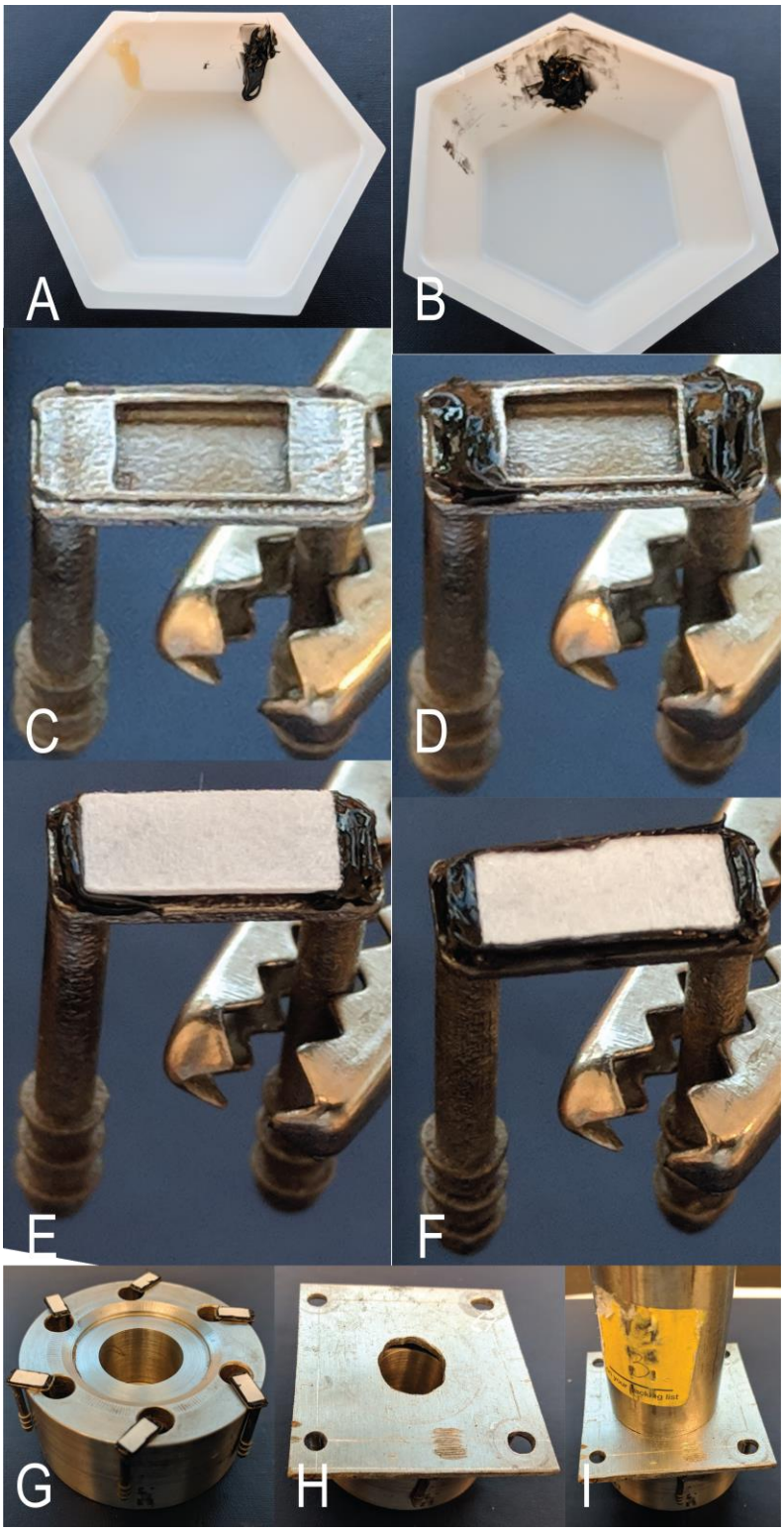
F. Add mixed epoxy around the sides of the membrane. Remove excess epoxy as needed.

G. Add devices to a holder that allows for them to have their membranes facing up.

H. Place study sheet on top of devices to ensure even weight distribution.

I. Add weight on top of sturdy sheet to ensure no gaps develop between the epoxy and membrane while the epoxy dries at room temperature over 3 days.

Figure I.1. SCI Device Fabrication Procedure



APPENDIX J.

CODE

Figure J.1. LabVIEW Code for Densimitry Method, Front

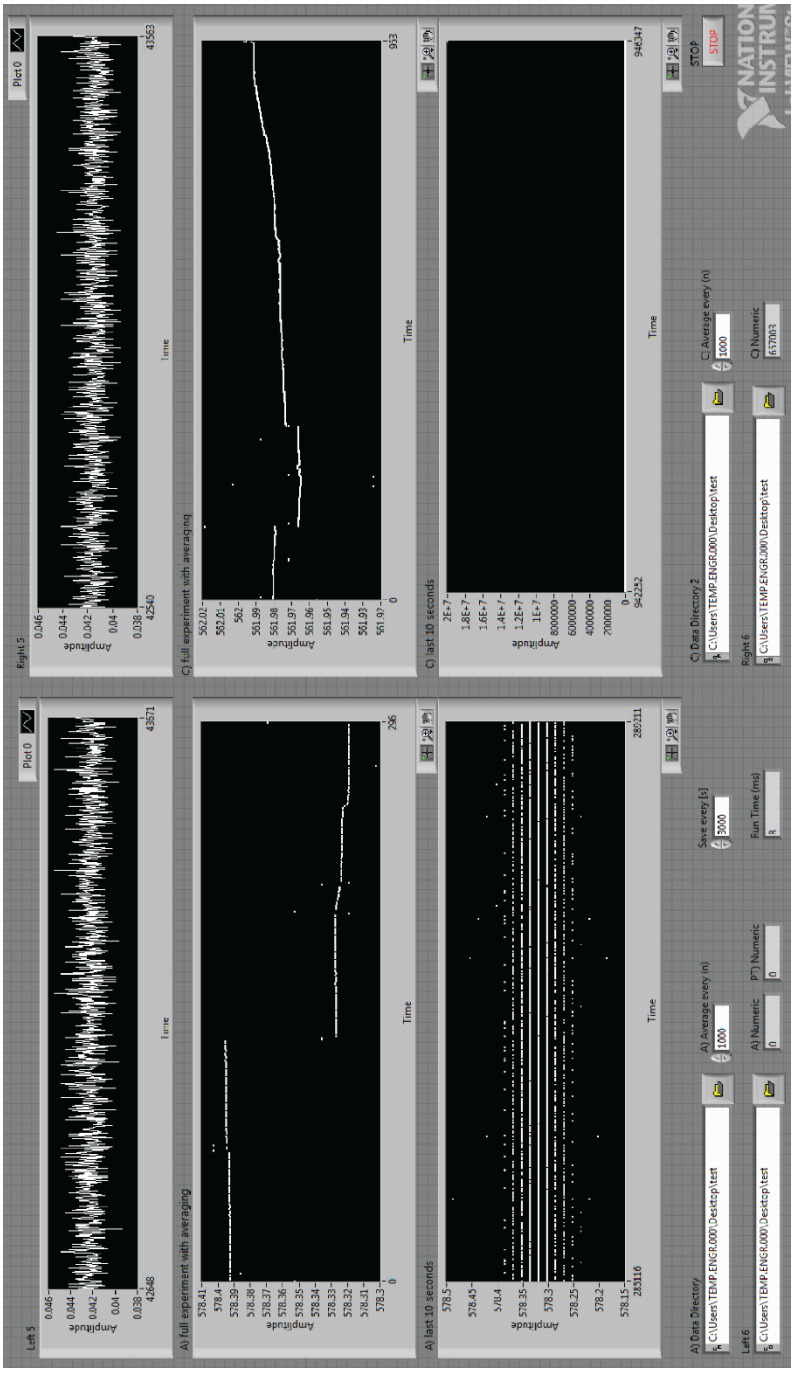


Figure J.2. LabVIEW Code for Densimitry Method, Block Diagram

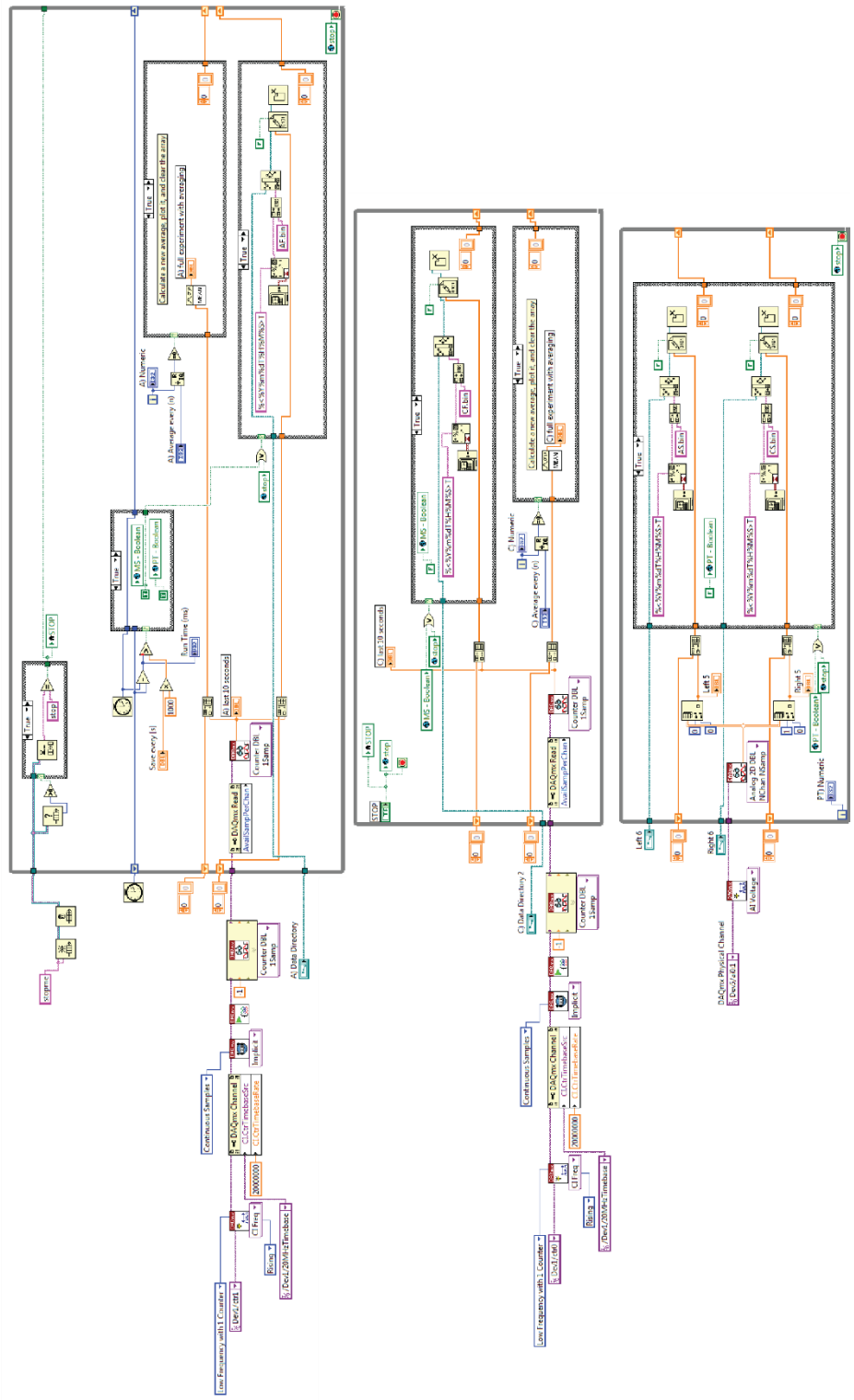


Table J.1. Table Curve Code for the Free-Solvent Based Model

Parameter	Equation
Protein 1 Mass (g / mol):	F1=66400
Protein 1 Specific Volume (mL / g):	F2=0.736
Water Mass (g / mol):	F3=18.015
Water Mass (mL / g, at 298k):	F4=1/0.997
Total Volume (L):	F5=1
Total Volume (mL):	F6=F5*1000
Salt (mol / L):	F7=0.15
Salt Volume (mL / mol):	F8=26.993
v12 Protein 1 (Hydration, g / g):	F9=A0
v12 Protein 1 (Hydration, mol / mol):	F10=F9*F1/F3
v32 Protein 1 (Ion-Binding, mol / mol):	F11=A1
Salt, Chamber 1 (mL):	F12=F5*F7*F8
Salt, Chamber 1 (mol):	F13=F12/F8
Water, Chamber 1 (mL):	F14=F6-F12
Water, Chamber 1 (mol):	F15=F14/(F3*F4)
Protein 1, Chamber 2 (mol):	F16=(X*F5)/F1
Protein 1, Chamber 2 (mL):	F17=F16*F1*F2
Salt, Chamber 2 (mL):	F18=((F6-F17)/F6)*F8*(F7*F5)
Salt, Chamber 2 (mol):	F19=F18/F8
Water, Chamber 2 (mL):	F20=F6-F17-F18
Water, Chamber 2 (mol):	F21=F20/(F3*F4)
X1, Chamber 1:	F22=F15/(F15+F13)
X1, Chamber 2:	F23=(F21-F16*F10)/(F21+F16*(1-F10-F11)+F19)
Fraction:	F24=LN(F22/F23)
R*T/V	F25=((1.20591*298)/(F3*F4/1000))
Pressure (psi):	Y=F25*F24

Table J.2. Table Curve Code for the Free-Solvent Based Model, Corrected for Salt Hydration

Parameter	Equation
Protein 1 Mass (g / mol):	F1=66400
Protein 1 Specific Volume (mL / g):	F2=0.736
Water Mass (g / mol):	F3=18.015
Water Mass (mL / g, at 298k):	F4=1/0.997
Total Volume (L):	F5=1
Total Volume (mL):	F6=F5*1000
Salt (mol / L):	F7=0.15
Salt Volume (mL / mol):	F8=26.993
v12 Protein 1 (Hydration, g / g):	F9=A0
v12 Protein 1 (Hydration, mol / mol):	F10=F9*F1/F3
v32 Protein 1 (Ion-Binding, mol / mol):	F11=A1
Salt, Chamber 1 (mL):	F12=F5*F7*F8
Salt, Chamber 1 (mol):	F13=F12/F8
Water, Chamber 1 (mL):	F14=F6-F12
Water, Chamber 1 (mol):	F15=F14/(F3*F4)
Protein 1, Chamber 2 (mol):	F16=(X*F5)/F1
Protein 1, Chamber 2 (mL):	F17=F16*F1*F2
Salt, Chamber 2 (mL):	F18=((F6-F17)/F6)*F8*(F7*F5)
Salt, Chamber 2 (mol):	F19=F18/F8
Water, Chamber 2 (mL):	F20=F6-F17-F18
Water, Chamber 2 (mol):	F21=F20/(F3*F4)
X1, Chamber 1:	F22=(F15-F13*12.7)/(F15+F13*(1-12.7))
X1, Chamber 2:	F23=(F21-F16*F10-(F19-F16*F11)*12.7)/(F21+F16*(1-F10-F11)+F19-(F19-F16*F11)*12.7)
Fraction:	F24=LN(F22/F23)
R*T/V	F25=((1.20591*298)/(F3*F4/1000))
Pressure (psi):	Y=F25*F24

Table J.3. Python Code for Densimitry Method

```
from __future__ import division
import os, cPickle as pickle, numpy, pandas as pd, scipy.signal, matplotlib as mpl, time, math
mpl.use('Tk
Agg')
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
from scipy import
stats
from matplotlib.ticker import FormatStrFormatter
from matplotlib.ticker import AutoMinorLocator


mpl.rcParams['agg.path.chunksize'] = 10000 # Prevent 'allocated too many blocks error'


print 'Start'
Timer =
time.time()




Experiment
= 4
Figure_Show = 0 # 0: Save Figure, 1: Show Figure
Figure_Resolution = 300 # DPI, 300 publication minimum or .pdf instead
Data_Shown = 1000 # Plots n number of points
Data_Used = 25000 # 0.015 hours


PMI = 35 # Initial Protein Mass (g), makes solution 350 g/L incase masses are not specified in
experimental data
SMI = 74.9 # Initial Solvent Mass (g), makes solution 350 g/L incase masses are not specified in
experimental data
Hydraulic_Permeability_Comparison = 1 * 10 ** -7


# Changes time data into different units
Time_Unit_Convers
ion = 3600
X_Label =
'Time (h)'




if
Experiment
== 1:
```

```
        Data_Date = '20180810'
        Device_Setup = 'Device'
        Directory_Signal = '/Research/Data/SCI/Mass Sensor/Device 20180810/Device Only/Signal - PT'
        Directory_Baseline = '/Research/Data/SCI/Mass Sensor/Device 20180810/Device Only/Baseline - PT'
        Calibration_Baseline = 103
        Calibration_Signal = 201
        Flowrate_Time_Changed = numpy.array([0, 0.75, 1.5, 2.25, 3, 3.75, 4.5, 5.5, 6.25, 7.75, 8, 8.75, 9.5])
        Flowrate = numpy.array([100, 100, 100, 100, 250, 100, 50, 100, 500, 100, 25, 100, 100]) # grams per minute
        Ylim_Min = [1.05, -0.003]
        Ylim_Max = [1.11, 0.003]
        Volume_from_Device_to_Valve = 0.8 # Volume (mL) to get from the device to the 3-way valve/solenoid
        Volume_from_Valve_to_Sensor = 0.8 # Volume (mL) to get from the 3-way valve/solenoid to the mass sensor
        Hydraulic_Permeability = 1.4 * 10 ** -7
    elif Experiment == 2:
        Data_Date = '20180810'
        Device_Setup = 'Device'
        Directory_Signal = '/Research/Data/SCI/Mass Sensor/Device 20180810/Device Only/Signal - MS'
        Directory_Baseline = '/Research/Data/SCI/Mass Sensor/Device 20180810/Device Only/Baseline - MS'
        Calibration_Baseline = 103
        Calibration_Signal = 201
        Flowrate_Time_Changed = numpy.array([0, 0.75, 1.5, 3, 3.75, 4.5, 5.5, 6.25, 7, 7.75, 8.75, 9.5])
        Flowrate_Time_Changed = numpy.array([0, 0.75, 1.5, 3, 3.67, 4.5, 5.44, 6.25, 7, 7.75, 8.75, 9.5])
        Flowrate = numpy.array([100, 100, 100, 250, 100, 50, 100, 500, 100, 25, 100, 100]) # grams per minute
        Flowrates_Measured = numpy.array([100, 250, 50, 500, 25]) # grams per minute
        Ylim_Min = [1.093, -0.0035]
        Ylim_Max = [1.101, 0.0005]
        Volume_from_Device_to_Valve = 0.8 # Volume (mL) to get from the device to the 3-way valve/solenoid
        Volume_from_Valve_to_Sensor = 0.8 # Volume (mL) to get from the 3-way valve/solenoid to the mass sensor
        Hydraulic_Permeability = 1.4 * 10 ** -7
    elif Experiment == 3:
        Data_Date = '20180813'
        Device_Setup = 'Device'
        Directory_Signal = '/Research/Data/SCI/Mass Sensor/Device 20180813/Device Only/Signal - MS'
```

```
        Directory_Baseline = '/Research/Data/SCI/Mass Sensor/Device 20180813/Device Only/Baseline - MS'
        Calibration_Baseline = 103
        Calibration_Signal = 201
        Flowrate_Time_Changed = numpy.array([0, 1.25, 2, 2.75, 3.5, 4.25, 5, 5.75, 6.5, 7.25, 8.75, 9.75])
        Flowrate_Time_Changed = numpy.array([0, 1.25, 2, 2.75, 3.44, 4.25, 4.96, 5.75, 6.5, 7.25, 8.76, 9.75])
        Flowrate = numpy.array([100, 100, 100, 250, 100, 50, 100, 500, 100, 25, 100, 100]) # grams per minute
        Flowrates_Measured = numpy.array([100, 250, 50, 500, 25]) # grams per minute
        Ylim_Min = [1.093, -0.004]
        Ylim_Max = [1.103, 0.001]
        Volume_from_Device_to_Valve = 0.8 # Volume (mL) to get from the device to the 3-way valve/solenoid
        Volume_from_Valve_to_Sensor = 0.8 # Volume (mL) to get from the 3-way valve/solenoid to the mass sensor
        Hydraulic_Permeability = 2.1 * 10 ** -7
elif Experiment == 4:
        Data_Date = '20180828'
        Device_Setup = 'Tissue'
        Directory_Signal = '/Research/Data/SCI/Mass Sensor/Device 20180828 - Tissue/Device Only/Signal - MS'
        Directory_Baseline = '/Research/Data/SCI/Mass Sensor/Device 20180828 - Tissue/Device Only/Baseline - MS'
        Calibration_Baseline = 103
        Calibration_Signal = 201
        Flowrate_Time_Changed = numpy.array([0, 0.75, 1.5, 2.25, 3, 3.75, 4.5, 5.25, 6, 6.75, 7.5, 8.25])
        Flowrate_Time_Changed = numpy.array([0, 0.75, 1.48, 2.23, 2.96, 3.7, 4.44, 5.25, 6, 6.75, 7.5, 8.25])
        Flowrate = numpy.array([100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100]) # grams per minute
        Flowrates_Measured = numpy.array([100, 100, 100, 100, 100]) # grams per minute
        Ylim_Min = [1.095, -0.003]
        Ylim_Max = [1.110, 0.0015]
        Volume_from_Device_to_Valve = 0.8 # Volume (mL) to get from the device to the 3-way valve/solenoid
        Volume_from_Valve_to_Sensor = 0.8 # Volume (mL) to get from the 3-way valve/solenoid to the mass sensor
        Hydraulic_Permeability = 1.9 * 10 ** -7


# Determines at what time the solution from the device reaches the sensor
Counter = 1
```

```
Sensor_Volume = 1 # Volume of the mass sensor
Flowrate_Time_Sensor = [0] # Time for the new solution to reach the sensor after a flowrate change
Flowrate_Time_Sensor_Fill = [0] # Time for the new solution to fill the sensor after a flowrate change
Flowrate_Time_Difference = [0]
while Counter < len(Flowrate_Time_Changed):
    if Counter < (len(Flowrate_Time_Changed) - 1):
        Volume_to_Sensor = Volume_from_Valve_to_Sensor

        if ((Volume_to_Sensor / (Flowrate[Counter] / 1000)) / 60) >
        (Flowrate_Time_Changed[Counter + 1] - Flowrate_Time_Changed[Counter]):
            Volume_to_Sensor_New = Volume_to_Sensor - (Flowrate[Counter] / 1000) *
            (Flowrate_Time_Changed[Counter + 1] - Flowrate_Time_Changed[Counter]) *
            60
            Flowrate_Time_Sensor.append(Flowrate_Time_Changed[Counter] +
            (Volume_to_Sensor_New / (Flowrate[Counter + 1] / 1000)) / 60)
        else:
            Flowrate_Time_Sensor.append(Flowrate_Time_Changed[Counter] +
            (Volume_to_Sensor / (Flowrate[Counter] / 1000)) / 60)

        if (((Volume_to_Sensor + Volume_from_Device_to_Valve + Sensor_Volume) /
        (Flowrate[Counter] / 1000)) / 60) > (Flowrate_Time_Changed[Counter + 1] -
        Flowrate_Time_Changed[Counter]):
            Volume_to_Sensor_New = (Volume_to_Sensor +
            Volume_from_Device_to_Valve + Sensor_Volume) - (Flowrate[Counter] / 1000)
            * (Flowrate_Time_Changed[Counter + 1] - Flowrate_Time_Changed[Counter]) *
            60
            Flowrate_Time_Sensor_Fill.append(Flowrate_Time_Changed[Counter] +
            (Volume_to_Sensor_New / (Flowrate[Counter + 1] / 1000)) / 60)
        else:
            Flowrate_Time_Sensor_Fill.append(Flowrate_Time_Changed[Counter] +
            ((Volume_to_Sensor + Volume_from_Device_to_Valve + Sensor_Volume) /
            (Flowrate[Counter] / 1000)) / 60)

        Flowrate_Time_Difference.append(Flowrate_Time_Sensor_Fill[len(Flowrate_Time_Sen
        sor_Fill) - 1] - Flowrate_Time_Sensor[len(Flowrate_Time_Sensor) - 1])

    else:
        Volume_to_Sensor = Volume_from_Valve_to_Sensor
        Flowrate_Time_Sensor.append(Flowrate_Time_Changed[Counter] +
        (Volume_to_Sensor / (Flowrate[Counter] / 1000)) / 60)
        Flowrate_Time_Sensor_Fill.append(Flowrate_Time_Changed[Counter] +
        ((Volume_to_Sensor + Sensor_Volume) / (Flowrate[Counter] / 1000)) / 60)
        Flowrate_Time_Difference.append(Flowrate_Time_Sensor_Fill[len(Flowrate_Time_Sen
        sor_Fill) - 1] - Flowrate_Time_Sensor[len(Flowrate_Time_Sensor) - 1])
    Counter = Counter + 1
```

```
# Outputs Density of Salt and Protein
PSV = 0.736 # Protein Density (mL/g)
SD = 0.991171346 # Solvent Density (mL/g)

PSDI = (PMI + SMI) / (PMI * PSV + SMI * SD) # Approximate Protein Solution Density - Initial (g/mL), ex: 1.0990
Concentration = PMI / (PMI * PSV + SMI * SD) * 1000

print('Estimated Initial Protein Solution Density (g/mL): {:.4f}'.format(PSDI))
print('Estimated Initial aCSF Density (g/mL): {:.4f}'.format(SD))

print('Inital Parameters Took: {:.0f}s or {:.1f}min'.format((time.time() - Timer), ((time.time() - Timer) / 60)))
Timer =
time.time()



def Bulk_Calibration(frequency):
    if Calibration == 101:
            #Sensor A, calibration from 1/24/18
       m = -75.4467770117 #Hz/(g/mL)
       b = 551.359410499 #Hz
    elif Calibration == 102:
            #Sensor A, calibration from 3/14/18
            m = -75.9534524552 #Hz/(g/mL)
            b = 551.906338338 #Hz
    elif Calibration == 103:
            #Sensor A, calibration from 8/6/18
            m = -73.3145134899 #Hz/(g/mL)
            b = 549.210591635 #Hz
    elif Calibration == 201:
            #Sensor B, calibration from 8/6/18
            m = -71.1742087041 #Hz/(g/mL)
            b = 545.178638323 #Hz
    elif Calibration == 301:
       #Sensor C, calibration from 3/14/18
       m = -74.573737545571 #Hz/(g/mL)
       b = 596.15711334 #Hz
    elif Calibration == 302:
            #Sensor C, calibration from 1/24/18
```

```
                m = -74.1164124322 #Hz/(g/mL)
                b = 595.757304004 #Hz

        return (frequency - b)/m



def Low_Pass_Filter(data, Sampling_Frequency, Frequency_Cut_Off, N, a=1): # 474, 3, 1001
        Frequency_Cut_Off = Frequency_Cut_Off / (0.5 * Sampling_Frequency)
        b = scipy.signal.firwin(N, cutoff=Frequency_Cut_Off, window='hamming') #Filter numerator
        return scipy.signal.lfilter(b, a, data)



# Creates a directory for plots, if needed. Equivalent to using mkdir -p on the command line
def
mkdir_p(mypath):
   from errno import
EEXIST
   from os import makedirs,path


try:

makedirs(mypath)
   except OSError as exc: # Python >2.5
      if exc.errno == EEXIST and path.isdir(mypath):
         pass
      else:
raise



def
Plot_Density_Figur
e():
        fig =
        plt.figure()
        # Raw and filtered data
        ax0 = plt.subplot(311)
        ax0.plot((Time_Total[::Data_Shown] / Time_Unit_Conversion),
        Density_Baseline_Unfiltered[::Data_Shown], color='b', linestyle='-', label='Baseline (Before)')
        ax0.plot((Time_Total[::Data_Shown] / Time_Unit_Conversion),
        Density_Baseline_Filtered[::Data_Shown], color='g', linestyle='-', label='Baseline (Before, Low-
        Pass Filter)')
```

```
ax0.plot((Time_Total[::Data_Shown] / Time_Unit_Conversion),
Density_Signal_Unfiltered[::Data_Shown], color='r', linestyle='-', label='Signal (After)')
ax0.plot((Time_Total[::Data_Shown] / Time_Unit_Conversion),
Density_Signal_Filtered[::Data_Shown], color='m', linestyle='-', label='Signal (After, Low-Pass
Filter)')

# Difference between filtered data sets
ax1 = plt.subplot(312, sharex = ax0)
ax1.plot((Time_Total[::] / Time_Unit_Conversion), Difference_Frequency_Filtered[::], color='k',
linestyle='-')

# Flowrate to second y-axis
ax2 = plt.subplot(313, sharex = ax0)
Counter = 0
while Counter <= (len(Flowrate_Time_Changed) - 2):
        ax2.plot([Flowrate_Time_Changed[Counter], Flowrate_Time_Changed[Counter+1]],
        [Flowrate[Counter], Flowrate[Counter]], linewidth=1.5, color='k', linestyle='-')
        ax2.plot([Flowrate_Time_Changed[Counter+1], Flowrate_Time_Changed[Counter+1]],
        [Flowrate[Counter], Flowrate[Counter+1]], linewidth=1.5, color='k', linestyle='-')
        Counter = Counter + 1

# Places Greyed areas showing the location that the data for comparison is taken from
Counter = 1
while Counter < (len(Flowrate_Time_Sensor) - 1):
        ax0.axvspan(Flowrate_Time_Sensor[Counter], (Flowrate_Time_Sensor[Counter] -
        Data_Used * 5.94832E-07), facecolor='0.5', alpha = 0.5) # 5.94832E-07 is time in hours
        per data point
        ax1.axvspan(Flowrate_Time_Sensor[Counter], (Flowrate_Time_Sensor[Counter] -
        Data_Used * 5.94832E-07), facecolor='0.5', alpha = 0.5) # 5.94832E-07 is time in hours
        per data point
        ax2.axvspan(Flowrate_Time_Sensor[Counter], (Flowrate_Time_Sensor[Counter] -
        Data_Used * 5.94832E-07), facecolor='0.5', alpha = 0.5) # 5.94832E-07 is time in hours
        per data point
        Counter = Counter + 1

ax0.legend(loc='upper center', fontsize=10, numpoints=1, bbox_to_anchor=(0.5, -2.75),
fancybox=True, frameon=False, handlelength=3, ncol=2)

# Shrink current axis by 10%
# box = ax2.get_position()
# ax2.set_position([box.x0, box.y0 + box.height * 0.1, box.width, box.height * 0.9])
# ax2.legend(loc='upper center', fontsize=10, numpoints=1, bbox_to_anchor=(0.5, -0.15),
fancybox=True, frameon=False, handlelength=3, ncol=4) # ncol: number of columns
```

```
    ax0.yaxis.set_major_formatter(FormatStrFormatter('%.4f'))
    ax1.yaxis.set_major_formatter(FormatStrFormatter('%.4f'))
    ax0.set_ylim(Ylim_Min[0], Ylim_Max[0])
    ax1.set_ylim(Ylim_Min[1], Ylim_Max[1])
    ax2.set_ylim(0,(max(Flowrate)+5))
    ax0.xaxis.set_visible(False)
    ax1.xaxis.set_visible(False)
    ax0.tick_params(top=False, right=False, bottom =True)
    ax1.tick_params(top=False, right=False, bottom =True)
    ax2.tick_params(top=False, right=False, bottom =True)
    ax0.spines['top'].set_visible(False)
    ax1.spines['top'].set_visible(False)
    ax2.spines['top'].set_visible(False)
    ax0.spines['right'].set_visible(False)
    ax1.spines['right'].set_visible(False)
    ax2.spines['right'].set_visible(False)
    ax0.yaxis.set_minor_locator(AutoMinorLocator(2)) # number of minor ticks on y-axis
    ax1.yaxis.set_minor_locator(AutoMinorLocator(2)) # number of minor ticks on y-axis
    ax2.yaxis.set_minor_locator(AutoMinorLocator(2)) # number of minor ticks on y-axis
    ax2.set_xlabel(X_Label, fontsize=10)
    ax0.set_ylabel(r'Density ($\frac{g}{mL}$)', fontsize=10)
    ax1.set_ylabel(r'Density Difference ($\frac{g}{mL}$)', fontsize=10)
    ax2.set_ylabel(r'Flowrate ($\frac{mL}{min}$)', fontsize=10)

    plt.suptitle(Device_Setup + ' ' + Data_Date, fontsize=12)

    if
    Figure_Show
    == 0:
            File_Name = Data_Date + '/' + Data_Date + ' ' + Device_Setup + '.png'
            plt.savefig(File_Name, dpi=Figure_Resolution, bbox_inches='tight')
    else:
            plt.show()
    return
    ()



def Plot_Flowrate_Dependency_Chart():
    fig =
    plt.figure()
    Markers = '.'
```

```
# Baseline
ax0 = plt.subplot(411)
ax0.plot(Flowrates_Measured[0:len(Baseline_Noise_Average)],
Baseline_Noise_Average[0:len(Baseline_Noise_Average)], color='b', marker=Markers,
markersize=7, markeredgewidth=1, linestyle='None', label='Baseline')
ax0.errorbar(Flowrates_Measured[0:len(Baseline_Noise_Average)],
Baseline_Noise_Average[0:len(Baseline_Noise_Average)],
yerr=Baseline_Noise_StdDev[0:len(Baseline_Noise_StdDev)], fmt='none', ecolor='b',
elinewidth=1, markeredgewidth=1, capsize=10)
ax0.yaxis.set_major_formatter(FormatStrFormatter('%.4f')) # Sets decimals to show
ax0.tick_params(top=False, right=False, bottom = True)
ax0.spines['top'].set_visible(False)
ax0.spines['right'].set_visible(False)
ax0.xaxis.set_visible(False)
ax0.yaxis.set_minor_locator(AutoMinorLocator(2)) # number of minor ticks on y-axis
ax0.set_ylabel(r'Baseline Density ($\frac{g}{mL}$)', fontsize=8)

# #
Signal
ax1 = plt.subplot(412, sharex = ax0)
ax1.plot(Flowrates_Measured[0:len(Signal_Noise_Average)],
Signal_Noise_Average[0:len(Signal_Noise_Average)], color='r', marker=Markers, markersize=7,
markeredgewidth=1, linestyle='None', label='Signal')
ax1.errorbar(Flowrates_Measured[0:len(Signal_Noise_Average)],
Signal_Noise_Average[0:len(Signal_Noise_Average)],
yerr=Signal_Noise_StdDev[0:len(Signal_Noise_StdDev)], fmt='none', ecolor='r', elinewidth=1,
markeredgewidth=1, capsize=10)
ax1.yaxis.set_major_formatter(FormatStrFormatter('%.4f')) # Sets decimals to show
ax1.tick_params(top=False, right=False, bottom = True)
ax1.spines['top'].set_visible(False)
ax1.spines['right'].set_visible(False)
ax1.xaxis.set_visible(False)
ax1.yaxis.set_minor_locator(AutoMinorLocator(2)) # number of minor ticks on y-axis
ax1.set_ylabel(r'Signal Density ($\frac{g}{mL}$)', fontsize=8)

# Difference
ax2 = plt.subplot(413, sharex = ax0)
ax2.plot(Flowrates_Measured[0:len(Difference_Frequency_Filtered_Average)],
Difference_Frequency_Filtered_Average[0:len(Difference_Frequency_Filtered_Average)],
color='k', marker=Markers, markersize=7, markeredgewidth=1, linestyle='None',
label='Difference')
ax2.errorbar(Flowrates_Measured[0:len(Difference_Frequency_Filtered_Average)],
Difference_Frequency_Filtered_Average[0:len(Difference_Frequency_Filtered_Average)],
yerr=Difference_Frequency_Filtered_StdDev[0:len(Difference_Frequency_Filtered_StdDev)],
fmt='none', ecolor='k', elinewidth=1, markeredgewidth=1, capsize=10)
```

```
ax2.yaxis.set_major_formatter(FormatStrFormatter('%.3f')) # Sets decimals to show
ax2.tick_params(top=False, right=False, bottom = True)
ax2.spines['top'].set_visible(False)
ax2.spines['right'].set_visible(False)
ax2.xaxis.set_visible(False)
ax2.yaxis.set_minor_locator(AutoMinorLocator(2)) # number of minor ticks on y-axis
ax2.set_ylabel(r'Density Difference ($\frac{g}{mL}$)', fontsize=8)

# Extraction
Rate
ax3 = plt.subplot(414, sharex = ax0)
ax3.plot(Flowrates_Measured[0:len(Extraction_Rate_Average)],
Extraction_Rate_Average[0:len(Extraction_Rate_Average)], color='k', marker=Markers,
markersize=7, markeredgewidth=1, linestyle='None', label='Extraction Rate')
ax3.errorbar(Flowrates_Measured[0:len(Extraction_Rate_Average)],
Extraction_Rate_Average[0:len(Extraction_Rate_Average)],
yerr=Extraction_Rate_StdDev[0:len(Extraction_Rate_StdDev)], fmt='none', ecolor='k',
elinewidth=1, markeredgewidth=1, capsize=10)
ax3.tick_params(top=False, right=False)
ax3.spines['top'].set_visible(False)
ax3.spines['right'].set_visible(False)
ax3.yaxis.set_minor_locator(AutoMinorLocator(2)) # number of minor ticks on y-axis
ax3.set_xlabel(r'Flowrate ($\frac{uL}{min}$)', fontsize=10)
ax3.set_ylabel(r'Extraction Rate ($\frac{uL}{h}$)', fontsize=8)
ax3.set_xlim(0,(max(Flowrate) + 50))

# Adds sequence of values to data points
Offset = 20 # Distance in ul/min from datat points
Counter = 0
while Counter < len(Baseline_Noise_Average):
        ax0.annotate(Counter, xy=(Flowrates_Measured[Counter],
        Baseline_Noise_Average[Counter]), xytext=((Flowrates_Measured[Counter] + Offset),
        Baseline_Noise_Average[Counter]))
        ax1.annotate(Counter, xy=(Flowrates_Measured[Counter],
        Signal_Noise_Average[Counter]), xytext=((Flowrates_Measured[Counter] + Offset),
        Signal_Noise_Average[Counter]))
        ax2.annotate(Counter, xy=(Flowrates_Measured[Counter],
        Difference_Frequency_Filtered_Average[Counter]),
        xytext=((Flowrates_Measured[Counter] + Offset),
        Difference_Frequency_Filtered_Average[Counter]))
        ax3.annotate(Counter, xy=(Flowrates_Measured[Counter],
        Extraction_Rate_Average[Counter]), xytext=((Flowrates_Measured[Counter] + Offset),
        Extraction_Rate_Average[Counter]))
        Counter = Counter + 1
```

```
        plt.suptitle(Device_Setup + ' ' + Data_Date + ' Flowrate Dependency Chart', fontsize=12)

        if
        Figure_Show
        == 0:
                File_Name = Data_Date + '/' + Data_Date + ' ' + Device_Setup + ' Flowrate Dependency
                Chart.png'
                plt.savefig(File_Name, dpi=Figure_Resolution)
        else:
                plt.show()
        return
        ()



# Makes Directory,
if needed
if
Figure_Sho
w == 0:
        mkdir_p(Data_Date) # Creates a directory for plots, if needed. Equivalent to using mkdir -p on the
        command line

        print('Making Directory Took: {:.0f}s or {:.1f}min'.format((time.time() - Timer), ((time.time() -
        Timer) / 60)))
        Timer =
        time.time()



# Calls in all Baseline files and adds them to allfreq's list
Counter = 0
Sensor_ID = 0 # Identifies which sensor is being down-sampled
Time_Total
= []
Frequency_Baseline
= []
Frequency_Signal =
[]
while Counter < len(os.listdir(Directory_Baseline)) or Counter < len(os.listdir(Directory_Signal)):
        Frequency_Baseline_Holder = []
        Frequency_Signal_Holder = []

        # Grabs Baseline Data from file location
        if Counter < len(os.listdir(Directory_Baseline)):
                path = os.path.join(Directory_Baseline, os.listdir(Directory_Baseline)[Counter])
```

```
        Holder_File_Name = os.listdir(Directory_Baseline)[Counter]
        if "F.bin" in Holder_File_Name: # Change between AF, BF, CF, F
            Frequency_Baseline_Holder = numpy.append(Frequency_Baseline_Holder,
            numpy.fromfile(path, dtype='>f8'))

            # Seperates file name into start date and start time
            Time = int(Holder_File_Name[9:15])
            if Counter == 0:



            # Determines the difference, in hours, between sequential files (converted to
            seconds)
            Time_Difference_Hour = math.floor((Time - Time_Start) / 10000) * 3600 #
            Hour difference in file name

            # Determines the difference, in minutes, between sequential files (converted to
            seconds)
            Time_Difference_Minute = ((math.floor((Time - Time_Start) / 100) / 100) -
            math.floor(math.floor((Time - Time_Start) / 100) / 100)) # Minute difference in
            file name
            if Time_Difference_Minute > 0.6:

            else:



            # Determines the difference, in seconds, between sequential files
            Time_Difference_Second = (((Time - Time_Start) / 100) - (math.floor((Time -
            Time_Start) / 100))) # Second difference in file name
            if Time_Difference_Second > 0.6:

            else:



            # Adds the file Hour, Minute, and second differences to find the time, in seconds,
            between between sequential files
            Time_Difference = Time_Difference_Hour + Time_Difference_Minute +
            Time_Difference_Second

    # Grabs Signal Data from file location
    if Counter < len(os.listdir(Directory_Signal)):
        path = os.path.join(Directory_Signal, os.listdir(Directory_Signal)[Counter])
        Holder_File_Name = os.listdir(Directory_Signal)[Counter]
        if "F.bin" in Holder_File_Name: # Change between AF, BF, CF, F
```

```
                    Frequency_Signal_Holder = numpy.append(Frequency_Signal_Holder,
                    numpy.fromfile(path, dtype='>f8'))

        # Downsampling data, if needed, to make both data sets the same length
        if len(Frequency_Baseline_Holder) > len(Frequency_Signal_Holder) and
        len(Frequency_Signal_Holder) > 1:
                Frequency_Baseline = numpy.append(Frequency_Baseline,
                scipy.signal.resample(Frequency_Baseline_Holder, len(Frequency_Signal_Holder)))
                Frequency_Signal = numpy.append(Frequency_Signal, Frequency_Signal_Holder)
                # Adds time (s) from saved file's time point
                Time_Total = numpy.append(Time_Total, (numpy.cumsum(1.0 /
                Frequency_Signal_Holder) + Time_Difference))

        elif len(Frequency_Baseline_Holder) < len(Frequency_Signal_Holder) and
        len(Frequency_Baseline_Holder) != 0:
                Frequency_Baseline = numpy.append(Frequency_Baseline,
                Frequency_Baseline_Holder)
                Frequency_Signal = numpy.append(Frequency_Signal,
                scipy.signal.resample(Frequency_Signal_Holder, len(Frequency_Baseline_Holder)))
                # Adds time (s) from saved file's time point
                Time_Total = numpy.append(Time_Total, (numpy.cumsum(1.0 /
                Frequency_Baseline_Holder) + Time_Difference))
                Sensor_ID = 1
        elif len(Frequency_Baseline_Holder) == len(Frequency_Signal_Holder) and
        len(Frequency_Baseline_Holder) != 0:
                Frequency_Baseline = numpy.append(Frequency_Baseline,
                Frequency_Baseline_Holder)
                Frequency_Signal = numpy.append(Frequency_Signal, Frequency_Signal_Holder)
                # Adds time (s) from saved file's time point
                Time_Total = numpy.append(Time_Total, (numpy.cumsum(1.0 /
                Frequency_Signal_Holder) + Time_Difference))

        Counter = Counter + 1

if Sensor_ID
== 1:
        print('Importing and Down-Sampling Baseline Data Took: {:.0f}s or
        {:.1f}min'.format((time.time() - Timer), ((time.time() - Timer) / 60)))
        Timer =
        time.time()
else
:
        print('Importing and Down-Sampling Signal Data Took: {:.0f}s or {:.1f}min'.format((time.time()
        - Timer), ((time.time() - Timer) / 60)))
        Timer =
        time.time()
print('There are {} data points, for {:.0f} seconds ({:.2f} hours) of data'.format(len(Frequency_Baseline),
Time_Total[len(Time_Total) - 1], (Time_Total[len(Time_Total) - 1] / 3600)))
```

```
# Converts frequancy signal to density measurement using called calibration
Calibration = Calibration_Baseline
Density_Baseline_Unfiltered = [Bulk_Calibration(x) for x in Frequency_Baseline]
Calibration = Calibration_Signal
Density_Signal_Unfiltered = [Bulk_Calibration(x) for x in Frequency_Signal]


# Filter Signal and
Baseline
Density_Baseline_Filtered = Low_Pass_Filter(Density_Baseline_Unfiltered, 474, 3, 1001)
Density_Signal_Filtered = Low_Pass_Filter(Density_Signal_Unfiltered, 474, 3, 1001)


print('Converting and Filtering Data Took: {:.0f}s or {:.1f}min'.format((time.time() - Timer),
((time.time() - Timer) / 60)))
Timer =
time.time()



# Finds Locations where solution reaches or fills the sensor. Finds Average and Standard Deviation of
Signal and Baseline
Counter = 0
Counter_Flowrate =
0
Counter_Flowrate_
Fill = 0
Holder_Flowrate_Time_Sensor = []
Holder_Flowrate_Time_Sensor_Fill = []
Holder_Flowrate_L
ocation = 0
Holder_Time_Start
= 0
Holder_Time_End
= 0
while Counter < len(Time_Total):
    # Determines the data location when the sensor is filled with new solution
    if (Time_Total[Counter] / 3600) >= Flowrate_Time_Sensor[Counter_Flowrate] and
    len(Flowrate_Time_Sensor) > 1: # Finds data number that corrisponds to flowrate changes
    reaching the sensor
        Holder_Flowrate_Time_Sensor.append(Holder_Flowrate_Location)
        Counter_Flowrate = Counter_Flowrate + 1
    else:
        Holder_Flowrate_Location = Counter
```

```
        # Determines the data location when the sensor is filled with new solution
        if (Time_Total[Counter] / 3600) >= Flowrate_Time_Sensor_Fill[Counter_Flowrate_Fill] and
        len(Flowrate_Time_Sensor_Fill) > 1: # Finds data number that corrisponds to flowrate changes
        filling the sensor
                Holder_Flowrate_Time_Sensor_Fill.append(Holder_Flowrate_Location)
                Counter_Flowrate_Fill = Counter_Flowrate_Fill + 1
                if Counter_Flowrate_Fill == len(Flowrate_Time_Sensor_Fill):
                        break
        else:
                Holder_Flowrate_Location = Counter


        # Finds start and end location of average start time specified in experiment (usually 45 minutes).
        Used to determine the noise in the sensors.
        if (Time_Total[Counter] / 3600) >= Flowrate_Time_Changed[1] and Holder_Time_Start == 0:
                Holder_Time_Start = Counter
        elif (Time_Total[Counter] / 3600) >= (Flowrate_Time_Changed[1] + 10/60) and
        Holder_Time_End == 0:
                Holder_Time_End = Counter
        Counter = Counter + 1
Holder_Flowrate_Time_Sensor.append(len(Time_Total))
Holder_Flowrate_Time_Sensor_Fill.append(len(Time_Total))


Baseline_Noise_Average =
numpy.average(Density_Baseline_Filtered[Holder_Time_Start:Holder_Time_End])
Baseline_Noise_StdDev =
numpy.std(Density_Baseline_Filtered[Holder_Time_Start:Holder_Time_End])
print('Baseline StdDev x 2 ({:.0f}:{:.0f}): {:.2E}'.format((Flowrate_Time_Changed[1]*60),
((Flowrate_Time_Changed[1]*60) + 10), Baseline_Noise_StdDev*2))
Signal_Noise_Average =
numpy.average(Density_Signal_Filtered[Holder_Time_Start:Holder_Time_End])
Signal_Noise_StdDev = numpy.std(Density_Signal_Filtered[Holder_Time_Start:Holder_Time_End])
print('Signal StdDev x 2 ({:.0f}:{:.0f}): {:.2E}'.format((Flowrate_Time_Changed[1]*60),
((Flowrate_Time_Changed[1]*60) + 10), Signal_Noise_StdDev*2))



# Finds Difference between Signal and Baseline
Counter = 0
Counter_Flowrate =
0
Difference_Frequency_Filtered = []
Extraction_
Rate = []
while Counter < len(Density_Signal_Filtered) and Counter_Flowrate <
len(Holder_Flowrate_Time_Sensor):
```

```
        Difference_Frequency_Filtered.append((Density_Signal_Filtered[Counter] -
        Signal_Noise_Average))# - (Density_Baseline_Filtered[Counter] - Baseline_Noise_Average))
        if Counter >= Holder_Flowrate_Time_Sensor[Counter_Flowrate]:
                Volume_Protein = Concentration * Flowrate[Counter_Flowrate] * PSV / 1000 # Protein
                Volume(uL/h) = Concentration(g/L) * Flowrate(uL/h) * Specific Volume(mL/g) / 1000
                (mL/L)
                Volume_Solvent_Initial = Flowrate[Counter_Flowrate] - Volume_Protein # Solvent
                Volume(uL/h) = Flowrate(uL/h) - Protein Volume(uL)
                Counter_Flowrate = Counter_Flowrate + 1


        Volume_Solvent_Extracted_1 = (-Difference_Frequency_Filtered[Counter]) * (Volume_Protein
        ** 2 + 2 * Volume_Protein * Volume_Solvent_Initial + Volume_Solvent_Initial ** 2)
        Volume_Solvent_Extracted_2 = Volume_Protein * ((-Difference_Frequency_Filtered[Counter]) -
        ((1 / PSV) - SD)) + Volume_Solvent_Initial * (-Difference_Frequency_Filtered[Counter])
        Volume_Solvent_Extracted = (-1 * Volume_Solvent_Extracted_1 /
        Volume_Solvent_Extracted_2) * 60
        Extraction_Rate.append(Volume_Solvent_Extracted)
        Counter = Counter + 1



# Finds Averages and Standard Deviations of the Data Sets and their Conversions
Counter = 1
Counter_Dat
a = 0
Baseline_Noise_Av
erage = []
Baseline_Noise_Std
Dev = []
Signal_Noise_Aver
age = []
Signal_Noise_StdD
ev = []
Difference_Frequency_Filtered_Average = []
Difference_Frequency_Filtered_StdDev = []
Extraction_Rate_Av
erage = []
Extraction_Rate_St
dDev = []

Counter_Baseline =
0
Baseline_Noise_Average_B = []
Baseline_Noise_Std
Dev_B = []
Signal_Noise_Aver
age_B = []
```

```
Signal_Noise_StdD
ev_B = []
Difference_Frequency_Filtered_Average_B = []
Difference_Frequency_Filtered_StdDev_B = []
Extraction_Rate_Average_B = []
Extraction_Rate_St
dDev_B = []

if
Figure_Sho
w == 0:
        File_Name = 'Data_Date.txt'
        Path = Data_Date + '/' + File_Name
        fname = open(Path, 'w')

while Counter <= (len(Holder_Flowrate_Time_Sensor) - 2):
        # Uses most recent baseline run
        if Counter %
        2 != 1:
                Baseline_Noise_Average.append(numpy.average(Density_Baseline_Filtered[(Holder_Fl
                owrate_Time_Sensor[Counter] -
                Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))# -
                (Baseline_Noise_Average_B[Counter_Baseline - 1] - Baseline_Noise_Average_B[0]))
                Baseline_Noise_StdDev.append(numpy.std(Density_Baseline_Filtered[(Holder_Flowrat
                e_Time_Sensor[Counter] - Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))# -
                (Baseline_Noise_StdDev_B[Counter_Baseline - 1] - Baseline_Noise_StdDev_B[0]))
                Signal_Noise_Average.append(numpy.average(Density_Signal_Filtered[(Holder_Flowr
                ate_Time_Sensor[Counter] - Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))# -
                (Signal_Noise_Average_B[Counter_Baseline - 1] - Signal_Noise_Average_B[0]))
                Signal_Noise_StdDev.append(numpy.std(Density_Signal_Filtered[(Holder_Flowrate_Ti
                me_Sensor[Counter] - Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))# -
                (Signal_Noise_StdDev_B[Counter_Baseline - 1] - Signal_Noise_StdDev_B[0]))
                Difference_Frequency_Filtered_Average.append(numpy.average(Difference_Frequency
                _Filtered[(Holder_Flowrate_Time_Sensor[Counter] -
                Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))# -
                (Difference_Frequency_Filtered_Average_B[Counter_Baseline - 1] -
                Difference_Frequency_Filtered_Average_B[0]))
                Difference_Frequency_Filtered_StdDev.append(numpy.std(Difference_Frequency_Filte
                red[(Holder_Flowrate_Time_Sensor[Counter] -
                Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))# -
                (Difference_Frequency_Filtered_StdDev_B[Counter_Baseline - 1] -
                Difference_Frequency_Filtered_StdDev_B[0]))
                Extraction_Rate_Average.append(numpy.average(Extraction_Rate[(Holder_Flowrate_Ti
                me_Sensor[Counter] - Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))# -
                (Extraction_Rate_Average_B[Counter_Baseline - 1] - Extraction_Rate_Average_B[0]))
                Extraction_Rate_StdDev.append(numpy.std(Extraction_Rate[(Holder_Flowrate_Time_S
                ensor[Counter] - Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))# -
                (Extraction_Rate_StdDev_B[Counter_Baseline - 1] - Extraction_Rate_StdDev_B[0]))
                if Figure_Show == 0:
```

```
                fname.write('Baseline Average ({:.0f} - {:.2f}:{:.2f}, Flowrate: {:.0f}): {:.4f} +/-
                {:.5f}, Signal Average: {:.4f} +/- {:.5f}\n'.format(Counter,
                Flowrate_Time_Sensor[Counter], Flowrate_Time_Sensor[Counter],
                (Flowrate[Counter - 1]), Baseline_Noise_Average[Counter_Data],
                Baseline_Noise_StdDev[Counter_Data], Signal_Noise_Average[Counter_Data],
                Signal_Noise_StdDev[Counter_Data]))
                fname.write('Difference Average ({:.0f} - {:.2f}:{:.2f}, Flowrate: {:.0f}): {:.4f}
                +/- {:.5f}\n'.format(Counter, Flowrate_Time_Sensor[Counter],
                Flowrate_Time_Sensor[Counter], (Flowrate[Counter - 1]),
                Difference_Frequency_Filtered_Average[Counter_Data],
                Difference_Frequency_Filtered_StdDev[Counter_Data]))
                fname.write('Extraction Rate Average ({:.0f} - {:.2f}:{:.2f}, Flowrate: {:.0f}):
                {:.1f} +/- {:.2f}\n\n'.format(Counter, Flowrate_Time_Sensor[Counter],
                Flowrate_Time_Sensor[Counter], (Flowrate[Counter - 1]),
                Extraction_Rate_Average[Counter_Data],
                Extraction_Rate_StdDev[Counter_Data]))
            else:
                print('Baseline Average ({:.0f} - {:.2f}:{:.2f}, Flowrate: {:.0f}): {:.4f} +/- {:.5f},
                Signal Average: {:.4f} +/- {:.5f}'.format(Counter,
                Flowrate_Time_Sensor[Counter], Flowrate_Time_Sensor[Counter],
                (Flowrate[Counter - 1]), Baseline_Noise_Average[Counter_Data],
                Baseline_Noise_StdDev[Counter_Data], Signal_Noise_Average[Counter_Data],
                Signal_Noise_StdDev[Counter_Data]))
                print('Difference Average ({:.0f} - {:.2f}:{:.2f}, Flowrate: {:.0f}): {:.4f} +/-
                {:.5f}'.format(Counter, Flowrate_Time_Sensor[Counter],
                Flowrate_Time_Sensor[Counter], (Flowrate[Counter - 1]),
                Difference_Frequency_Filtered_Average[Counter_Data],
                Difference_Frequency_Filtered_StdDev[Counter_Data]))
                print('Extraction Rate Average ({:.0f} - {:.2f}:{:.2f}, Flowrate: {:.0f}): {:.1f} +/-
                {:.2f}'.format(Counter, Flowrate_Time_Sensor[Counter],
                Flowrate_Time_Sensor[Counter], (Flowrate[Counter - 1]),
                Extraction_Rate_Average[Counter_Data],
                Extraction_Rate_StdDev[Counter_Data]))

            Counter_Data = Counter_Data + 1

    # Baseline run average and standard deviations
    else:
        Baseline_Noise_Average_B.append(numpy.average(Density_Baseline_Filtered[(Holder
        _Flowrate_Time_Sensor[Counter] -
        Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))
        Baseline_Noise_StdDev_B.append(numpy.std(Density_Baseline_Filtered[(Holder_Flow
        rate_Time_Sensor[Counter] - Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))
        Signal_Noise_Average_B.append(numpy.average(Density_Signal_Filtered[(Holder_Flo
        wrate_Time_Sensor[Counter] - Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))
        Signal_Noise_StdDev_B.append(numpy.std(Density_Signal_Filtered[(Holder_Flowrate
        _Time_Sensor[Counter] - Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))
        Difference_Frequency_Filtered_Average_B.append(numpy.average(Difference_Frequen
        cy_Filtered[(Holder_Flowrate_Time_Sensor[Counter] -
        Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))
```

```
                Difference_Frequency_Filtered_StdDev_B.append(numpy.std(Difference_Frequency_Fi
                ltered[(Holder_Flowrate_Time_Sensor[Counter] -
                Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))
                Extraction_Rate_Average_B.append(numpy.average(Extraction_Rate[(Holder_Flowrate
                _Time_Sensor[Counter] - Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))
                Extraction_Rate_StdDev_B.append(numpy.std(Extraction_Rate[(Holder_Flowrate_Tim
                e_Sensor[Counter] - Data_Used):Holder_Flowrate_Time_Sensor[Counter]]))
                if Figure_Show == 0:
                        fname.write('Baseline Average ({:.0f} - {:.2f}:{:.2f}, Flowrate: {:.0f}): {:.4f} +/-
                        {:.5f}, Signal Average: {:.4f} +/- {:.5f}\n'.format(Counter,
                        Flowrate_Time_Sensor[Counter], Flowrate_Time_Sensor[Counter],
                        (Flowrate[Counter - 1]), Baseline_Noise_Average_B[Counter_Baseline],
                        Baseline_Noise_StdDev_B[Counter_Baseline],
                        Signal_Noise_Average_B[Counter_Baseline],
                        Signal_Noise_StdDev_B[Counter_Baseline]))
                        fname.write('Difference Average ({:.0f} - {:.2f}:{:.2f}, Flowrate: {:.0f}): {:.4f}
                        +/- {:.5f}\n'.format(Counter, Flowrate_Time_Sensor[Counter],
                        Flowrate_Time_Sensor[Counter], (Flowrate[Counter - 1]),
                        Difference_Frequency_Filtered_Average_B[Counter_Baseline],
                        Difference_Frequency_Filtered_StdDev_B[Counter_Baseline]))
                        fname.write('Extraction Rate Average ({:.0f} - {:.2f}:{:.2f}, Flowrate: {:.0f}):
                        {:.1f} +/- {:.2f}\n\n'.format(Counter, Flowrate_Time_Sensor[Counter],
                        Flowrate_Time_Sensor[Counter], (Flowrate[Counter - 1]),
                        Extraction_Rate_Average_B[Counter_Baseline],
                        Extraction_Rate_StdDev_B[Counter_Baseline]))
                else:
                        print('Baseline Average ({:.0f} - {:.2f}:{:.2f}, Flowrate: {:.0f}): {:.4f} +/- {:.5f},
                        Signal Average: {:.4f} +/- {:.5f}'.format(Counter,
                        Flowrate_Time_Sensor[Counter], Flowrate_Time_Sensor[Counter],
                        (Flowrate[Counter - 1]), Baseline_Noise_Average_B[Counter_Baseline],
                        Baseline_Noise_StdDev_B[Counter_Baseline],
                        Signal_Noise_Average_B[Counter_Baseline],
                        Signal_Noise_StdDev_B[Counter_Baseline]))
                        print('Difference Average ({:.0f} - {:.2f}:{:.2f}, Flowrate: {:.0f}): {:.4f} +/-
                        {:.5f}'.format(Counter, Flowrate_Time_Sensor[Counter],
                        Flowrate_Time_Sensor[Counter], (Flowrate[Counter - 1]),
                        Difference_Frequency_Filtered_Average_B[Counter_Baseline],
                        Difference_Frequency_Filtered_StdDev_B[Counter_Baseline]))
                        print('Extraction Rate Average ({:.0f} - {:.2f}:{:.2f}, Flowrate: {:.0f}): {:.1f} +/-
                        {:.2f}'.format(Counter, Flowrate_Time_Sensor[Counter],
                        Flowrate_Time_Sensor[Counter], (Flowrate[Counter - 1]),
                        Extraction_Rate_Average_B[Counter_Baseline],
                        Extraction_Rate_StdDev_B[Counter_Baseline]))
                Counter_Baseline = Counter_Baseline + 1


        Counter = Counter + 1


print('Differences, Averages, and Standard Deviations Took: {:.0f}s or {:.1f}min'.format((time.time() -
Timer), ((time.time() - Timer) / 60)))
Timer =
time.time()
```

Plot_Density_Figur
e()
Plot_Flowrate_Dependency_Chart()

Table J.6. Python Code for Chapter 9 Data Figure

```
import matplotlib.pyplot as plt, numpy, time
from matplotlib.ticker import FormatStrFormatter
from matplotlib.ticker import AutoMinorLocator



print 'Start'
Timer =
time.time()

Figure_Show = 0 # 0: Saves figure as a file, 1: Shows figure in python
Figure_Resolution = 300 # DPI, 300 publication minimum or .pdf instead
Legend = 2 # Changes location of legend - 1: top left corner of plot, 2: below plot
File_Type1
= '.png'
File_Type2
= '.tif'
File_Type3
= '.pdf'

Data_Concentration_Vilker = [84, 91, 211, 211, 289, 325, 325, 354, 357, 413, 428, 448]
Data_Pressure_Vilker = [6.399476193, 7.866022821, 44.26304367, 44.52968851, 112.5241231,
132.789131, 132.789131, 189.7178046, 218.3821251, 349.3047422, 374.1027125, 485.2936113]
Fitting_Concentration = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170,
180, 190, 200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310, 320, 330, 340, 350, 360, 370, 380,
390, 400, 410, 420, 430, 440, 450, 460, 470, 480, 490, 500, 510, 520, 530, 540, 550, 560, 570, 580, 590,
600]
Fitting_Pressure_Vilker = [0, 1.808789096, 3.687681784, 5.640834129, 7.672737344, 9.78825226,
11.99264816, 14.29164658, 16.69147092, 19.19890268, 21.8213455, 24.5668981, 27.44443789,
30.46371681, 33.63547179, 36.97155226, 40.48506817, 44.19056206, 48.10421037, 52.24405942,
56.63030372, 61.28561538, 66.23553608, 71.50894595, 77.13862736, 83.16194713, 89.62168692,
96.56706088, 104.0549719, 112.1515742, 120.9342328, 130.494004, 140.9388034, 152.3974979,
165.0252494, 179.0105821, 194.5848583, 212.0351793, 231.7222468, 254.1055622, 279.7797419,
309.5281203, 344.4040703, 385.8583412, 435.9459546, 497.6773928, 575.6469213, 677.2318347,
815.0758671, 1012.809573, 1320.296228, 1866.011857, 3089.674028, 8374.775209, 6947.016297,
20694.50663]
Data_Concentration_Trial1 = [312.4335768, 316.726334, 324.1492126, 331.9283699, 335.1455923]
Data_Pressure_Trial1 = [120.4187429, 128.27506, 139.6817385, 149.6962305, 159.3315949]
Fitting_Pressure_Trial1 = [0, 1.139118866, 2.332148194, 3.583007827, 4.896007063, 6.275894259,
7.727914205, 9.257874743, 10.87222441, 12.5781433, 14.3836498, 16.29772666, 18.33047042,
20.49326958, 22.79901809, 25.26237257, 27.90006423, 30.73127935, 33.77812675, 37.06621633,
40.62538059, 44.4905824, 48.70306733, 53.31184133, 58.3755859, 63.96517, 70.16698729,
77.08745354, 84.85916362, 93.64946936, 103.6726668, 115.2076979, 128.6245161, 144.4245011,
163.3045088, 186.2624242, 214.77939, 251.152645, 299.1473948, 365.3923125, 462.7406993,
619.8185253, 915.9254378, 1682.182137, 8352.866843]
```

Data_Concentration_Trial2 = [220.1661421, 227.3354129, 267.5449532, 301.2960422, 305.3508148, 325.0349169, 325.0349169, 329.7588139, 332.1726304, 337.1078526, 342.1919356, 344.7919207, 339.6308687, 354.2115061, 352.8344626]
Data_Pressure_Trial2 = [56.95837557, 69.08339249, 93.62553987, 113.2974461, 131.9119707, 168.1378754, 184.6302838, 206.2883471, 212.5127495, 235.9097641, 239.0902497, 258.0123522, 296.5135768, 302.9135812, 330.1745154]
Fitting_Pressure_Trial2 = [0, 1.124216179, 2.306901199, 3.55273816, 4.866924007, 6.255241998, 7.724146782, 9.280864748, 10.93351291, 12.69124048, 14.56439834, 16.56474301, 18.70568363, 21.00258296, 23.4731266, 26.13777921, 29.02035277, 32.14872007, 35.55571898, 39.28030945, 43.36907005, 47.8781559, 52.87589323, 58.44626576, 64.69367231, 71.74953138, 79.78162716, 89.00762314, 99.71508553, 112.2919946, 127.2747656, 145.4267428, 167.8723935, 196.3395037, 233.6264886, 284.5836832, 358.4161725, 474.9855098, 686.5115097, 1188.753991, 3897.794771]
Data_Concentration_Trial3 = [190.534939, 201.4968012, 206.7842309, 224.4537433, 227.696484, 253.3143664, 255.3665818, 270.719116, 277.8788133, 285.4275027, 298.9632643, 304.7439854, 310.752665, 344.7389994, 348.5510252, 352.4482982, 368.949731, 368.949731, 373.3193823]
Data_Pressure_Trial3 = [49.09250403, 52.95325592, 56.69166029, 61.15056019, 70.1499181, 81.50107865, 91.8598626, 102.517724, 106.7455219, 113.0124519, 115.2011245, 127.911718, 137.9442488, 145.6657698, 158.7434162, 184.4908942, 196.9432204, 205.1949277, 212.141588]
Fitting_Pressure_Trial3 = [0, 1.844674867, 3.758067169, 5.744089554, 7.806957454, 9.951218955, 12.18178827, 14.50398334, 16.92356814, 19.44680051, 22.08048612, 24.8320398, 27.7095553, 30.72188472, 33.87872954, 37.19074496, 40.66966007, 44.32841659, 48.18132981, 52.24427574, 56.53490979, 61.07292322, 65.88034532, 70.98190104, 76.40543631, 82.18242651, 88.34858777, 94.94461602, 102.0170863, 109.6195541, 117.813914, 126.6720888, 136.2781458, 146.7309733, 158.1476974, 170.6680898, 184.4603184, 199.7285447, 216.7230978, 235.7543068, 257.2116225, 281.5905506, 309.5313897, 341.8762875, 379.755585, 424.722629, 478.9720726, 545.7089234, 629.8055901, 739.0483283, 886.6983705, 1097.33055, 1422.169371, 1988.650896, 3226.106461, 8058.884984]
Data_Concentration_Trial4 = [245.8698687, 249.9712661, 252.0737084, 254.7520235, 258.5987253, 261.9895716, 269.0452211, 272.717503, 287.7745459, 294.1290143, 294.1290143, 295.5794167, 308.5106434, 311.7194273, 309.3066301, 320.8978886, 322.6250839]
Data_Pressure_Trial4 = [54.6587339, 60.53142502, 67.27414867, 72.87496216, 77.29307611, 82.49965757, 91.9204439, 97.86111046, 102.3472033, 107.7576973, 114.7994984, 122.4122614, 135.1166117, 142.4309868, 151.6422483, 160.3895613, 166.9806761]
Fitting_Pressure_Trial4 = [0, 1.058161728, 2.171271931, 3.343724765, 4.580395743, 5.886709508, 7.268719376, 8.733201132, 10.28776412, 11.94098352, 13.70255856, 15.58350298, 17.59637549, 19.75556058, 22.07761275, 24.5816818, 27.29004217, 30.2287573, 33.42852113, 36.92573418, 40.76389444, 44.99541584, 49.68403609, 54.90804925, 60.76471293, 67.37635967, 74.89903482, 83.53496909, 93.55103112, 105.306795, 119.2986242, 136.2315545, 157.1418313, 183.6172842, 218.2206663, 265.373589, 333.4152042, 440.1753205, 631.8449565, 1076.598279, 3249.713938]
Data_Concentration_Trial5 = [309.9478805, 316.3389845, 321.3079961, 328.1813698, 329.9459084, 335.35524, 337.1979822, 342.8497533, 346.7240449, 358.890721, 363.1382735, 361.0020035, 365.2999772, 367.4875716, 371.9423156]
Data_Pressure_Trial5 = [108.247732, 120.6583, 115.831968, 128.932012, 142.032056, 153.063672, 174.437428, 188.916424, 206.8428, 216.495464, 215.805988, 224.0797, 233.042888, 244.074504, 259.242976]
Fitting_Pressure_Trial5 = [0, 1.126741988, 2.306731443, 3.543834259, 4.842299841, 6.206809871, 7.642534716, 9.155198891, 10.75115735, 12.43748473, 14.22208018, 16.11379112, 18.1225599, 20.25959863, 22.5375985, 24.9709821, 27.57620899, 30.37214858, 33.38053783, 36.62654757, 40.13948836, 43.95369814, 48.10966829, 52.65548678, 57.64870724, 63.15879857, 69.27039679, 76.08768323, 83.74037238, 92.39204465, 102.2519702, 113.5922587, 126.7733601, 142.2830811, 160.7982836, 183.286298, 211.1794647, 246.6927217, 293.4426564, 357.7646107, 451.8587013, 602.6198172, 883.2946711, 1588.934742, 6692.832277]

```
Data_Concentration_Trial6 = [324.4112007, 328.3228737, 332.5144989, 335.3063127, 339.1024821,
344.960689, 349.9994041, 353.0939126, 355.7147799]
Data_Pressure_Trial6 = [143.2619759, 170.8442654, 165.8832239, 189.0865503, 175.5818324,
193.2693497, 214.0393256, 187.6113965, 218.1285628]
Fitting_Pressure_Trial6 = [0, 1.646990969, 3.364164164, 5.156103196, 7.027799695, 8.984699755,
11.03275685, 13.17849233, 15.42906481, 17.79234991, 20.27703237, 22.89271271, 25.65003113,
28.56081227, 31.63823467, 34.89703025, 38.35372018, 42.02689499, 45.93754906, 50.10948212,
54.56978404, 59.34942369, 64.48396908, 70.01447398, 75.98857797, 82.46188224, 89.49968527,
97.17919334, 105.5923646, 114.8496089, 125.0846608, 136.4610816, 149.1810661, 163.4975649,
179.7312773, 198.2949603, 219.7290144, 244.7549593, 274.3582532, 309.921129, 353.444637,
407.9385903, 478.148898, 572.0202419, 703.9372101, 902.9079006, 1237.479392, 1917.926838,
4054.532688]
Fitting_Pressure_Aggregate = [0, 1.622559937, 3.314889929, 5.081589021, 6.927669551, 8.858604634,
10.88038237, 12.99956784, 15.22337437, 17.55974553, 20.01744992, 22.6061912, 25.33673608,
28.22106405, 31.27254304, 34.50613658, 37.93864924, 41.58901871, 45.47866533, 49.63191279,
54.07649722, 58.84418738, 63.97154522, 69.5008653, 75.4813441, 81.97054784, 89.03627147,
96.75891609, 105.234562, 114.5789863, 124.9329837, 136.4695086, 149.4034147, 164.0049601,
180.6188969, 199.6920384, 221.814045, 247.7794695, 278.6852167, 316.089482, 362.2827281,
420.7751094, 497.2330239, 601.4352098, 751.830274, 987.8813763, 1411.901855, 2396.957842,
7233.949309]



def
Plot_Figure(
):
    Alpha  = 0.6 # Changes transparency
    Marker_Color = ['#e41a1c', '#ff7f00', '#ffff33', '#4daf4a', '#377eb8', '#984ea3', 'k'] # Marker Colors in
    hexcode
    Marker_Size = 7
    Marker_Edge_Width = 1


    fig = plt.figure()
    ax0 = plt.subplot()
    plt.title('Bovine Serum Albumin', fontsize=12)
    ax0.plot(Fitting_Concentration[:(len(Fitting_Pressure_Vilker)-1)],
    Fitting_Pressure_Vilker[:(len(Fitting_Pressure_Vilker)-1)], color=Marker_Color[6], linestyle='-',
    linewidth=2.0, label='FSB Model - Vilker et al.')
    ax0.plot(Fitting_Concentration[:(len(Fitting_Pressure_Aggregate)-1)],
    Fitting_Pressure_Aggregate[:(len(Fitting_Pressure_Aggregate)-1)], color=Marker_Color[0],
    dashes=[10, 10, 10, 10], linewidth=2.0, label='FSB Model - Aggregate') # dashes: "fill, space, fill,
    space", must be even number
    ax0.plot(Data_Concentration_Vilker[::], Data_Pressure_Vilker[::], color=Marker_Color[6],
    marker='s', fillstyle='none', markersize=Marker_Size, markeredgewidth=Marker_Edge_Width,
    linestyle='None')
    ax0.plot(Data_Concentration_Vilker[::], Data_Pressure_Vilker[::], color=Marker_Color[6],
    marker='s', alpha=Alpha, markersize=Marker_Size, markeredgewidth=Marker_Edge_Width,
    linestyle='None')
    ax0.plot([], [], color=Marker_Color[6], marker='s', alpha=1, markersize=Marker_Size,
    markeredgewidth=Marker_Edge_Width, linestyle='None', label='Vilker et al. (1980)')
```

```
ax0.plot([], [], color=Marker_Color[0], marker='o', alpha=1, markersize=Marker_Size,
markeredgewidth=Marker_Edge_Width, linestyle='None', label='Trial 1')
ax0.plot(Data_Concentration_Trial1[::], Data_Pressure_Trial1[::], color=Marker_Color[0],
marker='o', fillstyle='none', markersize=Marker_Size, markeredgewidth=Marker_Edge_Width,
linestyle='None')
ax0.plot(Data_Concentration_Trial1[::], Data_Pressure_Trial1[::], color=Marker_Color[0],
marker='o', alpha=Alpha, markersize=Marker_Size, markeredgewidth=Marker_Edge_Width,
linestyle='None')
ax0.plot([], [], color=Marker_Color[3], marker='o', alpha=1, markersize=Marker_Size,
markeredgewidth=Marker_Edge_Width, linestyle='None', label='Trial 4')
ax0.plot([], [], color=Marker_Color[1], marker='o', alpha=1, markersize=Marker_Size,
markeredgewidth=Marker_Edge_Width, linestyle='None', label='Trial 2')
ax0.plot(Data_Concentration_Trial2[::], Data_Pressure_Trial2[::], color=Marker_Color[1],
marker='o', fillstyle='none', markersize=Marker_Size, markeredgewidth=Marker_Edge_Width,
linestyle='None')
ax0.plot(Data_Concentration_Trial2[::], Data_Pressure_Trial2[::], color=Marker_Color[1],
marker='o', alpha=Alpha, markersize=Marker_Size, markeredgewidth=Marker_Edge_Width,
linestyle='None')
ax0.plot([], [], color=Marker_Color[4], marker='o', alpha=1, markersize=Marker_Size,
markeredgewidth=Marker_Edge_Width, linestyle='None', label='Trial 5')
ax0.plot([], [], color=Marker_Color[2], marker='o', alpha=1, markersize=Marker_Size,
markeredgewidth=Marker_Edge_Width, linestyle='None', label='Trial 3')
ax0.plot(Data_Concentration_Trial3[::], Data_Pressure_Trial3[::], color=Marker_Color[2],
marker='o', fillstyle='none', markersize=Marker_Size, markeredgewidth=Marker_Edge_Width,
linestyle='None')
ax0.plot(Data_Concentration_Trial3[::], Data_Pressure_Trial3[::], color=Marker_Color[2],
marker='o', alpha=Alpha, markersize=Marker_Size, markeredgewidth=Marker_Edge_Width,
linestyle='None')
ax0.plot(Data_Concentration_Trial4[::], Data_Pressure_Trial4[::], color=Marker_Color[3],
marker='o', fillstyle='none', markersize=Marker_Size, markeredgewidth=Marker_Edge_Width,
linestyle='None')
ax0.plot(Data_Concentration_Trial4[::], Data_Pressure_Trial4[::], color=Marker_Color[3],
marker='o', alpha=Alpha, markersize=Marker_Size, markeredgewidth=Marker_Edge_Width,
linestyle='None')
ax0.plot(Data_Concentration_Trial5[::], Data_Pressure_Trial5[::], color=Marker_Color[4],
marker='o', fillstyle='none', markersize=Marker_Size, markeredgewidth=Marker_Edge_Width,
linestyle='None')
ax0.plot(Data_Concentration_Trial5[::], Data_Pressure_Trial5[::], color=Marker_Color[4],
marker='o', alpha=Alpha, markersize=Marker_Size, markeredgewidth=Marker_Edge_Width,
linestyle='None')
ax0.plot([], [], color=Marker_Color[5], marker='o', alpha=1, markersize=Marker_Size,
markeredgewidth=Marker_Edge_Width, linestyle='None', label='Trial 6')
ax0.plot(Data_Concentration_Trial6[::], Data_Pressure_Trial6[::], color=Marker_Color[5],
marker='o', fillstyle='none', markersize=Marker_Size, markeredgewidth=Marker_Edge_Width,
linestyle='None')
ax0.plot(Data_Concentration_Trial6[::], Data_Pressure_Trial6[::], color=Marker_Color[5],
marker='o', alpha=Alpha, markersize=Marker_Size, markeredgewidth=Marker_Edge_Width,
linestyle='None')

plt.minorticks_on()
ax0.xaxis.set_major_formatter(FormatStrFormatter('%.0f')) # Sets decimals to show on x-axis label
```

```
    ax0.yaxis.set_major_formatter(FormatStrFormatter('%.0f')) # Sets decimals to show on y-axis label
    plt.setp(ax0.get_xticklabels(), fontsize=10) # Sets x-axis label text size
    plt.setp(ax0.get_yticklabels(), fontsize=10) # Sets y-axis label text size
    ax0.tick_params(which='minor', top=False, right=False) # Removes minor ticks from edges of figure
    ax0.tick_params(which='major', top=False, right=False) # Removes major ticks from edges of figure
    ax0.xaxis.set_minor_locator(AutoMinorLocator(2)) # number of minor ticks on x-axis
    ax0.yaxis.set_minor_locator(AutoMinorLocator(2)) # number of minor ticks on y-axis
    ax0.set_xlim(0, 400) # Sets x-axis limits (min, max)
    ax0.set_ylim(0, 350) # Sets y-axis limits (min, max)
    ax0.set_xlabel('Protein Concentration (g $\mathdefault{L^{-1}}$)', fontsize=12)
    ax0.set_ylabel('Osmotic Pressure (kPa)', fontsize=12)

    if Legend == 1:
                plt.legend(loc=2, fontsize=10, numpoints=1)
    elif Legend == 2:
                # Shrink current axis by 10%
                box = ax0.get_position()
                ax0.set_position([box.x0, box.y0 + box.height * 0.1, box.width, box.height * 0.9])

                ax0.legend(loc='upper center', fontsize=10, numpoints=1, bbox_to_anchor=(0.5, -0.15),
                fancybox=True, frameon=False, handlelength=3, ncol=4) # ncol: number of columns

    if Figure_Show == 0:
                File_Name = 'Concentrating Osmometer Data'
                plt.savefig(File_Name + File_Type1, dpi=Figure_Resolution, bbox_inches='tight')
                plt.savefig(File_Name + File_Type2, dpi=Figure_Resolution, bbox_inches='tight')
                plt.savefig(File_Name + File_Type3, dpi=Figure_Resolution, bbox_inches='tight')
                plt.savefig(File_Name, dpi=Figure_Resolution, bbox_inches='tight')
    else:
                plt.show()
    return ()

print('Inputting Data Took: {:.0f}s or {:.1f}min'.format((time.time() - Timer), ((time.time() - Timer) /
60)))
Timer =
time.time()

Plot_Figure(
)

print('Plotting the Figure Took: {:.0f}s or {:.1f}min'.format((time.time() - Timer), ((time.time() - Timer)
/ 60)))
```

```
Timer =
time.time()
```

REFERENCES

1. Christopher and Dana Reeve Foundation.

2. AlexWillWalkAgain.com.

3. McBride, D. W., Hsu, M. S., Rodgers, V. G. J. & Binder, D. K. Improved survival following cerebral edema using a novel hollow fiber-hydrogel device. *J. Neurosurg.* **116,** 1389–1394 (2012).

4. McBride, D. W. *et al.* Reduction of Cerebral Edema after Traumatic Brain Injury Using an Osmotic Transport Device. *J. Neurotrauma* **31,** 1948–1954 (2014).

5. McBride, D. W. *et al.* Reduction of Cerebral Edema via an Osmotic Transport Device Improves Functional Outcome after Traumatic Brain Injury in Mice. *Acta Neurochir. Suppl.* **121,** 285–289 (2016).

6. faculty.washington.edu/chudler/facts.html.

7. sigmaaldrich.com/content/dam/sigma-aldrich/docs/Sigma/Product_Information_Sheet/a2153pis.pdf.

8. Soubeyrand, M., Badner, A., Vawda, R., Chung, Y. S. & Fehlings, M. G. Very High Resolution Ultrasound Imaging for Real-Time Quantitative Visualization of Vascular Disruption after Spinal Cord Injury. *J. Neurotrauma* **31,** 1767–1775 (2014).

9. Jain, A., Kim, Y. T., McKeon, R. J. & Bellamkonda, R. V. In situ gelling hydrogels for conformal repair of spinal cord defects, and local delivery of BDNF after spinal cord injury. *Biomaterials* **27,** 497–504 (2006).

10. McBride, D. W. & Rodgers, V. G. J. Obtaining protein solvent accessible surface area when structural data is unavailable using osmotic pressure. *AIChE J.* **58,** 1012–1017 (2012).

11. McBride, D. W. & Rodgers, V. G. J. Interpretation of negative second virial coefficients from non-attractive protein solution osmotic pressure data: An alternate perspective. *Biophys. Chem.* **184,** 79–86 (2013).

12. Yousef, M. A., Datta, R. & Rodgers, V. G. J. Free-solvent model of osmotic pressure revisited: Application to concentrated IgG solution under physiological conditions. *J. Colloid Interface Sci.* **197,** 108–118 (1998).

13. Vilker, V. L., Colton, C. K. & Smith, K. A. The osmotic pressure of concentrated protein solutions: Effect of concentration and ph in saline solutions of bovine serum albumin. *J. Colloid Interface Sci.* **79,** 548–566 (1981).

14. Steudle, E. & Stumpf, B. Process and apparatus for the determination of the concentration of a substance dissolved in a solvent by means of an osmometer. (1991).

15. Myrenne, K.-D. S. High-speed microvolume membrane osmometry. (1972).

16. Boris, D. C. Method for measuring changes in osmotic pressure. (2001).

17. Wu, J. & Prausnitz, J. M. Osmotic pressures of aqueous bovine serum albumin solutions at high ionic strength. *Fluid Phase Equilib.* **155,** 139–154 (1999).

18. McMillan Jr, W. G. & Mayer, J. E. The statistical thermodynamics of multicomponent systems. *J. Chem. Phys.* **13,** 276–305 (1945).

19. Yousef, M. A., Datta, R. & Rodgers, V. G. J. Confirmation of free solvent model assumptions in predicting the osmotic pressure of concentrated globular proteins. *J. Colloid Interface Sci.* **243,** 321–325 (2001).

20. Yousef, M. A., Datta, R. & Rodgers, V. G. J. Model of osmotic pressure for high concentrated binary protein solutions. *AIChE J.* **48,** 913–917 (2002).

21. Hofmeister, F. Zur Lehre von der Wirkung der Salze. *Arch. für Exp. Pathol. und Pharmakologie* **24,** 247–260 (1888).

22. Medda, L. *et al.* Hofmeister Challenges: Ion Binding and Charge of the BSA Protein as Explicit Examples. *Langmuir* **28,** 16355–16363 (2012).

23. Zhang, Y. & Cremer, P. S. Chemistry of Hofmeister anions and osmolytes. *Annu Rev Phys Chem* **61,** 63–83 (2010).

24. Traube, J. The Attraction Pressure. *J. Phys. Chem.* **14,** 452–470 (1909).

25. Chaplin, M. Water Structure and Science. (2000).

26. Omta, A. W., Kropman, M. F., Woutersen, S. & Bakker, H. J. Negligible effect of ions on the hydrogen-bond structure in liquid water. *Science (80-.).* **301,** 347–349 (2003).

27. Funkner, S. *et al.* Watching the Low-Frequency Motions in Aqueous Salt Solutions: The Terahertz Vibrational Signatures of Hydrated Ions. *J. Am. Chem. Soc.* **134,** 1030–1035 (2012).

28. Smith, J. D., Saykally, R. J. & Geissler, P. L. The Effects of Dissolved Halide Anions on Hydrogen Bonding in Liquid Water. *J. Am. Chem. Soc.* **129,** 13847–13856 (2007).

29. Mancinelli, R., Botti, A., Bruni, F., Ricci, M. A. & Soper, A. K. Hydration of Sodium, Potassium, and Chloride Ions in Solution and the Concept of Structure Maker/Breaker. *J. Phys. Chem. B* **111,** 13570–13577 (2007).

30. Tobias, D. J. & Hemminger, J. C. Chemistry - Getting specific about specific ion effects. *Science (80-.).* **319,** 1197–1198 (2008).

31. Zangi, R. Can Salting-In/Salting-Out Ions be Classified as Chaotropes/Kosmotropes? *J. Phys. Chem. B* **114,** 643–650 (2010).

32. Zhang, Y. & Cremer, P. S. Interactions between macromolecules and ions: the Hofmeister series. *Curr. Opin. Chem. Biol.* **10,** 658–663 (2006).

33. Marcus, Y. Electrostriction in Electrolyte Solutions. *Chem. Rev.* **111,** 2761–2783 (2011).

34. Jones, G. & Dole, M. THE VISCOSITY OF AQUEOUS SOLUTIONS OF STRONG ELECTROLYTES WITH SPECIAL REFERENCE TO BARIUM CHLORIDE. *J. Am. Chem. Soc.* **51,** 2950–2964 (1929).

35. Collins, K. D. Charge density-dependent strength of hydration and biological structure. *Biophys. J.* **72,** 65–76 (1997).

36. Poiseuille, J. L. M. Sur le mouvement des liquides de nature différente dans les tubes de trèspetits diamètres. *Ann. Chim. Phys.* 76–110 (1847).

37. Collins, K. D. & Washabaugh, M. W. The Hofmeister Effect and the Behavior of Water at Interfaces. *Q. Rev. Biophys.* **18,** 323–422 (1985).

38. Collins, K. D. Ions from the Hofmeister series and osmolytes: Effects on proteins in solution and in the crystallization process. *Methods* **34,** 300–311 (2004).

39. Zangi, R., Hagen, M. & Berne, B. J. Effect of ions on the hydrophobic interaction between two plates. *J. Am. Chem. Soc.* **129,** 4678–4686 (2007).

40. Waldron, T. T., Modestou, M. A. & Murphy, K. P. Anion binding to a protein-protein complex lacks dependence on net charge. *Protein Sci.* **12,** 871–874 (2003).

41. Ajroud, K. *et al.* Binding affinity of metal ions to the CD11b A-domain is regulated by integrin activation and ligands. *J. Biol. Chem.* **279,** 25483–25488 (2004).

42. Light, T. P., Corbett, K. M., Metrick, M. A. & MacDonald, G. Hofmeister Ion-Induced Changes in Water Structure Correlate with Changes in Solvation of an Aggregated Protein Complex. *Langmuir* **32,** 1360–1369 (2016).

43. Majumdar, R. *et al.* Effects of salts from the Hofmeister series on the conformational stability, aggregation propensity, and local flexibility of an IgG1 monoclonal antibody. *Biochemistry* **52,** 3376–3389 (2013).

44. Okur, H. I. *et al.* Beyond the Hofmeister Series: Ion-Specific Effects on Proteins and Their Biological Functions. *J Phys Chem B* **121,** 1997–2014 (2017).

45. Lo Nostro, P. & Ninham, B. W. Hofmeister phenomena: an update on ion specificity in biology. *Chem Rev* **112,** 2286–2322 (2012).

46. Ries-Kautt, M. M. & Ducruix, A. F. Relative effectiveness of various ions on the

solubility and crystal growth of lysozyme. *J. Biol. Chem.* **264,** 745–748 (1989).

47. Paterova, J. *et al.* Reversal of the Hofmeister Series: Specific Ion Effects on Peptides. *J. Phys. Chem. B* **117,** 8150–8158 (2013).

48. Nandi, P. K. & Robinson, D. R. Effects of Salts on Free-Energy of Peptide Group. *J. Am. Chem. Soc.* **94,** 1299- (1972).

49. Nandi, P. K. & Robinson, D. R. Effects of Salts on Free-Energies of Nonpolar Groups in Model Peptides. *J. Am. Chem. Soc.* **94,** 1308- (1972).

50. Algaer, E. A. & van der Vegt, N. F. A. Hofmeister Ion Interactions with Model Amide Compounds. *J. Phys. Chem. B* **115,** 13781–13787 (2011).

51. Rembert, K. B. *et al.* Molecular mechanisms of ion-specific effects on proteins. *J Am Chem Soc* **134,** 10039–10046 (2012).

52. Jungwirth, P. & Cremer, P. S. Beyond Hofmeister. *Nat Chem* **6,** 261–263 (2014).

53. Linderstrøm-Lang, K. On the Ionization of Proteins. *Compt. Rend. Trav. Lab. Carlsb.* **15,** 1–29 (1924).

54. Debye, P. & Hückel, E. De La Theorie Des Electrolytes. I. Abaissement Du Point De Congelation et Phenomenes Associes. *Phys. Zeitschrift* **24,** 185–206 (1923).

55. Tanford, C. Preparation and Properties of Serum and Plasma Proteins. XXIII. Hydrogen Ion Equilibria in Native and Modified Human Serum Albumin. *J. Am. Chem. Soc.* **72,** 441–451 (1950).

56. Scatchard, G. & Black, E. S. The Effect of Salts on the Isoionic and Isoelectric Points of Proteins. *J. Phys. Colloid Chem.* **53,** 88–99 (1949).

57. Scatchard, G., Scheinberg, I. H. & Armstrong, S. H. Physical Chemistry of Protein Solutions .4. The Combination of Human Serum Albumin with Chloride Ion. *J. Am. Chem. Soc.* **72,** 535–540 (1950).

58. Scatchard, G., Scheinberg, I. H. & Armstrong, S. H. Physical Chemistry of Protein Solutions .5. The Combination of Human Serum Albumin with Thiocyanate Ion. *J. Am. Chem. Soc.* **72,** 540–546 (1950).

59. Fox, J. M. *et al.* Interactions between Hofmeister Anions and the Binding Pocket of a Protein. *J. Am. Chem. Soc.* **137,** 3859–3866 (2015).

60. Ninham, B. W. & Yaminsky, V. Ion Binding and Ion Specificity: The Hofmeister Effect and Onsager and Lifshitz Theories. *Langmuir* **13,** 2097–2108 (1997).

61. Parsons, D. F., Bostrom, M., Lo Nostro, P. & Ninham, B. W. Hofmeister effects: interplay of hydration, nonelectrostatic potentials, and ion size. *Phys Chem Chem Phys* **13,** 12352–

12367 (2011).

62. Zimmerman, S. B. & Trach, S. O. Estimation of macromolecule concentrations and excluded volume effects for the cytoplasm of Escherichia coli. *J. Mol. Biol.* **222,** 599–620 (1991).

63. Reboiras, M. D., Pfister, H. & Pauly, H. Activity-Coefficients of Salts in Highly Concentrated Protein Solutions .2. Potassium-Salts in Isoionic Bovine Serum-Albumin Solutions. *Biophys. Chem.* **24,** 249–257 (1986).

64. Curtis, R. A., Prausnitz, J. M. & Blanch, H. W. Protein-protein and protein-salt interactions in aqueous protein solutions containing concentrated electrolytes. *Biotechnol. Bioeng.* **57,** 11–21 (1998).

65. Lima, E. R. A., Biscaia, E. C., Boström, M., Tavares, F. W. & Prausnitz, J. M. Osmotic Second Virial Coefficients and Phase Diagrams for Aqueous Proteins from a Much-Improved Poisson−Boltzmann Equation. *J. Phys. Chem. C* **111,** 16055–16059 (2007).

66. Moon, Y. U. *et al.* Protein–Protein Interactions in Aqueous Ammonium Sulfate Solutions. Lysozyme and Bovine Serum Albumin (BSA). *Interactions* **29,** 699–717 (2000).

67. Zhou, H.-X., Rivas, G. & Minton, A. P. Macromolecular Crowding and Confinement: Biochemical, Biophysical, and Potential Physiological Consequences. *Annu. Rev. Biophys.* **37,** 375–397 (2008).

68. Yousef, M. A., Datta, R. & Rodgers, V. G. J. Free-solvent model of osmotic pressure revisited: Application to concentrated IgG solution under physiological conditions. *J. Colloid Interface Sci.* **197,** 108–118 (1998).

69. Yousef, M. A., Datta, R. & Rodgers, V. G. J. Monolayer hydration governs nonideality in osmotic pressure of protein solutions. *AIChE J.* **48,** 1301–1308 (2002).

70. Wang, Y. & Rodgers, V. G. J. Free-solvent model shows osmotic pressure is the dominant factor in limiting flux during protein ultrafiltration. *J. Memb. Sci.* **320,** 335–343 (2008).

71. Wang, Y. H. & Rodgers, V. G. J. Determining Fouling-Independent Component of Critical Flux in Protein Ultrafiltration Using the Free-Solvent-Based (FSB) Model. *Aiche J.* **56,** 2756–2759 (2010).

72. Wang, Y. H. & Rodgers, V. G. J. Electrostatic contributions to permeate flux behavior in single bovine serum albumin ultrafiltration. *J. Memb. Sci.* **366,** 184–191 (2011).

73. McBride, D. W. & Rodgers, V. G. J. A generalized free-solvent model for the osmotic pressure of multi-component solutions containing protein-protein interactions. *Math. Biosci.* **253,** 72–87 (2014).

74. Lotz, P. & Frazer, J. C. W. The osmotic pressures of concentrated solutions of sucrose as determined by the water interferometer. *J. Am. Chem. Soc.* **43,** 2501–2507 (1921).

75. Frazer, J. C. W. & Myrick, R. T. The osmotic pressure of sucrose solutions at 30 degrees. *J. Am. Chem. Soc.* **38,** 1907–1922 (1916).

76. Scatchard, G. Physical Chemistry of Protein Solutions .1. Derivation of the Equations for the Osmotic Pressure. *J. Am. Chem. Soc.* **68,** 2315–2319 (1946).

77. Yousef, M. A., Datta, R. & Rodgers, V. G. J. Understanding nonidealities of the osmotic pressure of concentrated bovine serum albumin. *J. Colloid Interface Sci.* **207,** 273–282 (1998).

78. Eagland, D. Nucleic Acids, Peptides, and Proteins. in *Water A Comprehensive Treatise: Aqueous Solutions of Amphiphiles and Macromolecules* (ed. Franks, F.) 305–518 (Springer US, 1975). doi:10.1007/978-1-4684-2958-9_5 LB - Eagland1975

79. Tombs, M. P., Newsom, B. G. & Wilding, P. Protein Solubility - Phase Separation in Arachin-Salt-Water Systems. *Int. J. Pept. Protein Res.* **6,** 253–277 (1974).

80. Ohtaki, H. & Radnai, T. Structure and dynamics of hydrated ions. *Chem. Rev.* **93,** 1157–1204 (1993).

81. Kiriukhin, M. Y. & Collins, K. D. Dynamic hydration numbers for biologically important ions. *Biophys. Chem.* **99,** 155–168 (2002).

82. Zhao, H. Are ionic liquids kosmotropic or chaotropic? An evaluation of available thermodynamic parameters for quantifying the ion kosmotropicity of ionic liquids. *J. Chem. Technol. Biotechnol.* **81,** 877–891 (2006).

83. Rupley, J. A. & Careri, G. Protein hydration and function. *Adv Protein Chem* **41,** 37–172 (1991).

84. Rembert, K. B., Okur, H. I., Hilty, C. & Cremer, P. S. An NH moiety is not required for anion binding to amides in aqueous solution. *Langmuir* **31,** 3459–3464 (2015).

85. Tanford, C., Buzzell, J. G., Rands, D. G. & Swanson, S. A. The Reversible Expansion of Bovine Serum Albumin in Acid Solutions1. *J. Am. Chem. Soc.* **77,** 6421–6428 (1955).

86. Morris, A. M., Watzky, M. A. & Finke, R. G. Protein aggregation kinetics, mechanism, and curve-fitting: A review of the literature. *Biochim. Biophys. Acta - Proteins Proteomics* **1794,** 375–397 (2009).

87. Kastelic, M., Kalyuzhnyi, Y. V, Hribar-Lee, B., Dill, K. A. & Vlachy, V. Protein aggregation in salt solutions. *Proc. Natl. Acad. Sci.* **112,** 6766 (2015).

88. Chan, P., Curtis, R. A. & Warwicker, J. Soluble expression of proteins correlates with a lack of positively-charged surface. *Sci. Rep.* **3,** (2013).

89. Pierce, B. G. *et al.* ZDOCK server: interactive docking prediction of protein–protein

complexes and symmetric multimers. *Bioinformatics* **30,** 1771–1773 (2014).

90. Bujacz, A. Structures of bovine, equine and leporine serum albumin. *Acta Crystallogr D Biol Crystallogr* **68,** 1278–1289 (2012).

91. Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Res.* **28,** 235–242 (2000).

92. Pettersen, E. F. *et al.* UCSF Chimera--a visualization system for exploratory research and analysis. *J Comput Chem* **25,** 1605–1612 (2004).

93. Alford, J. R., Kendrick, B. S., Carpenter, J. F. & Randolph, T. W. Measurement of the second osmotic virial coefficient for protein solutions exhibiting monomer-dimer equilibrium. *Anal Biochem* **377,** 128–133 (2008).

94. Jiménez, M., Rivas, G. & Minton, A. P. Quantitative Characterization of Weak Self-Association in Concentrated Solutions of Immunoglobulin G via the Measurement of Sedimentation Equilibrium and Osmotic Pressure. *Biochemistry* **46,** 8373–8378 (2007).

95. Scherer, T. M. Cosolute Effects on the Chemical Potential and Interactions of an IgG1 Monoclonal Antibody at High Concentrations. *J. Phys. Chem. B* **117,** 2254–2266 (2013).

96. Imai, T., Harano, Y., Kinoshita, M., Kovalenko, A. & Hirata, F. A theoretical analysis on hydration thermodynamics of proteins. *J. Chem. Phys.* **125,** (2006).

97. Zhou, H. X. Protein folding and binding in confined spaces and in crowded solutions. *J. Mol. Recognit.* **17,** 368–375 (2004).

98. Eggers, D. K. & Valentine, J. S. Crowding and hydration effects on protein conformation: A study with sol-gel encapsulated proteins. *J. Mol. Biol.* **314,** 911–922 (2001).

99. Kawamura, K. *et al.* X-ray and neutron protein crystallographic analysis of the trypsin-BPTI complex. *Acta Crystallogr. Sect. D Biol. Crystallogr.* **67,** 140–148 (2011).

100. Sanner, M. F., Olson, A. J. & Spehner, J. Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers* **38,** 305–320 (1996).

101. Vincent, J. P. & Lazdunski, M. Trypsin-Pancreatic Trypsin Inhibitor Association. Dynamics of the Interaction and Role of Disulfide Bridges. *Biochemistry* **11,** 2967–2977 (1972).

102. McBride, D. W. & Rodgers, V. G. J. Interpretation of negative second virial coefficients from non-attractive protein solution osmotic pressure data: An alternate perspective. *Biophys. Chem.* **184,** 79–86 (2013).