A random walk in physical biology

Thesis by

Eric L. Peterson

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

2008

(Defended May 22, 2008)

To my grandparents

Bob and Emma Lou

Eddie and Bobby

together forever

Acknowledgements

Science is a collaborative effort and I feel greatly indebted to the many individuals who helped me make this journey and have been companions along the way. Many teachers contributed to my enthusiasm for science, including **Meg Cox**, who did such a fantastic job of teaching chemistry that I went to Brigham Young University believing that was my calling in life. To that end I must thank **Barbara Hinshaw**, whose organic chemistry lab during my freshman year disabused me of that misguided notion. Thanks to the required courses in physics and to great teachers like **Ross Spencer, Grant Mason, Scott Bergeson, and Grant Hart** I found myself solidly and happily in the field of physics.

The complexity, robustness, and beauty of biology has long fascinated me and the marriage of physics and biology during my graduate studies has been a happy one. Here at Caltech I am indebted first and foremost to my adviser **Rob Phillips**. From the minute I first met him he has been a source of creative ideas and bottomless enthusiasm for physics and biology. Each of my labmates has also taught me something and made this experience fun, and I want to thank each of them: **Paul Wiggins, Paul Grayson, Heun Jin Lee, Darren Segall, Frosso Seitaridou, Lin Han, Tristan Ursell, Hernan Garcia, Stephanie Johnson, Dave Wu, and Dave Van Valen**. From outside Caltech I also especially want to thank **Jane Kondev** (Brandeis) and **Julie Theriot** (Stanford). I have immensely enjoyed my association with **Bill Klug** (UCLA), a fellow computational jock who also helped rescue me at a group retreat when I bumped my head on a tree during some early attempts at snowboarding!

I would like to thank **Ralf Bundschuh, John Chodera, Ken Dill, Shura Grosberg, Liisa Holm, Chris Myers, Eugene Shakhnovich, John Spouge, Peter Swain, Ned Wingreen, Chris Wiggins, and Jasmine Zhou** for helpful discussions and suggestions on the reduced alphabet project. One of the neatest experiences for me was the chance to be a mentor to **Jessica Su**, a very talented SURF student who made impressive inroads on

the next step in the reduced alphabet project: finding translational motors in prokaryotes! My bet still stands—I think they're there!

The membrane forces project would not have been possible without equipment generously loaned to us by **Steve Quake** and I want to specifically thank **Feng Feng, Grant Jensen, Lin Ma, and Sriram Subramaniam** for their interest and suggestions. **Tristan Ursell** provided the electroformation equipment and protocol for making vesicles and gave many helpful hints in the execution of the membrane experiments. Hearty thanks are due to **Rob Bao** for his diffusion analysis code and helpful insights. Finally a huge thanks to **Paul Wiggins**—he first dreamed of actually doing the experiment to measure forces on membranes from their shape, and **Heun Jin Lee**, who built the microscope and optical tweezers used to perform the experiments, made that dream a reality. Paul, Heun Jin, and I worked together closely throughout the membrane forces project and it has been a pleasure interacting with them both.

Who knew Cheerios and MscL had something in common? Working together with **Tristan Ursell** on the interacting proteins project was not only a lot of fun, it allowed me an opportunity to see how to successfully navigate the treacherous waters of getting a paper from concept to draft, through innumerable reviews and finally to long-awaited publication. What a journey that was! I relished the opportunity to review and apply Monte Carlo techniques as well as demonstrate the virtues of natively compiled code—the C++ simulation was about 10,000 times faster than a MATLAB script! Eat your heart out, MathWorks.

The journey at Caltech has not been a purely scientific one and I am grateful to have found so many good friends here. **Tristan Smith, Nate Bode, and Hernan Garcia** took me under their wing during my first year here, sharing with me their culinary skills and spirit of adventure. Whether it was traipsing through the verdant coasts of Brazil, braving the train system in Europe, or exploring the hidden wonders of LA, Tristan, Nate, and Hernan have been my faithful compadres. Fellow grad students **Jenny Roizen and Jenn Stockdill**, much braver than I, are forging ahead in chemistry and besides being grateful for their friendship I stand in awe of their dedication and pluck. Outside the Caltech community, I consider myself lucky to call **Lucia Cordeiro, Bob Davis, Dave Mortensen, Erin Shepard, and Tamar Hill** my friends and am grateful for some non-nerd influence in my life.

Thanks and love go to my four younger brothers: **Brandon, Neil, Sean, and Brian**. Last of all, I want to thank my **Mom and Dad** for instilling in me a love of science and technology and for inspiring me with their curiosity about the universe.

Abstract

Biology as a scientific discipline is becoming evermore quantitative as tools become available to probe living systems on every scale from the macro to the micro, and now even to the nanoscale. In quantitative biology, the challenge is to understand the living world in an *in vivo* context, where it is often difficult for simple theoretical models to connect with the full richness and complexity of the observed data. Computational models and simulations offer a way to bridge the gap between simple theoretical models and real biological systems; towards that aspiration are presented in this thesis three case studies in applying computational models that may give insight into native biological structures. The first is concerned with soluble proteins; proteins, like DNA, are linear polymers written in a twenty-letter "language" of amino acids. Despite the astronomical number of possible protein sequences, a great amount of similarity is observed among the folded structures of globular proteins. One useful way of discovering similar sequences is to align their sequences, as done, e.g., by the popular BLAST program. By clustering together amino acids and reducing the alphabet that proteins are written in to fewer than twenty letters, we find that pairwise sequence alignments are actually *more* sensitive to proteins with similar structures. The second case study is concerned with the measurement of forces applied to a membrane. We demonstrate a general method for extracting the forces applied to a fluid lipid bilayer of arbitrary shape and show that the subpiconewton forces applied by optical tweezers to vesicles can be accurately measured in this way. In the third and final case study we examine the forces between proteins in a lipid bilayer membrane. Due to the bending of the membrane surrounding them, such proteins feel mutually attractive forces which can help them to self-organize and act in concert. These findings are relevant at the areal densities estimated for membrane proteins such as the MscL mechanosensitive channel. The findings of the analytical studies were confirmed by a Monte Carlo Markov Chain simulation using the fully two-dimensional potentials between two model proteins in a membrane. Living

systems present us with beautiful and intricate structures, from the helices and sheets of a folded protein to the dynamic morphology of cellular organelles and the self-organization of proteins in a biomembrane and a synergy of theoretical and *in silico* approaches should enable us to build and refine models of *in vivo* biological data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Biology finds itself at the opening of this new millenium with one of the most exciting outlooks of any field in science. Techniques to observe, probe, and quantify biological processes now range from the macroscopic down to the level of nanometers and single molecules. As biological systems are probed in ever greater and ever more quantitative detail, the demand for quantitative models and a theoretical understanding of biology will only increase. Although sometime accused by detractors of "stamp collecting"[1], it is with deep respect that we acknowledge the contributions and genius of biologists working to build a description of the living world around us. However like all other good citizens of the universe, biological systems must obey the laws of physics! Applying those laws and ideas to biology forms the basis of the field of physical biology. The challenge in quantitative biology has always been to understand the living world in its native, *in vivo*, context.

As physicists, our understanding begins with making order-of-magnitude estimates. In cellular biology the typical length scales are nanometers to microns, and the time scales vary from picosecond thermal fluctuations in a folded protein to many minutes for the time it takes an *Escherichia coli* cell to divide. Back-of-the-envelope estimates give way to abstracted theoretical models, treating, e.g., DNA as a bent beam or a protein as a polymer on a regular lattice.

If biologists have been accused of stamp collecting, their retort to physicists must be that their models, while perhaps successful in describing *in vitro* behavior, often have little to do with the full complexity and subtle interplay between *in vivo* systems. How, for example, can a pithy equation from a simplified model hope to shed light on the full richness and

[1] Ernest Rutherford once (in)famously pronounced, "In science, there is only physics. All the rest is stamp collecting."

complexity of, e.g., globular protein structure or the intricate morphology of cell organelles?

The bridge between these two worlds, theoretical biology on the one hand and *in vivo* observations on the other, lies in the world of computation and simulation. Simulations allow us to unfetter our theoretical models from the simplifications necessary to obtain tidy (and useful) analytical expressions and allow their full splendour to be unleashed and compared with experiment. In this thesis I present three case studies in applying computational techniques to biological systems to generate insights into their structure.

In **Chapter 2** we study the use of reduced alphabets as a tool for identifying proteins with structural similarity. When the sequences of two proteins are compared with one another, a reference table or matrix must be used to specify the reward or penalty for aligning any two amino acids with one another. Given that there are twenty naturally occurring amino acids, this scoring matrix must contain 210 such parameters, and when better performance is desired, even more parameters are added to the mix. Our study in bioinformatics departs from the seeming *status quo* in a field which typically dictates that the more parameters, the better. By taking a contrarian approach of coarse-graining, we show that the sensitivity of pairwise alignments between protein sequences with structural similarity is actually *increased*. See Fig. 2.15 where we observe that two reduced alphabets, SDM12 and HSDM17 with 12 and 17 letters, respectively, outperform the popular and widely-used BLOSUM62 scoring matrix in identifying structurally similar proteins with pairwise sequence alignments [1, 2]. Our study was sweeping in its scope, testing over 150 different schemes for reducing the amino acid alphabet and included statistical tests of the significance of those results using state-of-the-art bootstrapping techniques [3]. We expect that the promising improvements afforded by reduced alphbets shown in this study of pairwise alignments will also be observed in future studies with reduced alphabets and profile or HMM alignments.

In **Chapter 3** we demonstrate the measurement of applied force by analyzing the observed conformation of a phospholipid bilayer. Since the 1970s when the membrane elasticity theory of Helfrich, Canham, and Evans [4–6] was first exposited it has been known that this measurement was possible; however the combined difficulty of accurately determining and numerically analyzing the shape of the membrane made it too difficult to attempt. We demonstrate for the first time a general technique for measuring the forces on a fluid lipid bilayer based on its conformation alone and verify the forces measured in this way

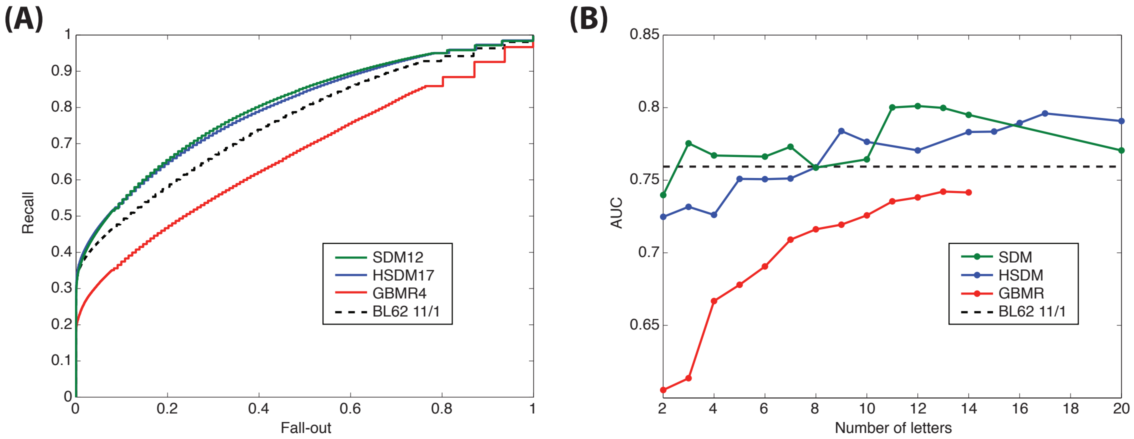


Figure 2.15: **Reduced alphabet performance in area under the Receiver Operating Characteristic curve.** (A) Receiver Operating Characteristic (ROC) curves for the top performing alphabets. The integral of this curve gives a measure of how well the entire pooled list of hits is sorted; a perfect method would have an ROC area of unity. (B) Overall sensitivity of the SDM alphabets as measured by the area under the ROC curve. The level of sensitivity of BL62 11/1 is shown with the black dashed line. Points indicate reduced alphabets that were tested; the connecting lines are a guide to the eye.

by comparison with force measurements from the optical trap used to apply force to giant unilamellar vesicles. The key result of this project is shown in Fig. 3.13 where we show that this technique accurately measures force in the subpiconewton regime. Given the explosion in fully three-dimensional membrane structures that are now becoming available through techniques like electron cryo-tomography and confocal microscopy we envision that this technique will become a valuable tool for informing models of membrane remodeling processes *in vivo*.

Finally in **Chapter 4** we study the physics of proteins in a lipid bilayer interacting via the bending of the membrane. Within the lipid bilayer of the cell are a multitude of proteins which help to broker the interactions of a cell with its outside environment. These proteins bend the membrane surrounding them and through this interaction create forces which may attract or repel other nearby proteins. This effect is observed in everyday life as attractive forces between Cheerios in a cereal bowl or bubbles in a champagne glass [7] and in this case has interesting implications for nanoscale interactions in a cell. We show that proteins interacting through bending of the surrounding lipid bilayer not only attract one another but may also have cooperative effects in gating of, e.g., membrane channels such

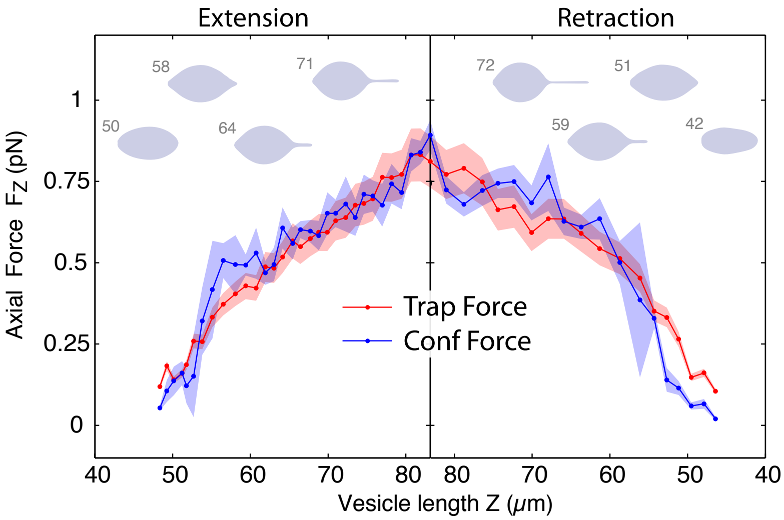


Figure 3.13: **Comparison of the applied and conformational force.** The computed conformational force (blue curve) and the applied trapping force (red curve) are in both qualitative and quantitative agreement from 0.2 to 1 pN. The red and blue shaded regions are the error in the trapping force and the r.m.s. variation of the conformational force, respectively. Note that the x-axis orientation has been inverted in the retraction data. **Inset above:** Membrane conformations as a function of axial length (μm).

as MscL, a protein channel with both open and closed states that helps prokaryotic cells cope with hypo-osmotic shock [8]. Fig. 4.5 shows how two protein channels, interacting through a membrane, increase their probability of being in the open state. Such effects are relevant at the estimated *in vivo* density of MscL channels and may have consequences for our thinking about how MscL and other such membrane proteins function in living cells.

The challenge going forward is to stand on the shoulders of such giants as Darwin and Mendel to describe biology quantitatively, ultimately allowing us to explore, understand and predict the behavior of living systems as never before possible. As a discipline, biology is one of the most demanding, requiring insights and understanding from such a wide variety of fields: chemistry, physics, and mathematics all play their part. It is my opinion that the integration and synergy of theoretical, *in silico*, *in vitro*, and *in vivo* insights from these areas will combine to make biology the source of some of the most exciting developments in science during the next century.

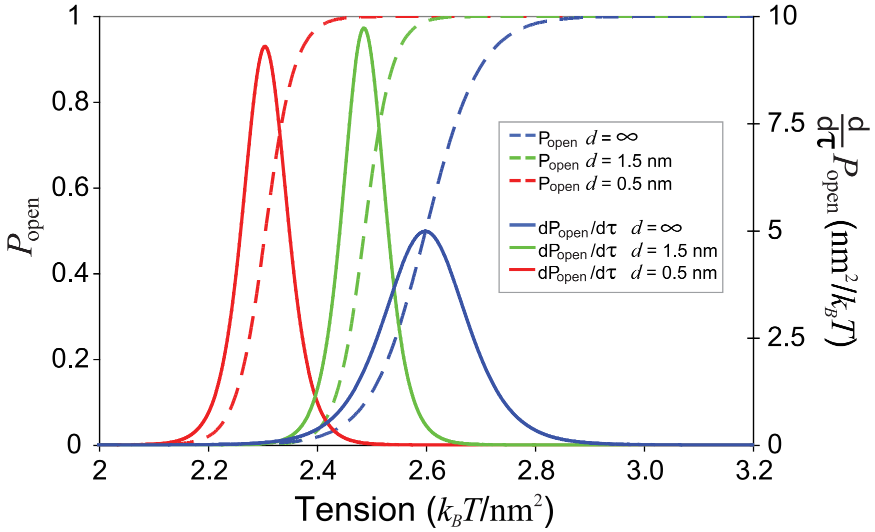


Figure 4.5: **Conformational statistics of interacting MscL proteins.** Interactions between neighboring channels lead to shifts in the probability that a channel will be in the open state (dashed lines). The sensitivity and range of response to tension, $\mathrm{d}P_{\text{open}}/\mathrm{d}\tau$, are also affected by bilayer deformations (solid lines). P_{open} and $\mathrm{d}P_{\text{open}}/\mathrm{d}\tau$ are shown for separations of 0.5 nm (red) and 1.5 nm (green) with reference to noninteracting channels at $d = \infty$ (blue). Interactions shift the critical gating tension for the closest separation by $\sim 12\%$. Additionally, the peak sensitivity is increased by $\sim 90\%$ from $\sim 5\,\mathrm{nm}^2/k_BT$ to $\sim 9.5\,\mathrm{nm}^2/k_BT$, indicating a Hill coefficient of ~ 2.

Bibliography

[1] A Prlić, F S Domingues, and M J Sippl. Structure-derived substitution matrices for alignment of distantly related sequences. *Protein Eng*, 13(8):545–550, 2000.

[2] S Henikoff and J G Henikoff. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci USA*, 89(22):10915–10919, 1992.

[3] G A Price, G E Crooks, R E Green, and S E Brenner. Statistical evaluation of pairwise protein sequence comparison with the Bayesian bootstrap. *Bioinformatics*, 21(20):3824–3831, Oct 2005.

[4] W Helfrich. Elastic properties of lipid bilayers: theory and possible experiments. *Z Naturforsch*, 28(11):693–703, 1973.

[5] P B Canham. The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell. *J Theor Biol*, 26(1):61–81, 1970.

[6] E A Evans. Bending resistance and chemically induced moments in membrane bilayers. *Biophys J*, 14:923–931, 1974.

[7] D Vella and L Mahadevan. The “Cheerios effect”. *Am J Phys*, 73(9):817–825, 2005.

[8] C D Pivetti, M R Yen, S Miller, W Busch, Y H Tseng, I R Booth, and M H Saier, Jr. Two families of mechanosensitive channel proteins. *Microbiol Mol Biol Rev*, 67:66–85, 2003.

Chapter 2

Reduced amino acid alphabets can improve the sensitivity and selectivity of pairwise sequence alignments

2.1 Introduction

Both DNA and proteins are linear polymers that can be thought of usefully in terms of their sequences: DNA as a sequence of nucleotides and proteins as a sequence of amino acids. The alphabet of DNA is well known; the (perhaps) less familiar alphabet of amino acids is shown in Table 2.1 below.

One-letter code	Three-letter code	Amino acid	One-letter code	Three-letter code	Amino acid
A	Ala	alanine	M	Met	methionine
C	Cys	cysteine	N	Asn	asparagine
D	Asp	aspartic acid	P	Pro	proline
E	Glu	glutamic acid	Q	Gln	glutamine
F	Phe	phenylalanine	R	Arg	arginine
G	Gly	glycine	S	Ser	serine
H	His	histidine	T	Thr	threonine
I	Ile	isoleucine	V	Val	valine
K	Lys	lysine	W	Trp	tryptophan
L	Leu	leucine	Y	Tyr	tyrosine

Table 2.1: **One and three letter codes for the 20 commonly occurring amino acids**

One instructive way to compare two sequences (DNA or protein) is to align them so that the conserved parts of the sequence (the parts that have stayed the same) line up with one another. Methods to do this quickly and accurately are an active area of research in bioinformatics, but most of them rely on one of two main schemes for comparing sequences:

global alignment [1] and local alignment [2]. In a global alignment we seek to align every letter in the sequences, whereas in a local alignment we only seek the best matching subsequences. This can be illustrated by a simple artificial example taken from Reference [3]. Suppose we want to align the sequences PELICAN and COELACANTH. In a global alignment we must use every letter from both sequences, but we allow ourselves to use gaps. One such global alignment is:

```
P-ELICAN--
COELACANTH
```

where a "–" character indicates a gap. With a local alignment we are only concerned with subsequences that match well; in this case the best matching subsequences are:

```
ELICAN
ELACAN
```

Local alignments are well-adapted to quickly finding the regions of greatest similarity in two sequences and will be discussed in more detail in Section 2.2.2. However both global and local alignments require a lookup table or matrix which prescribe a score (positive or negative) for lining up letters in the two sequences being compared; in the simplest case, this table will be symmetric and at least 20 by 20 in the case of proteins since there are 20 naturally occurring amino acids. The score for the entire alignment is calculated by simply adding up the score tabulated in the matrix for each pair of aligned letters. These matrices are called substitution or scoring matrices and they tell us nature's affinity for substituting one amino acid in favor of another as proteins evolve. Substitution matrices will be discussed in more detail in Section 2.2.1 but here it will suffice to say that they are formulated such that a favorable or likely substitution is assigned a positive score while one that is unlikely will have a negative score. Pairwise alignment of sequences has become a useful tool for quickly finding relationships among proteins and DNA, as manifest by the tremendous popularity of tools such as BLAST [4].

An interesting property of proteins beyond their linear sequence is that they can fold into compact, globular structures. These compact structures exhibit common themes or "folds", i.e., a common group of secondary structures with the same orientation and topology. In 1972 Anfinsen, Moore, and Stein shared the Nobel Prize in Chemistry for showing compelling evidence for the so-called “thermodynamic hypothesis” that the final three-dimensional configuration of a protein in its native environment is determined entirely by its linear sequence. Since that time it has been an area of intense research to successfully predict the folding of proteins from linear polymers to their final compact, globular native state.

As an example of theoretical efforts to understand protein folding we showcase the “HP model”, first introduced by Dill [5]. This model derives from the observation that hydrophobicity will tend to dictate a minimum free energy protein conformation with hydrophobic residues buried in the interior and the hydrophilic residues exposed at the surface of a folded protein, suggesting that these gross features are dominant in dictating the fold. The HP model has been used fruitfully with lattice folding methods to generate structures with motifs analogous to those in natural proteins [6] as well as to design *de novo* small globular proteins by patterning of polar and nonpolar residues [7].

We show below in Fig. 2.1 examples of two-dimensional and three-dimensional structures generated in an HP folding study [6] which highlight the types of insights that may be gained from HP lattice folding models. The authors of that study used a simple model for the energy of a particular lattice configuration:

$$H = \sum_{i,j} \delta_{ij} E_{\sigma_i \sigma_j}, \tag{2.1}$$

where H is the total energy of the structure, σ_i is the type (H or P) of residue i, $E_{\sigma_i \sigma_j}$ is the energy of interaction of two given residue types (one of E_{HH}, E_{HP} or E_{PP}), and δ_{ij} has a value of one if residues i and j are immediate neighbors on the lattice and do not share a bond (i.e., they do not follow one another in the primary sequence) and zero otherwise. By choosing $E_{HH} < E_{HP} < E_{PP}$ and $E_{PP} + E_{HH} < 2E_{HP}$ configurations that place hydrophobic residues on the interior and segregate H and P residue types are favored. Even with such simple rules, protein-like secondary structures, reminiscent of alpha helices and beta sheets, are observed in the lowest-energy configurations as shown in Fig. 2.1. Despite these auspicious developments, the protein folding problem remains, as yet, unsolved in its most general form and adding to the difficulty is the fact that many sequences may adopt or “design” the same protein fold. This fact was recognized in the just-mentioned theoretical study [6] and is observed widely among natural proteins as well.

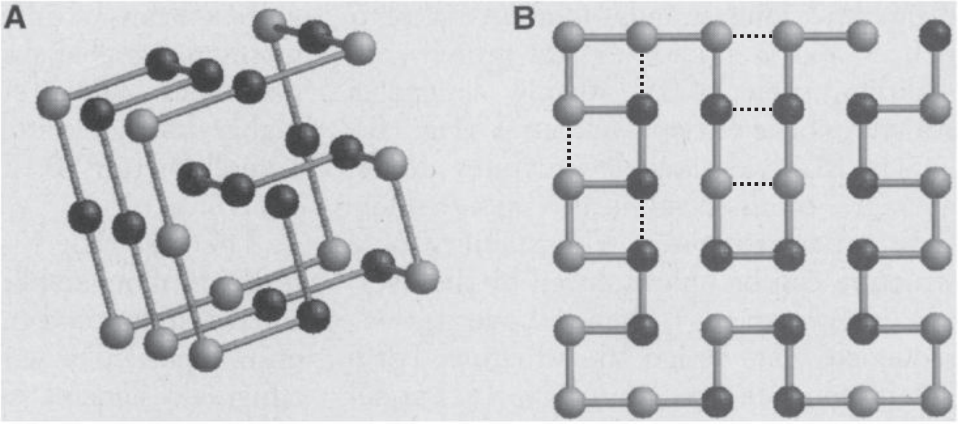


Figure 2.1: **Examples of minimum energy configurations in three dimensions (panel A) and two dimensions (panel B) generated by a simple HP folding model.** The light and dark colored balls represent polar and hydrophobic residues, respectively. In both of these structures rudimentary secondary elements akin to alpha helices and beta sheets can be discerned, arising from the favorable, non-bonded interactions of like residues. In (B) the horizontal dashed lines highlight beta-sheet-like interactions while the vertical dashed lines highlight alpha-helix-like interactions. Figure adapted from Reference [6].

An example of diverse sequences adopting a common fold is shown below[1] in Fig. 2.2. The protective protein shell or capsid surrounding the nucleic acid of three viruses (PBCV-1, adenovirus, and PRD1) have very similar architectures due to the fact that the major protein constituent of their capsids (Vp54, hexon, and P3, respectively) share a common fold. Even more fascinating is the fact that these three viruses afflict a diverse set of hosts: green algae (PBCV-1), mammals (adenovirus), and bacteria (PRD1). However this similarity was only discovered after the major capsid proteins of these viruses were crystallized and then compared with one another; since the three sequences have negligible sequence similarity to one another the common fold and probable evolutionary relationship between them could not have been discovered by pairwise sequence alignment.

This example from viruses highlights a weakness of pairwise sequence alignment: the sequence similarity shared by two proteins must exceed a certain threshold in order to be detectable above the "noise" of aligning random sequences. Protein relationships such as those shared by the three major capsid proteins of PBCV-1, adenovirus, and PRD1 belong to a so-called "twilight zone" [9] of homologous sequences having less than about

[1] Copyright ©2002 by The National Academy of Sciences of the United States of America, all rights reserved.

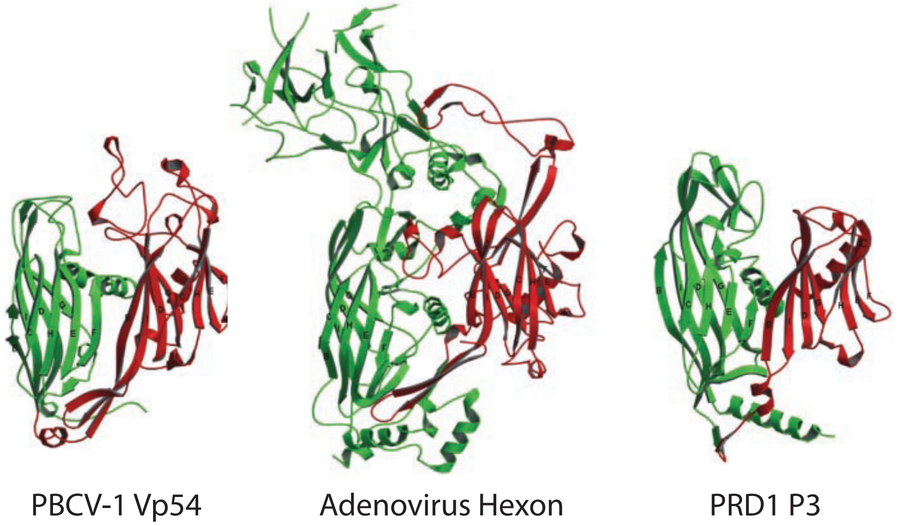


Figure 2.2: **Comparison of the major capsid proteins of three viruses: PBCV-1, adenovirus, and PRD1.** The PBCV-1, adenovirus, and PRD1 viruses share similar capsid architectures and their major capsid proteins, Vp54, hexon, and P3, respectively, share a common fold consisting of two eight-stranded β-barrels termed "jelly rolls". In the case of Vp54 and hexon there have been insertions creating the extra loops above the jelly rolls. These proteins share negligible sequence similarity and their common structure was discovered by direct comparison of crystal structures. Figure adapted, with permission, from Reference [8].

30% sequence identity, i.e., less than 30% of their residues have been conserved identically. Many interesting relationships exist in the twilight zone and given the difficulty and expense of crystallizing proteins any improvements in our ability to detect homology using sequence-based methods is of great interest.

Along these lines we highlight an encouraging example from proteins with homology to actin, a key component of the cytoskeleton in eukaryotes. Using a "property pattern" technique with then-known proteins with structural similarity to actin, Bork et al. identified five motifs common to the actin ATPase fold. Searching for these motifs yielded many hits, among them MreB, FtsA, and ParM, proteins which now after crystallization have been confirmed as having a common fold with actin. Like the viral capsid proteins, these proteins belong to the twilight zone of low mutual sequence identity. Actin-like proteins are found across all three domains of life as depicted below in Fig. 2.3. The figure shows ParM which is encoded on a transferable plasmid found in bacteria such as *E. coli*; actin, which is found in eukaryotes, and the recently crystallized protein Ta0583 from the archaeon *T. acidophilum.*

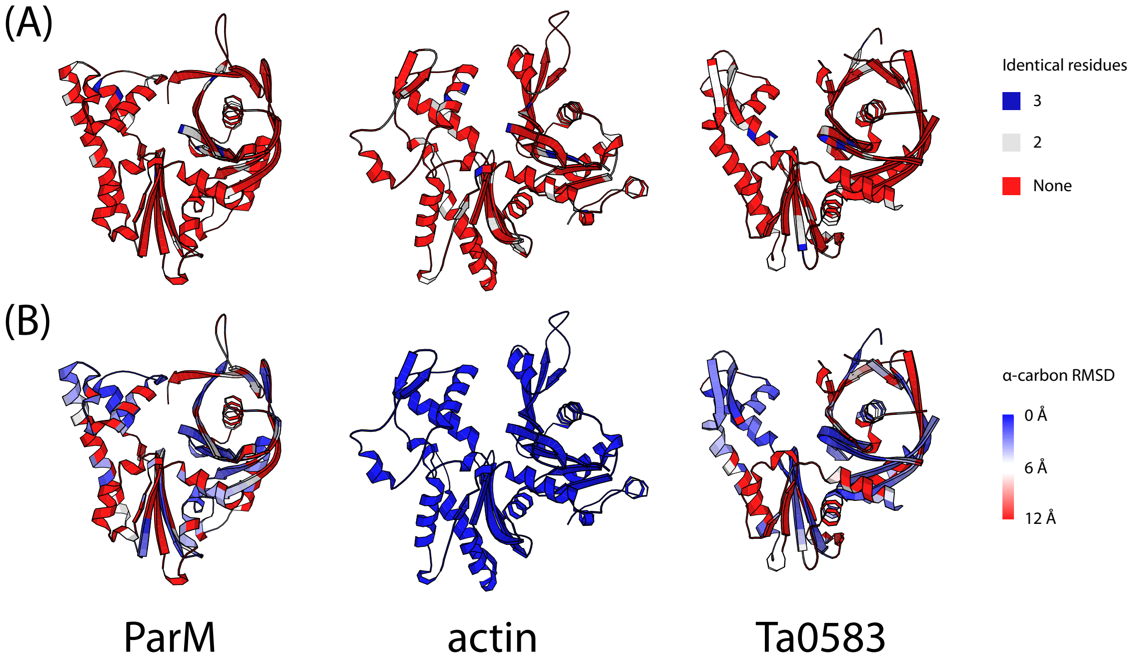


Figure 2.3: **Comparison of sequence vs. structural similarity.** Shown are eukaryotic actin from humans and two actin homologs, prokaryotic ParM from *E. coli* and archaeal Ta0583 from *T. acidophilum.* These three proteins share a common fold but have very low sequence identity, illustrated here by comparing sequence vs. structural agreement following superposition of the three structures. In both panels A and B red indicates low, white moderate, and blue high agreement. In panel A we see that sequence conservation is poor between the three proteins overall, with only a few residues conserved identically in all three (blue) or in two of three (white). In panel B we observe that the structures themselves show much higher similarity; the structures of ParM and Ta0583 are colored by the RMSD of their α-carbon backbones from actin. There are numerous examples of proteins in this “twilight zone”[10] of low sequence identity ($\lesssim 30\%$) that have a common fold. The proteins shown here were aligned using STAMP [11], included in the MultiSeq [12] extension of VMD [13] and the figure was made with MolScript v2.1.2 [14]. The PDB accession codes are 1YAG for actin, 1MWM for ParM, and 2FSJ for Ta0583.

How were Bork and coworkers able to correctly identify remote homologs of actin more than 10 years before any of those three proteins were crystallized? The key was to leverage the physicochemical properties of amino acids to construct motifs with a pattern of properties observed in the actin ATPase fold. The motifs they constructed can be represented approximately as drawing on 5 classes of amino acids: purely hydrophobic (VLIFWY), partly hydrophobic (VLIFWYMCGATKHR), tiny (GSAT), small (GSATNDVCP), and tiny plus polar (GSATNDQEKHR). The structure of actin, together with the five motifs found by Bork et al., are shown in Fig. 2.4. This study gave an early indication of the advantages that could be gained from grouping the amino acids together into classes with common properties.

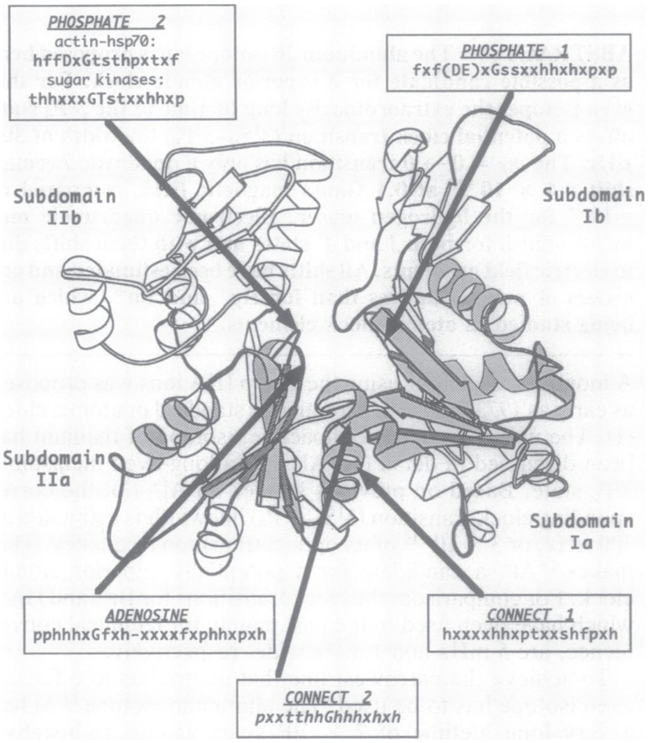


Figure 2.4: **Structure of actin together with the property pattern motifs found by Bork et al. [15].** The motifs are written in terms of the classes of amino acids allowed at that position, according to a five letter code: h, purely hydrophobic (VLIFWY); f, partly hydrophobic (VLIFWYMCGATKHR); t, tiny (GSAT); s, small (GSATNDVCP), and p, tiny plus polar (GSATNDQEKHR). An upper-case letter indicates a conserved amino acid, `-` denotes a gap, and `x` indicates any amino acid. Figure reproduced, with permission, from Reference [15].

Furthermore, previous experimental work with reduced amino acid alphabets in protein folding studies has shown that, in many cases, a reduced alphabet is sufficient to produce native-like proteins. The four-helix bundle protein Rop was studied by Munson et al., who showed that 32 amino acids in the hydrophobic core comprising eight different residues (ACEFILQT) could be replaced by patterning with just two amino acids (AL) to produce native-like proteins that showed activity *in vitro* [16], though only one mutant showed activity *in vivo* [17]. Schafmeister et al. designed *de novo* a 108 residue four-helix bundle with a seven letter alphabet (AEGKLQS) and validated their results with a crystal structure [18]. Riddle et al. were able to produce functional variants of the 57 residue Src SH3 β-sheet domain in which 38 of 40 targeted residues comprising 15 distinct amino acids were successfully mutated to a reduced alphabet of just 5 amino acids (AEGIK) [19].

Let us step back for a moment and consider the sequence and structural universe that proteins occupy. Current estimates of the number of protein folds in Nature is estimated to be between one thousand and ten thousand in total [20], an astonishingly low number compared with the huge space of possible amino acid sequences. Let us follow Frances Arnold [21] and make a simple estimate to get a feel for the numbers we are dealing with. Taking as a rule of thumb that a typical protein is composed of roughly 300 amino acids, the number of possible protein sequences is

$$N_{\text{seq}} = 20^{300} \approx 2 \times 10^{390}, \tag{2.2}$$

an unfathomably large number! However we realize that not all amino acid sequences will adopt a compact, well-ordered globular structure as is found in native proteins. In fact it was estimated by Gutin and Shaknovich that the number of sequences which adopt a protein-like nondegenerate ground state which is well removed from other low energy states decreases exponentially with chain length [22]. Still, the sequence space that proteins have to explore is large and already the number of known protein sequences is upwards of 6.5 million, the current size of the NCBI *nr* database as of this writing. It has been suggested that natural protein folds are those which have high designability, i.e., many sequences will fold to that particular structure [23]. As we have seen in two examples from viruses and actin-like proteins, such sequences may share little to no similarity with one another in the conventional 20 letter alphabet. This large degeneracy of sequences designing the same fold invites us to look for a coarse-grained sequence description that will reveal the underlying structural similarities between these apparently dissimilar sequences.

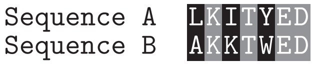
Given the success of reduced alphabet models in reproducing important features of protein structure and folding together with the experimental success in designing native-like proteins from reduced alphabets, we surmise that these simple folding ideas might also be reflected in pairwise alignments of the sequences of structurally similar proteins. We hypothesize that by properly grouping the 20 naturally occurring amino acids into classes and thereby coarse-graining the scoring matrices, similarities in protein sequence that are not readily seen in the full 20 letter alphabet would be revealed. By all of the measures we used, reduced alphabets showed increased effectiveness at identifying structurally similar proteins as defined by the DALI database by a modest though statistically significant amount. Based on these gains in pairwise alignments and other past successes in the literature, we believe that the reduced alphabet approach applied to more sensitive methods, e.g., PSI-BLAST profile searches, holds promise for detecting structurally related proteins with weak sequence similarity.

The remainder of this chapter is organized as follows. In Section 2.2 we describe our procedure for coarse-graining substitution matrices, outline the reduced alphabet schemes tested, and reference databases used in this study, as well as describe the principal metrics for this work: area under the Receiver Operating Characteristic curve, mean pooled precision, and recall at 0.01 errors per query. In Section 2.3 we present the results of all vs. all sequence alignments using each of the reduced alphabets, showing the performance of reduced alphabets in comparison with various common full 20 letter substitution matrices. The results of a study comparing structural alignments with sequence alignments are also shown for the full and reduced alphabets. Finally in Section 2.4 we compare the results of this study with other similar work and speculate on promising avenues for further development with the reduced alphabet concept.

2.2 Methods

2.2.1 Substitution matrices

Suppose that we have a trial alignment of two sequences A and B, and that we would like to evaluate the quality of the alignment. During the course of evolution, as sequences change with time, we may (correctly) suppose that there are preferred mutations for each of the 20 amino acids, and we would like to reward alignments that tend to reflect such preferred substitutions over random mutations. Consider the hypothetical alignment shown below in Fig. 2.5.



```
Sequence A  LKITYED
Sequence B  AKKTWED
```

Figure 2.5: **Hypothetical sequence alignment.** Matching residues have a gray background; non-identical pairs are shown with a black background.

What is the likelihood that sequences A and B are related by evolution? One way to begin to answer this question would be to compare the probability of observing the aligned residue pairs according to the preferred substitution patterns observed in actual evolution vs. the likelihood of those pairings due to drawing sequences randomly according to some background frequency of amino acids. We may write this (relative) probability thusly:

$$p_{\mathrm{rel}}(\Xi^{AB}) = \frac{p_{\mathrm{E}}(AL)\, p_{\mathrm{E}}(KK)\, p_{\mathrm{E}}(IK)\, p_{\mathrm{E}}(TT)\, p_{\mathrm{E}}(WY)\, p_{\mathrm{E}}(EE)\, p_{\mathrm{E}}(DD)}{p_{\mathrm{R}}(AL)\, p_{\mathrm{R}}(KK)\, p_{\mathrm{R}}(IK)\, p_{\mathrm{R}}(TT)\, p_{\mathrm{R}}(WY)\, p_{\mathrm{R}}(EE)\, p_{\mathrm{R}}(DD)} \tag{2.3}$$

$$= \prod_i \frac{p_{\mathrm{E}}(\Xi_i^{AB})}{p_{\mathrm{R}}(\Xi_i^{AB})}, \tag{2.4}$$

where Ξ^{AB} is the particular alignment of sequences A and B, and $p_{\mathrm{rel}}(\Xi)$ is the relative probability of an alignment Ξ due to evolution vs. random chance. The quantities $p_{\mathrm{E}}(X_1X_2)$ and $p_{\mathrm{R}}(X_1X_2)$ are the probabilities of observing a pairing of amino acids X_1 and X_2 in sequences related by evolution and by random chance, respectively. The symbol Ξ_i^{AB} denotes the pair of amino acids observed at position i in the alignment Ξ^{AB}.

Working with this large fraction is rather unwieldy, however. Let us take the logarithm of both sides of Eq. 2.4; we will define the logarithm of the relative probabilities $p_{\mathrm{E}}/p_{\mathrm{R}}$ to be scores for various pairs of amino acids and the sum of the scores for each of the amino acid pairings will be the overall score for the alignment. Since the logarithm is a monotonic function of its argument, this score will still provide a consistent means of comparing alignments:

$$\begin{aligned} S(\Xi_{AB}) &= \log\left[\prod_i \frac{p_{\mathrm{E}}(\Xi_i^{AB})}{p_{\mathrm{R}}(\Xi_i^{AB})}\right] && (2.5) \\ &= \sum_i \log\left[\frac{p_{\mathrm{E}}(\Xi_i^{AB})}{p_{\mathrm{R}}(\Xi_i^{AB})}\right] && (2.6) \\ &= \sum_i s(\Xi_i^{AB}). && (2.7) \end{aligned}$$

The quantity s is the score associated with a pair of aligned amino acids and is calculated by taking the logarithm of the ratio of the observed pairing probability to the expected pairing probability (often called a log-odds ratio). We note that s can be represented as a 20 by 20 symmetric matrix which can be calculated if the probabilities p_{E} and p_{R} are known. One way to determine these probabilities is from a reference set of aligned proteins that have been curated in some way, e.g., by hand or by structural alignment. With the reference alignments in hand, one may determine the background frequencies of the amino acids as well as the observed pairings of various amino acid pairs through time in sequences believed to be related by evolution. This is the basis of the scheme proposed by Henikoff and Henikoff [24] in their landmark work introducing the BLOSUM series of scoring matrices (also called substitution matrices).

We will briefly describe here the BLOSUM method for deriving substitution matrices from a reference set of alignments. Let us label the naturally occurring amino acids with indices 1–20; we may then derive a matrix c_{ij} with each entry of the matrix being the tally of the observed pairings of amino acid i with amino acid j in the reference alignments. Pairwise alignments do not distinguish between aligning, e.g., AD and DA, so if the total count of ij and ji pairs is C (with $i \neq j$) we assign $c_{ij} = c_{ji} = C/2$ to reflect this symmetry. The underlying reason for this symmetry is that we assume no *a priori* knowledge of the order in which the sequences arose; without such knowledge, the likelihood of a substitution from, e.g., A to G and G to A, are equal. The observed probability matrix o_{ij} is the normalized c_{ij} matrix:

$$o_{ij} = \frac{c_{ij}}{\sum_{i=1}^{20}\sum_{j=1}^{20} c_{ij}}. \qquad (2.8)$$

The background frequencies of each of the amino acids p_i is now easily calculated from o_{ij}:

$$p_i = \sum_{j=1}^{20} o_{ij}, \tag{2.9}$$

and the expected (random) probability of aligning amino acid i with j is:

$$e_{ij} = p_i \, p_j. \tag{2.10}$$

The substitution matrix score m_{ij} is calculated from the observed and expected probabilities as a log-odds ratio:

$$m_{ij} = m_{ji} = \log_b \left(\frac{o_{ij}}{e_{ij}} \right), \tag{2.11}$$

where the base of the logarithm is usually chosen as $b = 2$ so that m_{ij} has units of bits of information (though substitution matrix values are sometimes measured in half-bits or other fractional bit units). For convenience of notation, the BLOSUM series of matrices will be referred to hereafter as BL followed by the level of sequence identity used in building the matrix, e.g., BL62 for the BLOSUM matrix with 62% identity cutoff.

We follow this log-odds method in formulating matrices based on a reduced alphabet. Each reduced alphabet scheme clusters amino acids together into groups where all amino acids within a group are considered identical. Given N groups of amino acids defining a reduced alphabet, the new frequency of group I is calculated as:

$$p_I = \sum_{k \in I} p_k, \tag{2.12}$$

where k runs over each amino acid in group I. The new expected and observed probabilities to align group I with group J are:

$$\begin{aligned} e_{IJ} &= p_I \, p_J, & (2.13) \\ o_{IJ} &= \sum_{i \in I} \sum_{j \in J} o_{ij}. & (2.14) \end{aligned}$$

Finally, the new matrix entries in the reduced $N \times N$ matrix are:

$$\begin{aligned} M_{IJ} &= \log_b \left(\frac{o_{IJ}}{e_{IJ}} \right) && (2.15) \\ &= \log_b \left[\frac{\sum_{i \in I} \sum_{j \in J} o_{ij}}{\sum_{i \in I} p_i \sum_{j \in J} p_j} \right]. && (2.16) \end{aligned}$$

This method differs from that used in some previous reduced alphabet studies [25, 26] which used the arithmetic mean of the substitution matrix entries; using the mean of the substitution matrix scores is inconsistent with the log-odds probability scheme upon which the substitution matrices are based.

Example of deriving a log-odds substitution matrix Let us illustrate the process of deriving a log-odds substitution matrix with a simple example. Shown below in Fig. 2.6 is a multiple alignment of human, cow, and carp alpha-chain hemoglobin sequences from which we will derive a simple substitution matrix. Here we have defined the hydrophobic residues to be VILMFWYAC (black background) and the polar residues to be EDRKGPSTHQN (gray background).

```
Human   MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DL    49
Cow     MVLSAADKGNVKAAWGKVGGHAAEYGAEALERMFLSFPTTKTYFPHF-DL    49
Carp    MSLSDKDKAAVKGLWAKISPKADDIGAEALGRMLTVYPQTKTYFAHWADL    50

Human   SHGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNF    99
Cow     SHGSAQVKGHGAKVAAALTKAVEHLDDLPGALSELSDLHAHKLRVDPVNF    99
Carp    SPGSGPVKKHGKVIMGAVGDAVSKIDDLVGGLAALSELHAFKLRVDPANF   100

Human   KLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR   142
Cow     KLLSHSLLVTLASHLPSDFTPAVHASLDKFLANVSTVLTSKYR   142
Carp    KILAHNVIVVIGMLYPGDFPPEVHMSVDKFFQNLALALSEKYR   143
```

Figure 2.6: **Multiple alignment of alpha-chain hemoglobin sequences from human, cattle, and carp.** Hydrophobic residues are shown with a black background and polar with a gray background. The NCBI Entrez Protein accession numbers used were 57013850 (human), 13634094 (cow), and 122392 (carp).

We first tabulate pairs at each position in the alignment to create the c_{ij} matrix. For instance, in the second position of the alignment we have H paired with H (human-cow), H with P (human-carp) and H with P (cow-carp). Repeating this process for each ungapped position yields:

$$c_{ij} = \begin{pmatrix} c_{\mathrm{HH}} & c_{\mathrm{HP}} \\ c_{\mathrm{PH}} & c_{\mathrm{PP}} \end{pmatrix} = \begin{pmatrix} 167 & 33 \\ 33 & 193 \end{pmatrix}.$$

Note that 66 pairings of H with P were observed, and these were split equally between c_{HP} and c_{PH}. Normalizing by the total number of pairs gives the observed probability of each kind of pair:

$$o_{ij} = \begin{pmatrix} 0.39 & 0.077 \\ 0.077 & 0.45 \end{pmatrix}.$$

The background frequencies of H and P are:

$$\begin{aligned} p_{\mathrm{H}} &= o_{\mathrm{HH}} + \frac{1}{2}o_{\mathrm{HP}} + \frac{1}{2}o_{\mathrm{PH}} = 0.39 + \frac{1}{2}0.077 + \frac{1}{2}0.077 = .47 \\ p_{\mathrm{P}} &= o_{\mathrm{PP}} + \frac{1}{2}o_{\mathrm{PH}} + \frac{1}{2}o_{\mathrm{HP}} = 0.45 + \frac{1}{2}0.077 + \frac{1}{2}0.077 = .53, \end{aligned}$$

which can now be used to form the expected probability matrix e_{ij}:

$$e_{ij} = \begin{pmatrix} p_{\mathrm{H}}\,p_{\mathrm{H}} & p_{\mathrm{H}}\,p_{\mathrm{P}} \\ p_{\mathrm{P}}\,p_{\mathrm{H}} & p_{\mathrm{P}}\,p_{\mathrm{P}} \end{pmatrix} = \begin{pmatrix} 0.22 & 0.25 \\ 0.25 & 0.28 \end{pmatrix}.$$

Already we can see a large discrepancy between the expected probabilities of pairing vs. the observed probabilities. The final log-odds HP scoring matrix in half-bit units is calculated to be:

$$s_{ij} = 2\log_2\left(\frac{o_{ij}}{e_{ij}}\right) = \begin{pmatrix} 1.65 & -3.37 \\ -3.37 & 1.39 \end{pmatrix}.$$

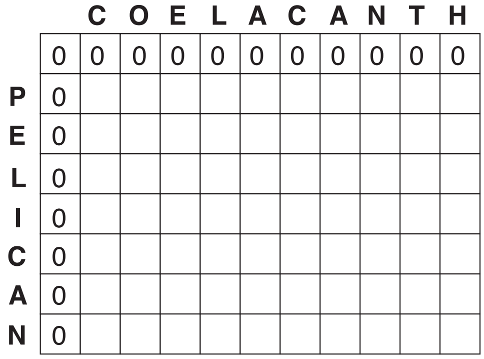
The large negative score for HP and PH pairs reflects the high conservation of the hydrophobic/polar pattern in the alpha-chain hemoglobin sequences: mutations from P to H and vice versa are highly suppressed. The BLOSUM matrices were derived in this same way for the full amino acid alphabet using blocks of ungapped multiple alignments. The Henikoff and Henikoff log-odds scheme was followed later by other authors who derived substitution matrices used in this study [27, 28].

2.2.2 Sequence alignment

We introduce the Smith-Waterman algorithm for local alignments here briefly with an example adapted from reference [3]. The algorithm proceeds in three stages:

1. Initialization
2. Fill
3. Traceback.

We will use the sequences COELACANTH and PELICAN and a simple scoring scheme: +1 for matching letters, -1 for mismatched letters and gaps. We write our sequences on the axes of a matrix and then initialize the matrix by filling in zeros on the first row and column as shown in Fig. 2.7. Now the alignment matrix is filled by starting at the upper-left and working right and down until we reach the first matching letters from both sequences which in this case is E, as shown in Fig. 2.8. Up until that point every cell has received a zero score because the letters in the two sequences do not match there and in a local alignment we do not consider negative scores. Proceeding with the fill, we reach the next two letters that align from each sequence, L, shown in Fig. 2.9. In order to get a score of 2, we proceeded down and right from the initial aligned E from each sequence and we make a note of where we came from with an arrow as shown in the upper-left hand corner of the cell. These arrows will help us traceback the best alignment in the final step of the algorithm. The next cell in the matrix is our first gap, shown in Fig. 2.10. By not adding a letter from PELICAN and proceeding from the L to the A of COELACANTH we add a



		C	O	E	L	A	C	A	N	T	H
	0	0	0	0	0	0	0	0	0	0	0
P	0										
E	0										
L	0										
I	0										
C	0										
A	0										
N	0										

Figure 2.7: **Initialization of the alignment matrix**

		C	O	E	L	A	C	A	N	T	H
	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	1							
L											
I											
C											
A											
N											

Figure 2.8: **Beginning to fill the alignment matrix**

gap which incurs a penalty of -1 and reduces the score for that cell to 1. We note where we came from by putting an arrow to the left in the corner of the cell. In general when we visit a cell in the matrix during the fill, we have three options:

1. Align letters from both sequences, incurring either a benefit or cost according to the scores in the substitution matrix.

2. Add a gap in the first sequence along the top of the matrix.

3. Add a gap in the second sequence along the side of the matrix.

Among those options we choose the one that will generate the highest score for the cell. If there is a tie we may take any of the highest scoring options. If the best score a cell can receive is negative, then we simply fill in a zero. After choosing one of these three options, we then note the choice we made for the traceback by putting either a diagonal, upwards

		C	O	E	L	A	C	A	N	T	H
	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	1	0	0	0	0	0	0	0
L	0	0	0	0	↖2						
I											
C											
A											
N											

Figure 2.9: **The next non-zero score in the matrix**

		C	O	E	L	A	C	A	N	T	H
	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	1	0	0	0	0	0	0	0
L	0	0	0	0	↖2	←1					
I											
C											
A											
N											

Figure 2.10: **A gap is added to the alignment**

or leftwards pointing arrow, respectively, to indicate which direction the best scoring path follows. We do not fill in arrows inside of or pointing to cells with a zero score since those do not contribute usefully to the alignment. The filled matrix is shown in Fig. 2.11. Finally we traceback the best alignment from our matrix. This is done by simply starting at the cell in the matrix with the highest score and tracing back through the cells using the arrows we noted during the fill phase. In this case, we start at the cell with the alignment of N from each sequence which had a score of 4 and follow the arrows to construct our final, globally optimized alignment with our chosen scoring scheme:

		C	O	E	L	A	C	A	N	T	H
	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	1	0	0	0	0	0	0	0
L	0	0	0	0	↖2	←1	0	0	0	0	0
I	0	0	0	0	↑1	↖1	0	0	0	0	0
C	0	0	0	0	0	0	↖2	0	0	0	0
A	0	0	0	0	0	1	0	↖3	←2	←1	0
N	0	0	0	0	0	0	0	↑2	↖4	←3	←2

Figure 2.11: **The final alignment matrix**

```
ELICAN
ELACAN
```

2.2.3 Statistics of optimal local alignments

In the previous section we saw that in an optimal local alignment as executed by the Smith-Waterman algorithm we always choose the best alignment from among the population of other suboptimal alignments calculated in the alignment matrix. That is, we generate a population of scores in the matrix and always choose from that population the highest score. The statistics of extremes describes the distribution of best or highest scores one would expect by alignment of sequences drawn from a random background of amino acids. These statistics were worked out by Gumbel [29] and the distribution associated with highest scores is often termed a Gumbel distribution; the connection to optimal, ungapped local alignments was first worked out by Karlin and Altschul in their seminal work on the statistics of sequence alignment [30]. The statistical significance of each hit is estimated with the so-called E-value, which is given by the Karlin-Altschul equation:

$$E = Kmn \exp\left(-\lambda S\right), \tag{2.17}$$

where S is the score of the alignment and mn is the product of the number of letters in the database and query sequence. The remaining variables, K and λ, are parameters of the distribution and are typically estimated from the distribution of scores produced in a database search. Strictly speaking the Karlin-Altschul equation only holds for ungapped alignments, but in practice gapped local alignments also follow extreme value statistics as long as the gap penalties are not too small. The probability of obtaining an alignment with a score S' at least as high as S is simply:

$$P(S' > S) = 1 - \mathrm{e}^{-E}, \tag{2.18}$$

and by analogy to Poisson statistics the E-value is roughly the number of alignments with score S or better that one would expect to obtain by random chance. For this reason the E-value is also often called the “Expect” value and this value serves as the basis for ranking alignments in this study.

2.2.4 Reduced alphabet schemes

A reduced alphabet is any clustering of amino acids based on some measure of their relative similarity. Many such schemes have been proposed; the ones used in this study are briefly reviewed here together with the abbreviations used to refer to them. If a name for the scheme is given by the authors (e.g., SDM and DSSP) it has also been used here, otherwise abbreviations are formed by using the first letters of the names of the first and last authors. Thomas and Dill [31] created a hierarchy of amino acid groupings based on intuitive physicochemical considerations (TD). Mirny and Shakhnovich [32] constructed a six letter alphabet based mostly upon intuition as well as a study of the effects of disulfide bonds on protein folding which suggested separating aliphatic hydrophobic and aromatic hydrophobic residues [33] (MS). Solis and Rackovsky [34] posited clusters based on maximum preservation of structural information (DSSP and GBMR). Andersen and Brunak [35] searched for clusters of amino acids based on the ability of standard methods to correctly predict secondary structure from the simplified sequences (AB). Cieplak et al. [36] used the Miyazawa-Jernigan interaction matrix [37] together with a distance-based clustering scheme to partition the naturally occurring amino acids into 2- and 5-letter groups (CB). Prlić et al. [28] derived new substitution matrices based on structural alignments of proteins with low sequence identity and then clustered the amino acids based on those matrices (SDM and HSDM). On the basis of a comparison of early substitution matrices, Landès and Risler [38] proposed a 10-letter alphabet that showed promise for increasing the sensitivity of protein alignment searches (LR). Li et al. [26] proposed grouping schemes based on preservation of information in global sequence alignments between a sequence and its reduced-alphabet version. They produced two groupings, one allowing amino acids to change their order or "interlace" (LW-I) and one where they were not allowed to change order (LW-NI). The LW schemes were identical at the levels of 2, 3, and 15 through 19 letters. We also note that the CB and LW schemes were identical at the 2 letter level. Melo and Marti-Renom [39] created a 5 letter clustering of amino acids based on the Johnson-Overington matrix (JO20) [27], which they found performed well in aligning homologous sequences and fold assessment (MM). Murphy, Wallqvist, and Levy [25], inspired by experimental successes in designing proteins with reduced alphabets, proposed clusters of amino acids based on the BL50 substitution matrix (ML). Liu et al. [40] studied the pair frequency counts in the

2	LIVFMYWC								DNTSKEQRAGPH											
3	LIVFMYW							C	DNTSKEQRAGPH											
4	LIVFMYW							C	DNTSKEQRAGP											H
5	LIVFMY						W	C	DNTSKEQRAGP											H
6	LIVFMY						W	C	DNTSKEQRAG										P	H
7	LIVFMY						W	C	DNTSKEQRA									G	P	H
8	LIVFM					Y	W	C	DNTSKEQRA									G	P	H
9	LIVFM					Y	W	C	DNTSKEQR								A	G	P	H
10	LIVFM					Y	W	C	DN		TSKEQR						A	G	P	H
12	LIVF				M	Y	W	C	DN		TSKEQ					R	A	G	P	H
14	LIV			F	M	Y	W	C	DN		TS		KEQ			R	A	G	P	H
15	LIV			F	M	Y	W	C	D	N	TS		KEQ			R	A	G	P	H
16	LIV			F	M	Y	W	C	D	N	TS		KE		Q	R	A	G	P	H
17	LIV			F	M	Y	W	C	D	N	T	S	KE		Q	R	A	G	P	H
20	L	I	V	F	M	Y	W	C	D	N	T	S	K	E	Q	R	A	G	P	H

Table 2.2: **The hierarchy of amino acid classes in the HSDM reduced alphabet scheme [28].** The first column gives the number of groups or "letters" in the reduced alphabet; the amino acid groupings at each level are shown on the right.

Miyazawa-Jernigan and BL50 matrices to find deviations from a random background and, based thereon, proposed a clustering of amino acids (LZ-MJ and LZ-BL). Finally, Wang and Wang [41] derived clusters from the Miyazawa-Jernigan matrix by preserving maximal similarity between a reduced-alphabet version of the matrix and the full 20 by 20 matrix (WW). They found a five letter alphabet (IKEAG) that matched with what Baker et al. [19] had found in their experimental study producing SH3 domains from reduced alphabets. Each of these schemes produced a hierarchy of amino acid classes, as typified by the dendrogram of the HSDM scheme [28] in Table 2.2. At each level in the hierarchy the number of classes or "letters" in the alphabet is increased. We tested each of the reduced alphabet schemes in the papers just cited; Table 2.3 shows the abbreviations for each scheme, the various levels of clusterings comprising the scheme and the frequency matrix and gap penalties used. In this work, reduced alphabet matrices will be referred to by the alphabet scheme (TD, SDM, HSDM, etc.) followed by the number of letters in the alphabet, e.g., HSDM17.

2.2.5 Protein database

Proteins can be thought of as being roughly made up of independently folding, compact globular units or domains. We chose the DALI (Distance mAtrix aLIgnment) database [42], which uses fully automated methods to cluster protein domains based on their structural similarity as our "gold standard" for determining the structural relatedness of proteins.

	Alphabet	**Substitution**	**Gap penalties**	
Scheme	**sizes**	**matrix**	*Open*	*Extend*
AB* [35]	2-19	BL62	11	1
CB* [36]	2,5	BL62	11	1
DSSP* [34]	2-14	BL62	11	1
GBMR* [34]	2-14	BL62	11	1
HSDM [28]	2-10,12,14-17	HSDM	19	1
LR* [38]	10	BL62	11	1
LW-I* / LW-NI* [26]	2-19	BL62	11	1
LZ-MJ* / LZ-BL [40]	2-16	BL50	11	1
MM [39]	5	JO20	140	0
ML [25]	2,4,8,10,15	BL50	12	2
MS [32]	6	BL62	11	1
SDM [28]	2-4,6-8,10-14	SDM	7	1
TD* [36]	2-10,14	BL62	11	1
WW* [41]	5	BL62	11	1

Table 2.3: **Reduced alphabet schemes investigated in this thesis.** Abbreviations and references are listed in the first column; alphabet sizes, matrix used, and gap penalties are also shown. Wherever possible we used the matrices and gaps given in the original articles referenced, though we note that the starred schemes were proposed independently of any particular substitution matrix. In those cases, the BL62 frequency counts were used to derive the coarse-grained matrices with 11/1 gaps. In addition to the reduced alphabet schemes tested above we also tested the following full 20 by 20 matrices: BL50 11/1, BL50 12/2, BL62 7/1, BL62 11/1, JO20 140/0, SDM 7/1 and HSDM 19/1.

DALI partitions each protein structure in the PDB into domains by maximizing criteria of compactness and recurrence of those domains [43]. After determining the domains, all vs. all structural alignments of the domains are executed and a z-score[2] estimated to indicate the statistical significance of those alignments [43]. Finally, the domains are clustered into families based on z-score cutoffs; a cut-off z-score greater than 2, indicating statistically significant structural similarity at the 2σ level, is used to define proteins with roughly the same "fold" [43].

As an example of how DALI domains are identified from a protein structure we show the domains determined by the DALI decomposition procedure from diphtheria toxin (1DDT) in Fig. 2.12, taken from reference [43]. In diphtheria toxin, three domains are present which each have a vital role in the infection process: receptor binding, membrane insertion, and catalysis of ADP-ribosylation of elongation factor 2, which inhibits the synthesis of new proteins by the host ribosome. The DALI protein decomposition procedure illustrated here was used on all proteins in the PDB to identify recurring domains or folds.

The sequence library for this study was drawn from the DALI *pdb90* database using each

[2] A z-score or standard score is defined to be the difference of an observation x from the population mean μ divided by the standard deviation σ: $z = (x - \mu)/\sigma$.

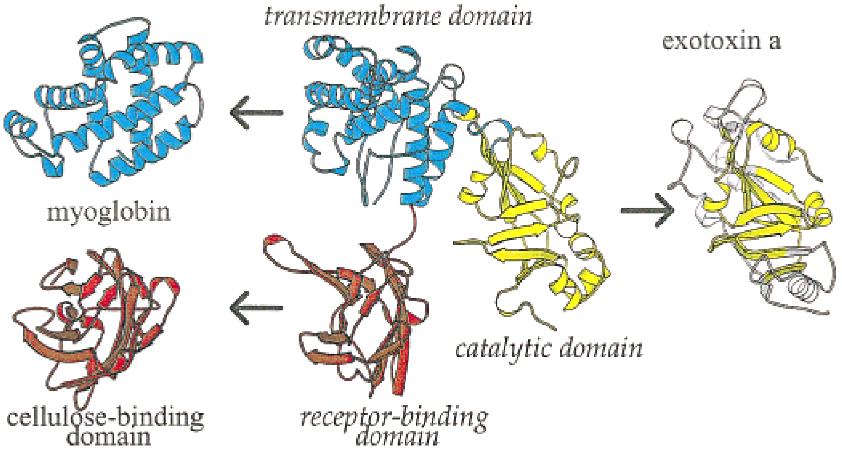


Figure 2.12: **The domains of diphtheria toxin.** Diphtheria toxin (1DDT) breaks down neatly into three compact, globular domains as identified here by the DALI automatic protein decomposition algorithm. Each of the domains present in diphtheria represents a recurring structural motif or fold found in other proteins, examples of which are shown adjacent to each colored domain: myoglobin (blue), cellulose-binding domain (red), and exotoxin A (yellow). Figure from reference [43].

of the representative sequences in the domain fold classes defined by the DALI Domain Dictionary [43, 44], both available for download at the DALI website [45]. All pairs of sequences within the same domain fold class are considered to be structurally related "hits" (true positives) in our database searches. In total 13,351 sequences were drawn from the *pdb90* database, representing 2780 fold classes. One domain fold class, number 1636, was not represented in the database because its representative sequence, 1mwxA_1, was not found in the latest version of *pdb90* available for download. We also note that there were 1264 sequences which were singletons, i.e., they were the only members of their DALI fold class; these sequences are in some sense undetectable since they have no relationship to other proteins in the database.

2.2.6 Alignment program

All vs. all Smith-Waterman alignments were executed using SSEARCH version 3.4 from the FASTA sequence alignment suite [2, 46]; the alignments were ranked by E-value as calculated by the default SSEARCH statistics option (specified by the "-z 1" switch on the command line).

2.2.7 Generation of search results

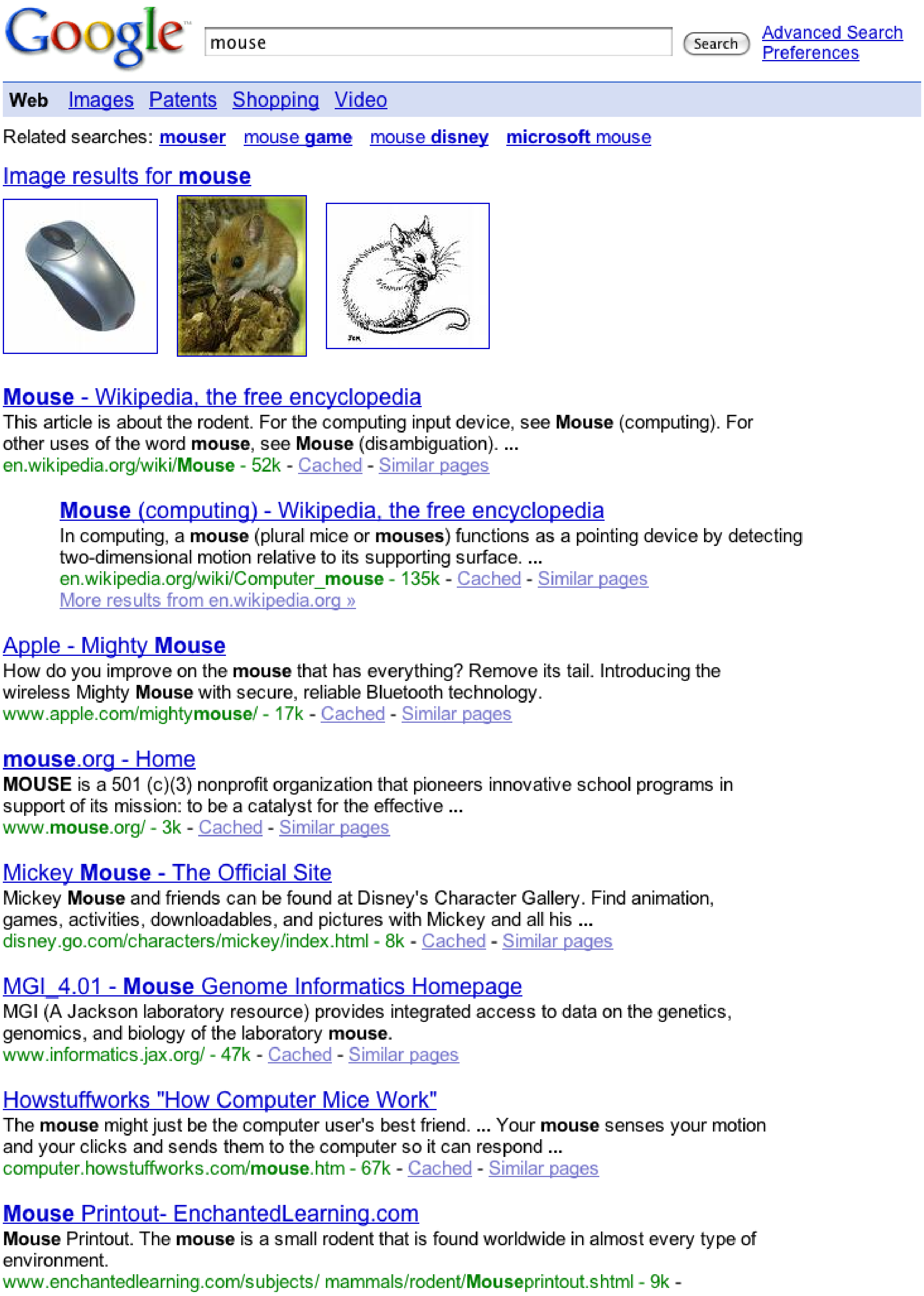
We executed all vs. all alignments, using each sequence in the DALI *pdb90* database in turn as a query against the remaining sequences. The results of all these searches were then pooled into a single list of results ranked by the E-value assigned by SSEARCH; when true and false positives shared an E-value, the false positive alignments were ranked ahead of the true positives to obtain a conservative estimate of discriminating power. After pooling the results of querying the database with each sequence, we choose a particular E-value and consider all results at this E-value or lower to be "hits". The recall, fall-out, precision, and errors per query (EPQ) are calculated from the list of hits as follows:

$$\begin{aligned}
\text{recall} &= \frac{\text{tp}}{\text{N}_{\text{tp}}}, && (2.19)\\
\text{fall-out} &= \frac{\text{fp}}{\text{N}_{\text{fp}}}, && (2.20)\\
\text{precision} &= \frac{\text{tp}}{\text{tp}+\text{fp}}, && (2.21)\\
\text{EPQ} &= \frac{\text{fp}}{\text{N}_{\text{seq}}}, && (2.22)
\end{aligned}$$

where tp is the number of true positive hits, fp is the number of false positive hits and N_{seq} is the total number of sequences; N_{tp} and N_{fp} are the total number of true and false positive relationships in the database, respectively. Moving down the pooled list of search results we generate successively larger groups of hits at increasing E-values with associated values for recall, fall-out, precision, and EPQ.

Some of the basic principles of information retrieval can be illustrated by an example from a Google search, in this case on the term "mouse". Suppose now that the actual subject of interest is the rodent; as one looks at the results returned in Fig. 2.13 one notes several different kinds of mice, only some of which relate to the biological organism. Google has used its own proprietary technology to try and rank the results of the database search based on their model for what will be of interest to a user of their search engine. In Table 2.4 we tally the various quantities just described above.

Now let us apply these metrics to an example of the data that is obtained in the SSEARCH database query, as shown in Table 2.5 which lists the first 20 hits from querying the DALI pdb90 sequence database with the first domain of 1BIG (1big_1), a toxin



Related searches: **mouser** mouse **game** mouse **disney** **microsoft** mouse

Image results for **mouse**

Mouse - Wikipedia, the free encyclopedia
This article is about the rodent. For the computing input device, see **Mouse** (computing). For other uses of the word **mouse**, see **Mouse** (disambiguation). ...
en.wikipedia.org/wiki/**Mouse** - 52k - Cached - Similar pages

Mouse (computing) - Wikipedia, the free encyclopedia
In computing, a **mouse** (plural mice or **mouses**) functions as a pointing device by detecting two-dimensional motion relative to its supporting surface. ...
en.wikipedia.org/wiki/Computer_**mouse** - 135k - Cached - Similar pages
More results from en.wikipedia.org »

Apple - Mighty **Mouse**
How do you improve on the **mouse** that has everything? Remove its tail. Introducing the wireless Mighty **Mouse** with secure, reliable Bluetooth technology.
www.apple.com/mighty**mouse**/ - 17k - Cached - Similar pages

mouse.org - Home
MOUSE is a 501 (c)(3) nonprofit organization that pioneers innovative school programs in support of its mission: to be a catalyst for the effective ...
www.**mouse**.org/ - 3k - Cached - Similar pages

Mickey **Mouse** - The Official Site
Mickey **Mouse** and friends can be found at Disney's Character Gallery. Find animation, games, activities, downloadables, and pictures with Mickey and all his ...
disney.go.com/characters/mickey/index.html - 8k - Cached - Similar pages

MGI_4.01 - **Mouse** Genome Informatics Homepage
MGI (A Jackson laboratory resource) provides integrated access to data on the genetics, genomics, and biology of the laboratory **mouse**.
www.informatics.jax.org/ - 47k - Cached - Similar pages

Howstuffworks "How Computer Mice Work"
The **mouse** might just be the computer user's best friend. ... Your **mouse** senses your motion and your clicks and sends them to the computer so it can respond ...
computer.howstuffworks.com/**mouse**.htm - 67k - Cached - Similar pages

Mouse Printout- EnchantedLearning.com
Mouse Printout. The **mouse** is a small rodent that is found worldwide in almost every type of environment.
www.enchantedlearning.com/subjects/ mammals/rodent/**Mouse**printout.shtml - 9k -

Figure 2.13: **Google results for a search on "mouse"**

tp	fp	Recall	Fall-out	Precision
0	1	0	0.17	0
1	1	0.2	0.17	0.5
2	1	0.4	0.17	0.67
3	1	0.6	0.17	0.75
3	2	0.6	0.33	0.6
3	3	0.6	0.5	0.5
3	4	0.6	0.67	0.43
3	5	0.6	0.83	0.38
4	5	0.8	0.83	0.44
4	6	0.8	1.0	0.4
5	6	1.0	1.0	0.45

Table 2.4: **Table of results for a Google search on the term "mouse".** The true positives (tp), false positives (fp), recall, fall-out, and precision for the list of hits where the subject of interest is the biological organism. We begin with the three images shown at the top and move down the list in order of relevance as calculated by Google. False positive rows are labeled with a gray background.

found in the venom of the Chinese scorpion *Buthus martensi*. In this case the parameter of "relevance" is the E-value, based on the model of extreme value statistics and the various quantities are tallied in the same way, by moving progressively down the list. For 1big_1 there are a total of 18 true hits that also belong to the same fold class, class number 2366. Since these results are for a single query, the number of false positives is also equal to the number of errors per query. In the actual computation, the full results of this database query would be pooled with the results from the other 13,350 database queries to obtain a single list ranked by E-value. The number of true positives, false positives, as well as recall, fall-out and precision, and errors per query can then all be tallied as one moves down the pooled list.

Finally we note some limitations of the DALI clustering scheme. The first "false positive" in the search results shown in Table 2.5 has an E-value of 1×10^{-10} so from the sequence alignment one would feel confident that 1big_1 and 1hlyA_1 are homologous. An alignment with DaliLite (http://www.ebi.ac.uk/DaliLite/) of chain A from these two PDB files confirms that they are structural homologs with an estimated z-score of 4.1. However, in the process of clustering domains some relationships such as these are lost: although all proteins within a fold class have a mutual z-score of 2 or higher, not all proteins with z-scores higher than 2 can be clustered into the same fold class. Ideally one would have all structural pairwise alignments and z-scores from DALI and measure the correlation of those scores with the pairwise SSEARCH sequences alignment results. Such a comparison was not possible because pairwise alignments of all domains in the PDB has proved impractical

Domain	Fold ID	Description	E-value	tp	fp	Recall	Fall-out	Precision
2bmt_1	2366	TOXIN BMTX2	2.4×10^{-15}	1	0	0.0556	0	1
1lglA_1	2366	BEKM-1 TOXIN	3.9×10^{-12}	2	0	0.111	0	1
1hp2A_1	2366	TITYUSTOXIN K ...	8.1×10^{-12}	3	0	0.167	0	1
1lir_1	2366	LQ2	2.3×10^{-11}	4	0	0.222	0	1
1hlyA_1	2379	HONGOTOXIN 1	1.0×10^{-10}	4	1	0.222	7.5×10^{-5}	0.8
1mtx_1	2366	MARGATOXIN	2.0×10^{-10}	5	1	0.278	7.5×10^{-5}	0.833
1m2sA_1	2374	TOXIN BMTX3	5.5×10^{-10}	5	2	0.278	0.00015	0.714
1sxm_1	2366	NOXIUSTOXIN	3.4×10^{-9}	6	2	0.333	0.00015	0.75
1wpdA_1	2492	MTX-HSTX1	6.0×10^{-9}	6	3	0.333	0.000225	0.667
1n8mA_1	2380	POTASSIUM CHANNEL ...	6.5×10^{-7}	6	4	0.333	0.0003	0.6
1bah_1	2366	CHARYBDOTOXIN	2.4×10^{-6}	7	4	0.389	0.0003	0.636
1qkyA_1	2366	TOXIN 7 ...	5.4×10^{-6}	8	4	0.444	0.0003	0.667
1tsk_1	2366	TS KAPA	7.1×10^{-6}	9	4	0.5	0.0003	0.692
1v56A_1	2381	SPINOXIN	2.0×10^{-5}	9	5	0.5	0.000375	0.643
2ktx_1	2366	KALIOTOXIN	3.1×10^{-5}	10	5	0.556	0.000375	0.667
1sco_1	2366	SCORPION TOXIN ...	3.1×10^{-5}	11	5	0.611	0.000375	0.688
1agt_1	2366	AGITOXIN 2 ...	6.3×10^{-5}	12	5	0.667	0.000375	0.706
1wmtA_1	2373	ISTX	6.7×10^{-5}	12	6	0.667	0.00045	0.667
1quzA_1	2366	HSTX1 TOXIN	0.00023	13	6	0.722	0.00045	0.684
1c49A_1	2366	TOXIN K-BETA	0.00034	14	6	0.778	0.00045	0.7

Table 2.5: **Results of a database query using SSEARCH** with 1big_1 of fold class 2366. The identification code, fold number, description, and E-value from the SSEARCH results are shown in the left half of the table. On the right half we show a running tally of the number of true and false positives found (tp and fp) together with the values for recall, fall-out, and precision. Rows with false positive hits are shown with a gray background.

to maintain without more computational resources (Sakari Kääriäinen, personal communication, August 15, 2007). The consequence for the current work is that there are some pairwise relationships like that between 1big_1 and 1hlyA_1 which are classified incorrectly as false positives.

2.2.8 Reference sequence alignments

Structural alignment of protein domains with the DALI method produces a reference list of structurally equivalent pairs of residues [42]. We compared these structure-based alignments with the alignments produced by SSEARCH and tallied the fraction of structurally equivalent residues found by SSEARCH local alignments. The database of structurally equivalent residues, *dali_fragments*, was obtained from the DALI downloads website [45].

2.3 Results

A scoring matrix should ideally be able to both detect related pairs of proteins (true positives) and reject non-related pairs (false positives); these properties are termed sensitivity and selectivity, respectively, and in many instances they compete with one another in the sense that as a matrix is tuned to be more sensitive it often loses selectivity and vice versa. The three curves analyzed in this work are precision vs. recall, recall vs. fall-out (also called the Receiver Operating Characteristic or ROC curve), and recall vs. EPQ (also called the coverage vs. errors per query or CVE plot), all parametrized by increasing E-value. We define the mean pooled precision to be the integral of the precision vs. recall curve for the combined list of search results; this number gives the average precision achieved over the entire range of recall and HSDM17 achieves the best result by this metric. The area under the ROC curve measures the ability of a matrix to identify related pairs by assigning them lower E-values than pairs of proteins that are not related over the entire list of pooled results; SDM12 is the top performer here. Both the mean pooled precision and the area under the ROC curve measure the sensitivity of a matrix since they are derived from the whole pooled list of results. Finally the recall at 0.01 EPQ gives the number of true positives returned at a fixed, low error rate, and gives an indication of the selectivity of a substitution matrix. Recall may be normalized in several ways, as defined by Green and Brenner [47]. Recall without normalization gives equal weight to each true positive relationship; quadratic normalization weights true positive hits so that each fold represented in the database has equal weight. Linear normalization is a compromise between these two, giving each sequence in the database equal weight, and is meant to take into account the fact that folds are not equally represented in Nature [20]. We find that the top performer in recall at 0.01 EPQ with linear normalization to be GBMR4; this matrix also ranked in the top ten in recall without normalization and under quadratic normalization (see Table 2.7).

We wish to also note that many reduced alphabets beyond the three we mentioned above outperform the BLAST-default matrix, BL62 with 11/1 gaps. Among the 151 scoring matrices tested in this work, BL62 ranked 38th overall in area under the ROC curve, 18th in mean pooled precision, and 111th, 102nd, and 104th in recall at 0.01 EPQ with no, linear, and quadratic normalization, respectively. In the remaining plots we will compare the top performers in mean pooled precision (HSDM17), area under the ROC curve (SDM12), and recall at 0.01 EPQ (GBMR4) with one another, using BL62 11/1 as the baseline.

2.3.1 Mean pooled precision

Precision vs. recall curves are shown in panel A of Fig. 2.14 for GBMR4, HSDM17, and SDM12; the mean pooled precision is the area under this curve. A perfect method would have a mean pooled precision value of unity, maintaining 100% precision until all true positives have been identified. The mean pooled precision for all of the HSDM, SDM, and GBMR alphabets is plotted in panel B of Fig. 2.14. Section A of Table 2.6 shows that mean pooled precision tends to favor larger alphabets, and the HSDM family of alphabets consistently achieves good results in this area.

Note that even the strongest performers in mean pooled precision cannot maintain a high level of precision beyond a recall value of about 0.4. This means that only about 40% of the total number of true positives can be reliably identified before additional true positives in the list of hits become buried in a flood of false positives in a sort of "needle-in-a-haystack" situation. The steep dropoff in mean precision in Fig. 2.14A indicates the limits of what can be achieved with pairwise sequence alignment.

2.3.2 Area under the ROC curve

The area under the ROC curve has a very specific interpretation: it is equal to the probability of assigning a lower Expect value to a true positive than to a false positive [48]. Therefore it gives a measure of the sensitivity of a scoring matrix to related sequences over the entire pooled list of results. The top 10 finishers according to this metric are shown in section B of Table 2.6. The top overall performer in detecting structurally related proteins by pairwise search is the SDM12 matrix; another notable high performer is LZ-MJ6 which finished in the top ten with only six letters. Receiver Operating Characteristic curves are shown in Fig. 2.15A for SDM12, HSDM17, and GBMR4. The total area under the curve vs. number of letters in these schemes is shown in panel B; note that the area under the ROC curve does not necessarily increase monotonically with alphabet size.

Although it is of interest that HSDM17 maintains the best selectivity as measured by mean pooled precision and SDM12 the best sensitivity as measured by the area under the ROC curve, what is of most interest to a typical user of an alignment program with a query protein and a target database is to find a scoring matrix that will yield the most number of true positives at a fixed, low error rate. Operationally, a researcher will have an intuition for what E-values indicate hits that are likely to be significant and will ignore hits below that intuitive threshhold. In the DALI database we used there are more than 150 times as many false positives as true positives, so much of the advantage shown by HSDM17 and SDM12 is in a regime beyond what could be reasonably processed "by hand". Therefore

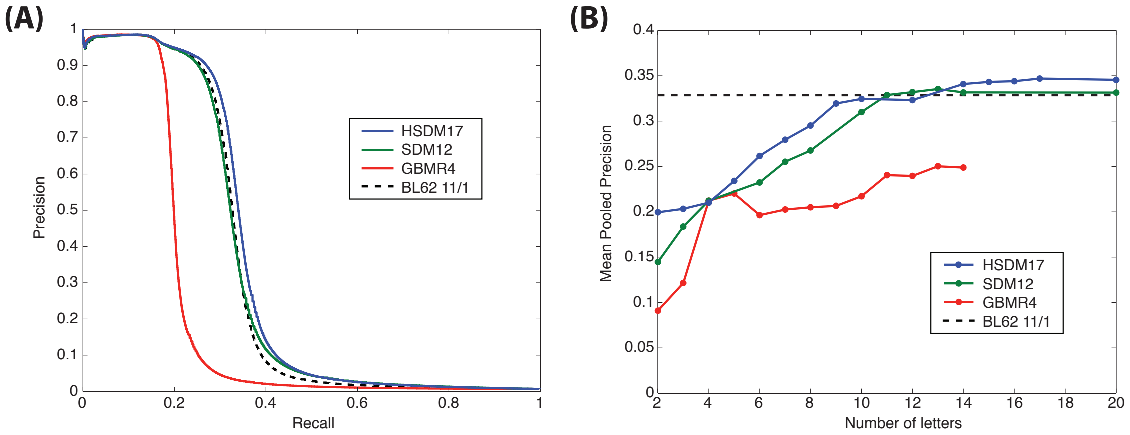


Figure 2.14: **Reduced alphabet performance in mean pooled precision.** (A) Precision vs. recall curves for the top reduced alphabet performers. The mean pooled precision is the area under this curve and indicates the ability of a particular matrix to maintain high selectivity over a wide range of error rates. At some point, each matrix loses the ability to selectively reject false positives and the curve drops precipitously to low precision values. (B) Mean pooled precision indicates the average precision achieved by a matrix over the entire range of recall. A perfect method would achieve a mean pooled precision value of unity, with all true positives ranked ahead of false ones. The HSDM17 matrix is the top performer in this metric; the dashed black lines in panels A and B show the performance of BL62 11/1 for reference. Points indicate reduced alphabets that were tested; the connecting lines are a guide to the eye.

we also examine the performance of reduced alphabets in recall of true relationships at an error per query rate of 0.01.

	(A) Mean pooled precision			**(B) Area under ROC curve**		
Rank	*Scheme*	*Letters*	*MPP*	*Scheme*	*Letters*	*AUC*
1	HSDM	17	0.347	SDM	12	0.801
2	BL62 7/1	20	0.347	SDM	11	0.800
3	BL50 11/1	20	0.346	SDM	13	0.800
4	LZ-BL	16	0.346	HSDM	17	0.796
5	HSDM	20	0.345	SDM	14	0.795
6	HSDM	16	0.344	LZ-MJ	6	0.793
7	HSDM	15	0.343	HSDM	20	0.791
8	HSDM	14	0.341	HSDM	16	0.789
9	LZ-BL	15	0.339	HSDM	9	0.784
10	SDM	13	0.335	HSDM	15	0.783

Table 2.6: **Top 10 performers in mean pooled precision (MPP) and area under the Receiver Operating Characteristic curve (AUC).** Mean pooled precision is a measure of the selectivity of a matrix, i.e., its ability to retain high recall of true positive relationships at low error rates. The area under the ROC curve measures the sensitivity of a matrix to true positive alignments over the entire list of results.

2.3.3 Recall at 0.01 EPQ

The second measure of the selectivity of each reduced alphabet scheme was calculated as the recall (also called coverage) at 0.01 errors per query; this is the metric of the most practical interest. Panel A of Fig. 2.16 shows the recall vs. error rate curves under linear normalization for GBMR4, HSDM17, and SDM12 with better-performing matrices generating curves that tend toward the lower-right hand corner, indicating high recall at low error rates. Comparing this with panel A of Fig. 2.14 we can see that GBMR4 is able to maintain the highest level of precision initially, but it rapidly loses precision at higher recall values. Panel B of Fig. 2.16 shows the recall at 0.01 EPQ with linear normalization vs. number of letters for the GBMR, SDM, and HSDM alphabets. Table 2.7 shows the top ten ranked matrices in recall at 0.01 EPQ under each of the normalizations described earlier with GBMR4 and other relatively small alphabets being the top performers.

These results would seem to indicate that reduced alphabets offer an advantage of immediate practical value over currently used matrices based on the full alphabet. To further investigate this possibility we performed all vs. all alignments with HSDM17, SDM12, GBMR4, and BL62 11/1 using proteins belonging to the same SCOP superfamily to define true positives. The results of a preliminary study with the scop40 and scop95 sequence

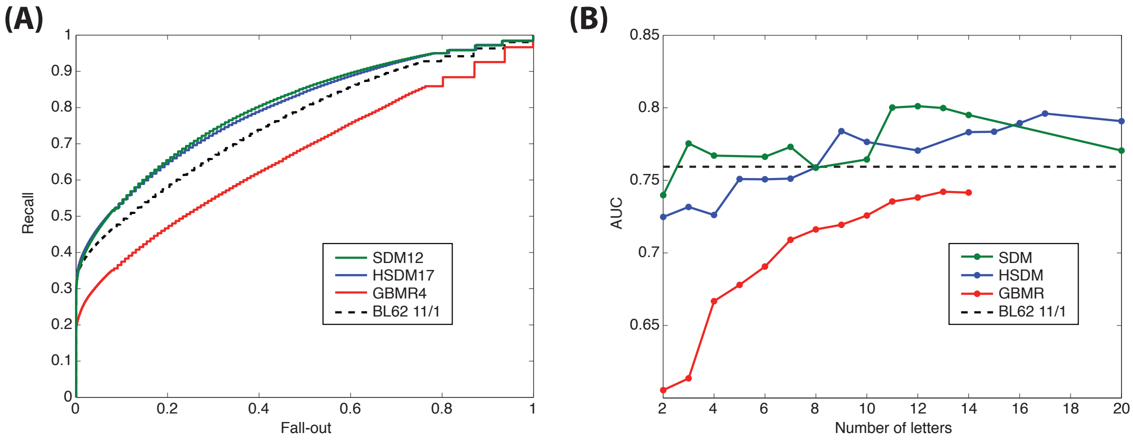


Figure 2.15: **Reduced alphabet performance in area under the Receiver Operating Characteristic curve.** (A) Receiver Operating Characteristic (ROC) curves for the top performing alphabets. The integral of this curve gives a measure of how well the entire pooled list of hits is sorted; a perfect method would have an ROC area of unity. (B) Overall sensitivity of the SDM alphabets as measured by the area under the ROC curve. The level of sensitivity of BL62 11/1 is shown with the black dashed line. Points indicate reduced alphabets that were tested; the connecting lines are a guide to the eye.

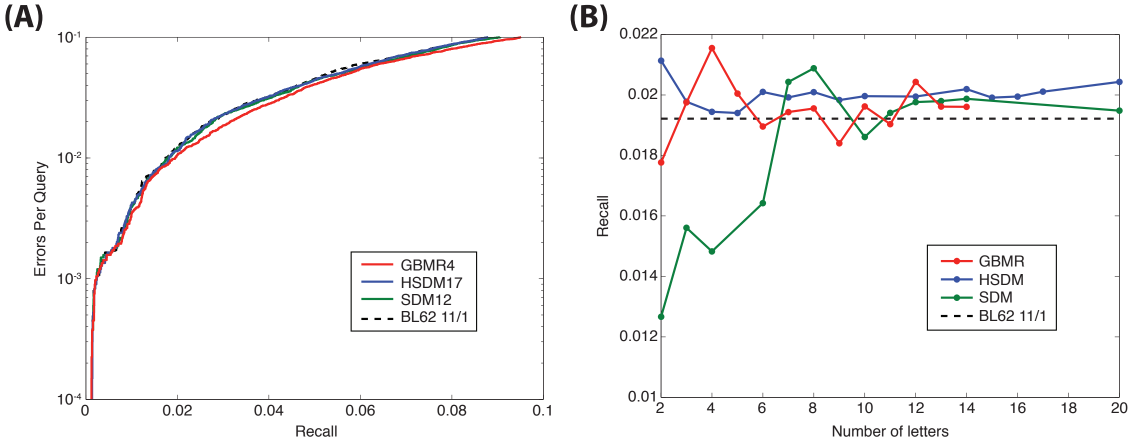


Figure 2.16: **Reduced alphabet performance in recall vs. errors per query with linear normalization.** (A) Linearly normalized recall (or coverage) vs. the number of errors per query (EPQ). Curves that tend toward the lower right-hand corner perform better, detecting more true positives at a given error rate. Small alphabets show good performance at lower error rates (EPQ < 0.1) with GBMR4 being the top performer. (B) Recall with linear normalization at 0.01 EPQ for various numbers of letters in the GBMR, HSDM, and SDM reduced alphabet schemes. The level of performance of BL62 11/1 is shown with the black dashed line. Points indicate reduced alphabets that were tested; the connecting lines are a guide to the eye.

	Recall at 0.01 EPQ								
	No normalization			**Linear normalization**			**Quadratic normalization**		
Rank	*Scheme*	*Letters*	*Recall*	*Scheme*	*Letters*	*Recall*	*Scheme*	*Letters*	*Recall*
1	GBMR	6	0.0097	GBMR	4	0.022	LZ-BL	7	0.036
2	GBMR	7	0.0089	CB/LW	2	0.021	GBMR	4	0.036
3	GBMR	8	0.0085	HSDM	2	0.021	HSDM	2	0.036
4	GBMR	9	0.0067	LZ-BL	7	0.021	HSDM	20	0.036
5	DSSP	2	0.0044	SDM	8	0.021	LZ-BL	8	0.036
6	GBMR	10	0.0038	TD	2	0.021	GBMR	12	0.035
7	ML	2	0.0036	BL50 11/1	20	0.021	LW-I	9	0.035
8	SDM	6	0.0030	LZ-BL	6	0.020	GBMR	3	0.035
9	GBMR	4	0.0029	HSDM	20	0.020	CB/LW	2	0.035
10	CB/LW	2	0.0029	SDM	7	0.020	SDM	8	0.035

Table 2.7: **Top 10 performers in recall at 0.01 errors per query with default, linear, and quadratic normalization.** Like mean pooled precision, recall at 0.01 EPQ measures the selectivity of a matrix but is drawn from a limited set of hits such as a researcher might reasonably peruse manually, rather than the entire pooled list of search results which comprises approximately 170 million alignments.

databases, containing sequences sharing no more than 40% and 95% sequence identity, respectively, are compared with DALI in Table 2.8. We note that BL62 11/1 outperformed GBMR4 in linearly normalized recall at 0.01 EPQ with both SCOP databases. However larger reduced alphabets maintained their advantage in selectivity and sensitivity with SCOP since HSDM17 and SDM12 both achieved higher mean pooled precision, area under the ROC curve and linear recall at 0.01 EPQ scores than BL62 11/1. This indicates that small reduced alphabets can show an increased sensitivity and selectivity for proteins that are *structurally* related, the only criteria used by DALI, but seem to lose selectivity when criteria such as function and evolution are taken into account, as is done with the human-curated SCOP superfamily classification [49]. To the extent that performance with the SCOP and DALI databases indicates real world performance, these results suggest that larger alphabets like SDM12 and HSDM17 offer increased sensitivity and selectivity over the standard BL62 11/1 matrix based on a full alphabet.

2.3.4 Statistical significance of results

The three top-performing alphabets (GBMR4, SDM12, and HSDM17) are shown in Table 2.9 together with the results of BL62 11/1 shown for reference. We used the Bayesian bootstrap method developed by Price et al. [50] to evaluate the statistical significance of the successes of the reduced alphabets in comparison with the standard BL62 11/1 scoring matrix. First the differences in performance are tabulated between each pair of bootstrap

	scop40			**scop95**			**DALI**		
Scheme	*MPP*	*AUC*	*Recall*	*MPP*	*AUC*	*Recall*	*MPP*	*AUC*	*Recall*
GBMR4	0.089	0.658	0.100	0.259	0.727	0.204	0.212	0.667	0.022
SDM12	0.156	0.734	0.136	0.419	0.833	0.250	0.332	0.801	0.020
HSDM17	0.173	0.751	0.148	0.436	0.840	0.259	0.347	0.796	0.020
BL62 11/1	0.156	0.714	0.134	0.408	0.812	0.245	0.329	0.759	0.019

Table 2.8: **Comparison of results from all vs. all studies with scop40, scop95, and DALI.** In the SCOP results GMBR4 is unable to maintain its advantage in linearly normalized recall at 0.01 EPQ over BL62 11/1. However both SDM12 and HSDM17 are able to match or better the results of BL62 11/1 in mean pooled precision (MPP), area under the ROC curve (AUC) and linearly normalized recall at 0.01 EPQ. Version 1.71 of the scop40 and scop95 sequence databases were used.

replicas and then a z-statistic is calculated by dividing the mean of distribution of differences by its standard deviation. This statistic, rather than the difference in mean performance, was found to be the most sensitive for evaluating the significance of differences in performance between two scoring matrices [50]. We find a strongly significant z-value of 6.17 for the superior performance of SDM12 relative to BL62 11/1 in area under the ROC curve, and marginally significant z-values of 1.33 for HSDM17 vs. BL62 11/1 in mean pooled precision and 1.49 for GBMR4 vs. BL62 11/1 in recall at 0.01 EPQ with linear normalization.

		Recall	AUC	MPP
GBMR4	ADKERNTSQ YFLIVMCWH G P	**0.022(0.001)**	0.667(0.004)	0.212(0.006)
SDM12	A D KER N TSQ YF LIVM C W H G P	0.020(0.001)	**0.801(0.005)**	0.332(0.009)
HSDM17	A D KE R N T S Q Y F LIV M C W H G P	0.020(0.001)	0.796(0.004)	**0.347(0.008)**
BL62 11/1	A D K E R N T S Q Y F L I V M C W H G P	0.019(0.001)	0.759(0.005)	0.329(0.009)

Table 2.9: **The results for the top performing alphabets found in this study** in linearly normalized recall at 0.01 errors per query (GBMR4), area under the ROC curve (SDM12), and mean pooled precision (HSDM17) with the standard deviation of 1000 bootstrap replicas given in parentheses. Results for BL62 11/1 are shown for comparison.

2.3.5 Alignment accuracy

We evaluated how well pairwise sequence alignments with reduced alphabets identified pairs of residues that are structurally equivalent as defined by DALI. The results are shown in Fig. 2.17, plotted as the fraction of structurally equivalent residue pairs identified by SSEARCH using the SDM, HSDM, and GBMR reduced alphabet schemes. The curves tend to saturate at around 10 letters, implying that expanding the alphabet beyond this point does not improve the alignments but tends to increase their sensitivity to more recently diverged proteins. The top 10 finishers in alignment accuracy are shown in Table 2.10; HSDM and SDM show the best performance, which is not surprising given that they were derived from structurally equivalent pairs of residues [28]. It is interesting that the highly simplified GBMR4 alphabet is able to achieve nearly the same level of accuracy as the full BL62 11/1 matrix. The DALI database of structurally equivalent residues is an exceedingly challenging test of pairwise sequence comparison since those residues share only 11% identity overall; even the best alphabet, HSDM17, achieves exact agreement with less than one tenth of all residues in the DALI structural alignments.

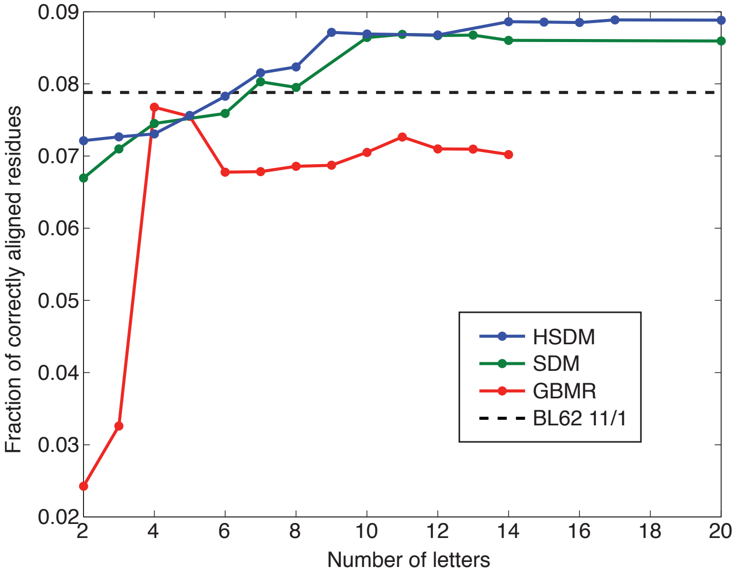


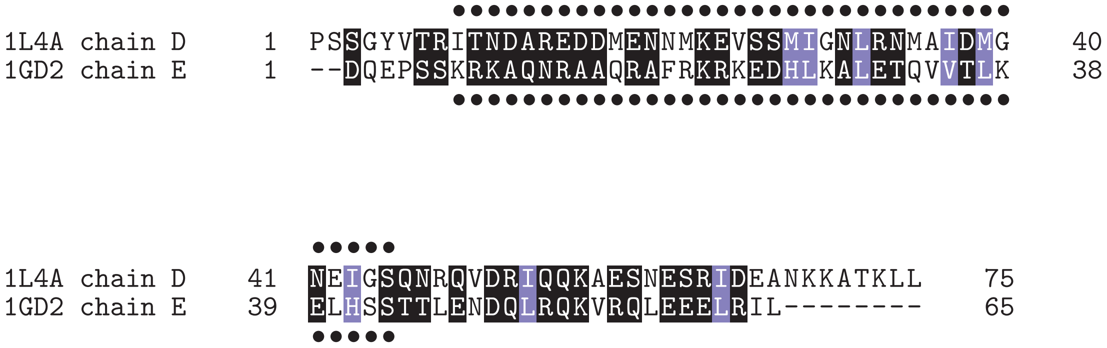
Figure 2.17: **Agreement of structural and sequence alignments.** The fraction of DALI equivalent residue pairs found by SSEARCH alignment is shown for various reduced alphabet schemes. Most of the gains are made as classes are added up until around 10 classes, after which the performance levels off. Points indicate reduced alphabets that were tested; the connecting lines are a guide to the eye.

The relative strengths and weaknesses of using a reduced alphabet for alignment of proteins is perhaps better illustrated by a specific example. In Fig. 2.18 and Fig. 2.19 below are shown alignments of a fragment of the SNAP-25 fusion protein (1L4A chain D) with the bZIP leucine zipper motif of the transcription factor Pap1 (1GD2 chain E) using the GBMR4 and BL62 11/1 substitution matrices. Both are coiled coils and GBMR4 is able to align all structurally equivalent residues correctly (37 total) according to the reference structure alignment by DALI. In contrast the standard BL62 11/1 matrix only aligns two

Rank	Scheme	Letters	Fraction aligned
1	HSDM	17	0.08887
2	HSDM	20	0.08882
3	HSDM	14	0.08862
4	HSDM	15	0.08857
5	HSDM	16	0.08849
6	HSDM	9	0.08714
7	HSDM	10	0.08691
8	SDM	11	0.08686
9	HSDM	12	0.08676
10	SDM	13	0.08675

Table 2.10: **The top 10 performers in agreement between sequence and structural alignments,** using DALI structurally equivalent residues as the "gold standard". As expected, the two structure-derived matrices, HSDM and SDM, completely dominate the results.

residues correctly. In this particular case, the coarse-grained GBMR4 alphabet is able to correctly identify the pattern of residues underlying the formation of these coiled coil proteins whereas BL62 is not. We note however that a coiled coil is a simple motif to generate and that, overall, BL62 11/1 is able to slightly outperform GBMR4.



```
1L4A chain D     1  PSSGYVTRITNDAREDDMENNMKEVSSMIGNLRNMAIDMG    40
1GD2 chain E     1  --DQEPSSKRKAQNRAAQRAFRKRKEDHLKALETQVVTLK    38

1L4A chain D    41  NEIGSQNRQVDRIQQKAESNESRIDEANKKATKLL    75
1GD2 chain E    39  ELHSSTTLENDQLRQKVRQLEEELRIL--------    65
```

Figure 2.18: **Pairwise sequence alignment of two coiled coil domains with the GBMR4 alphabet.** Structurally equivalent residues as determined by DALI are indicated with a black dot above and below the alignment. GBMR4 correctly identified all of the DALI-equivalenced residues in this case. In GBMR4, glycine and proline are specifically singled out, with the remaining residues forming hydrophobic (YFLIVMCWH) and polar (ADKERNTSQ) classes, shown with white characters on blue and black backgrounds, respectively.

```
1L4A chain D    1  -----PSSGYVTRITNDAREDDMENNMKEVSSMIGNLRNM   35
1GD2 chain E    1  DQEPSSKRKAQNRAAQRAFRKRKEDHLKALETQVVTLKEL   40


                           ●●
1L4A chain D   36  AIDMGNEIGSQNRQVDRIQQKAESNESRIDEANKKATKLL   75
1GD2 chain E   41  H-------SSTTLENDQLRQKVRQLEEELRIL--------   65
                           ●●
```

Figure 2.19: **Pairwise sequence alignment of two coiled coil domains with the BL62 11/1 matrix.** Structurally equivalent residues as determined by DALI are indicated with a black dot above and below the alignment. BL62 11/1 correctly identified only two DALI-equivalenced residues. Identical residues are shown with white characters on a blue background.

2.3.6 Comparison of detected relationships

It is also valuable to compare the hits returned by two matrices at a given errors per query level to see what types of relationships are more easily detected by one relative to another. We compared the hits returned by the SDM12 and BL62 11/1 matrices at or above 0.01 EPQ and found that each matrix finds about 3000 true positives at that error level. After separating out the hits that were unique to each matrix (they share 2724 hits in common) SDM12 was left with 271 unique hits and BL62 11/1 with 139. The approximate mean percent identity of the SDM12 unique hits is 60% whereas for BL62 11/1 it is 70%. Although SDM12 and BL62 have essentially identical relative entropy (-0.703 and -0.699 bits, respectively) SDM12 is able to detect more distant relationships than BL62. A histogram of the hits unique to each matrix is shown in Fig. 2.20.

2.4 Discussion

We find, perhaps counter to common intuition, that reduced alphabets *increase* the selectivity and sensitivity to pairs of proteins with structural similarity as measured by the mean pooled precision, area under the ROC curve, and recall at 0.01 errors per query (under all normalizations). In addition we found that reduced alphabets can return more distantly related pairs of proteins. This is in contrast to some earlier studies [25, 26, 40] which found that reduced alphabets would only result in losses in performance relative to a full alpha-

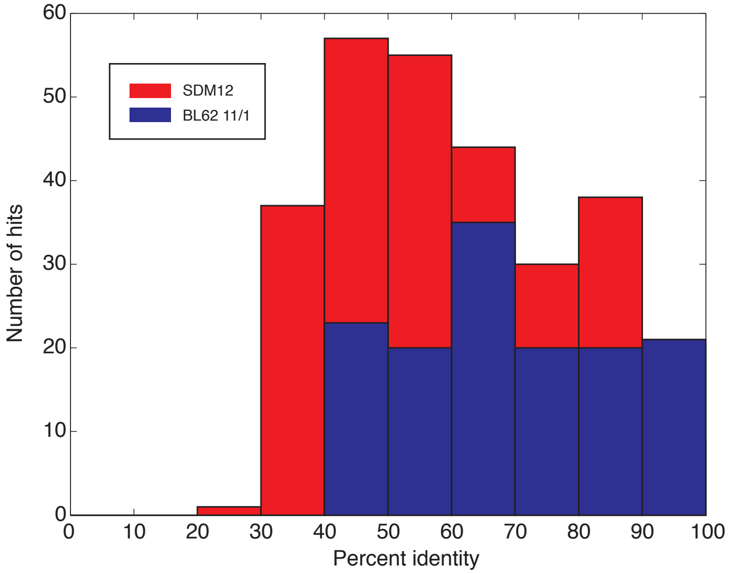


Figure 2.20: **Histogram of the hits at or above 0.01 errors per query unique to the SDM12 and BL62 11/1 matrices.** The results from SDM12 are both more numerous and shifted towards lower identity, showing its increased ability to detect more remote relationships.

bet. Landès and Risler also observed improved sensitivity with reduced alphabets; in an early study with aminoacyl-tRNA synthetases, LR10 showed an increased ability to identify distant homologs over methods using the full alphabet [38]. This work also adds to the encouraging results with reduced alphabets found by Melo and Marti-Renom [39], who tested the 20 letter Johnson-Overington matrix against several small reduced alphabets: WW5, GBMR5, ML4, MM5, and 100 randomly reduced five-letter alphabets. They found that the GBMR5 alphabet produced performance gains over the full matrix in alignment accuracy as measured by the root-mean-square deviation of C^{α} atoms after using the pairwise alignment as the initial seed for an optimal structural superposition.

As an example of how the reduced alphabets studied here might offer a practical advantage we estimate the number of additional true hits that would be found by the best alphabet in unnormalized recall at 0.01 EPQ, GBMR6, vs. the standard BL62 11/1 matrix. In the DALI pdb90 database there were 13,351 sequences with a total of 1,133,086 true relationships. The number of true relationships should scale as the square of the number of database sequences so with the current NCBI nr database, comprising 4.6 million sequences, and assuming the same scaling as with DALI pdb90 we would estimate that there are about

250 billion true relationships. Assuming in addition that GBMR6 and BL62 11/1 return the same fraction of true hits that they did with the DALI pdb90 database we find that GBMR6 returns an average of about 400 true hits per protein whereas for BL62 it is just 100 true hits.

A point of diminishing returns is observed in mean pooled precision (Fig. 2.14, panel B), area under the ROC curve (Fig. 2.15, panel B) and alignment accuracy (Fig. 2.17) at an alphabet size of about 10 letters. For instance, we find that in area under the ROC curve, HSDM10 achieves 98% of HSDM20's performance, GBMR10 achieves 96% of the performance of BL62 11/1, and SDM10 slightly exceeds the total area achieved by SDM20. Similar results have been observed elsewhere in reduced alphabet studies. Weathers et al. found that a Support Vector Machine model using reduced alphabets (as small as 4 letters) retained nearly as much ability to detect intrinsically disordered proteins as the full 20 letter alphabet [51]. Rackovsky and coworkers showed that using seven groups of amino acids best conserves information about local backbone conformation [34] and in a separate work found that current knowledge-based statistical potentials, though often assumed to discriminate between all 20 amino acids, actually only distinguish at most between about eight classes of residues [52]. In their work testing the ENERGI method for building of statistical potentials, Dill and Thomas found that using a 20 letter alphabet was no more successful than a 5 letter alphabet for fold detection, with the peak of detection occurring at an alphabet of 10 letters [31]. In their study of pairwise sequence alignment, Murphy et al. estimated that a minimum of a 10–12 letter alphabet is necessary to design foldable sequences for most protein families [25]. Fan and Wang also found that the minimum alphabet size for protein folding requires approximately 10 types of amino acids [53]. The consensus seems to be the there is little to lose and possibly gains to be made by properly clustering the amino acids.

It is interesting that the top performing alphabets, shown in Table 2.9, are compatible with one another in the sense that SDM12 can be derived from GBMR4 and HSDM17 can be derived from SDM12 by simply breaking down larger clusters into smaller ones without needing to interchange the grouping of any of the amino acids. In the GBMR4 alphabet glycine and proline are singled out as being structurally dissimilar from the other amino acids; the remaining two groups reflect a hydrophobic (YFLIVMCWH) and polar (ADKERNTSQ) classification. In this sense, the GBMR4 alphabet is a modest refinement of the simple HP concept. The SDM12 alphabet maintains clusters for acidic/basic (KER), polar (TSQ), aromatic (YF), and mostly aliphatic (LIVM) groups. Two non-intuitive results in these groupings are the omission of aspartic acid from the acidic/basic KER cluster and the inclusion of methionine in the otherwise aliphatic LIVM cluster. In HSDM17 only the strongest associations among these are maintained: acidic/basic (KE) and aliphatic (LIV). By clustering together amino acids with similar properties in this way we increase the signal to noise in our database searches and avoid over-assigning importance to differences among the naturally occurring amino acids.

We wish to note several promising avenues for further investigation which could not be pursued in this work for lack of sufficient time and computing resources. In theory an optimum alphabet could be searched for at each alphabet size by, e.g., Monte Carlo search. Likewise it would be ideal to optimize the gap penalties with respect to each reduced alphabet. Lack of sufficient computational resources made it impractical to carry out these optimizations. We chose to use the DALI database as our standard for determining structural relationships among proteins in the PDB over databases like, e.g., SCOP [49], because its determinations are informed only by structural similarity and require no human curation. Although we obtained some preliminary results with SCOP it would be instructive to compare the results using the DALI database to what would be obtained by testing all the reduced alphabets in this work using SCOP superfamilies as the “gold standard” for structural relatedness. Given the encouraging results shown by SDM12 and HSDM17 with both SCOP and DALI we believe that further investigation into the practical advantages of reduced alphabets for general use with pairwise alignment matrices merits additional exploration.

Another promising area for application of the results of this study is in the building of protein profiles or hidden Markov models (HMMs). Such models are built up from a multiple alignment of many putatively homologous proteins. At each position in the alignment, a number can be assigned for the probability of observing a particular amino acid based on the sequences in the multiple alignment. The simplest type of protein profile is simply a consensus sequence of the most commonly occurring amino acid at each position. One current limitation of these methods is the limited sample of sequences with which to build up the multiple alignment; experimentally determined sequences account for only a fraction of the total sequence space available to a given protein fold. By thinking of a protein as being made up of amino acids drawn from classes with particular physical properties we can leverage the physicochemical similarities of amino acids to help make up for this lack of statistics in our sampling of sequence space. This problem of undersampling was recognized by Sjölander et al. [54] who developed a method of Dirichlet mixtures for use with multiple alignments to improve detection of remote homologs. The method of Dirichlet mixtures estimates the most likely expected distribution of amino acids at a given position in a multiple alignment and could be extended to estimate the most likely expected distribution of *classes* of amino acids, as studied here, instead of individual amino acids.

One especially exciting opportunity for the reduced alphabet concept to be applied is the search for translational motors in prokaryotes. Motor proteins such as myosin and kinesin, which translate on the cellular cytoskeleton, are presently only known to exist in eukaryotes. Now that it has been established that prokaryotes also possess a cytoskeleton, the possibility arises that they may also have motors which use prokaryotic cytoskeletal filaments to translate. In the specific case of myosin and kinesin there is a heart-shaped ATPase domain, shown below[3] in Fig. 2.21, which was found to be common to both proteins and indicates an ancient ancestor of these two motors [55]. A combination of a reduced alphabet studied here together with the motif-based technique of Bork et al. or perhaps applying an HMM to a diverse set of known sequences of this heart-shaped domain should produce a "lure" which could be used to search prokaryotic genomes and identify possible homologs.

In sum, a reduced alphabet approach to building up protein profiles, motifs or HMMs may improve our ability to detect proteins with structural homology by leveraging our knowledge of the chemical properities of the amino acids in building up a physical picture of a fold.

[3] Adapted by permission from Macmillan Publishers Ltd: Nature 380(6574):550, copyright 1996.

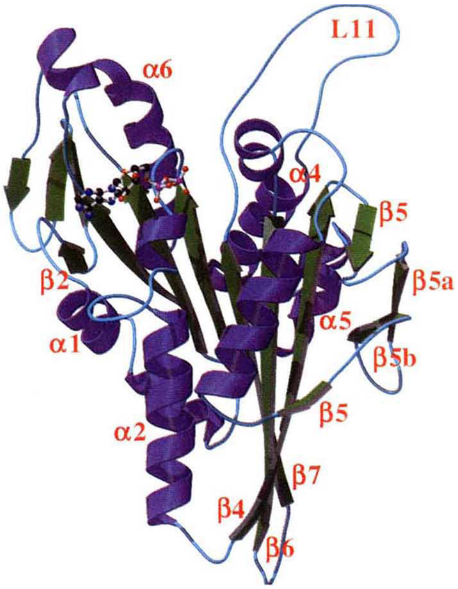


Figure 2.21: **The heart-shaped ATPase domain common to myosin and kinesin.** Figure adapted, with permission, from Reference [55].

Bibliography

[1] S B Needleman and C D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–453, 1970.

[2] T F Smith and M S Waterman. Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–197, 1981.

[3] I Korf, M Yandell, and J Bedell. *BLAST*. O'Reilly & Associates, Inc, 2003.

[4] S F Altschul, W Gish, W Miller, E W Myers, and D J Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3):403–410, 1990.

[5] K A Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24: 1501–1509, 1985.

[6] H Li, R Helling, C Tang, and N Wingreen. Emergence of preferred structures in a simple model of protein folding. *Science*, 273(5275):666–669, 1996.

[7] M H Hecht, A Das, A Go, L H Bradley, and Y Wei. De novo proteins from designed combinatorial libraries. *Protein Sci*, 13(7):1711–1723, 2004.

[8] N Nandhagopal, A A Simpson, J R Gurnon, X Yan, T S Baker, M V Graves, J L Van Etten, and M G Rossmann. The structure and evolution of the major capsid protein of a large, lipid-containing DNA virus. *Proc Natl Acad Sci USA*, 99(23):14758–14763, 2002.

[9] B Rost. Twilight zone of protein sequence alignments. *Protein Eng*, 12(2):85–94, 1999.

[10] R F Doolittle. *Of URFs and ORFs: A Primer on How to Analyze Derived Amino Acid Sequences*. University Science Books, 1986.

[11] R B Russell and G J Barton. Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins*, 14(2): 309–323, 1992.

[12] E Roberts, J Eargle, D Wright, and Z Luthey-Schulten. MultiSeq: unifying sequence and structure data for evolutionary analysis. *BMC Bioinformatics*, 7:382, 2006.

[13] W Humphrey, A Dalke, and K Schulten. VMD: Visual Molecular Dynamics. *J Mol Graph*, 14(1):27–28, 33–8, 1996.

[14] P J Kraulis. *MOLSCRIPT*: a program to produce both detailed and schematic plots of protein structures. *J Appl Crystallogr*, 24(5):946–950, Oct 1991.

[15] P Bork, C Sander, and A Valencia. An ATPase domain common to prokaryotic cell cycle proteins, sugar kinases, actin, and hsp70 heat shock proteins. *Proc Natl Acad Sci USA*, 89(16):7290–7294, 1992.

[16] M Munson, R O'Brien, J M Sturtevant, and L Regan. Redesigning the hydrophobic core of a four-helix-bundle protein. *Protein Sci*, 3(11):2015–2022, 1994.

[17] T J Magliery and L Regan. A cell-based screen for function of the four-helix bundle protein Rop: a new tool for combinatorial experiments in biophysics. *Protein Eng Des Sel*, 17(1):77–83, 2004.

[18] C E Schafmeister, S L LaPorte, L J Miercke, and R M Stroud. A designed four helix bundle protein with native-like structure. *Nat Struct Biol*, 4(12):1039–1046, 1997.

[19] D S Riddle, J V Santiago, S T Bray-Hall, N Doshi, V P Grantcharova, Q Yi, and D Baker. Functional rapidly folding proteins from simplified amino acid sequences. *Nat Struct Biol*, 4(10):805–809, 1997.

[20] A Grant, D Lee, and C Orengo. Progress towards mapping the universe of protein folds. *Genome Biol*, 5(5):107, 2004.

[21] F H Arnold. Unnatural selection: Molecular sex for fun and profit. *Engineering and Science*, LXII(1–2):41–50, 1999.

[22] A M Gutin and E I Shakhnovich. Ground state of random copolymers and the discrete random energy model. *J Chem Phys*, 98(10):8174–8177, 1993.

[23] R Helling, H Li, R Melin, J Miller, N Wingreen, C Zeng, and C Tang. The designability of protein structures. *J Mol Graph Model*, 19(1):157–167, 2001.

[24] S Henikoff and J G Henikoff. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci USA*, 89(22):10915–10919, 1992.

[25] L R Murphy, A Wallqvist, and R M Levy. Simplified amino acid alphabets for protein fold recognition and implications for folding. *Protein Eng*, 13(3):149–152, Mar 2000.

[26] T Li, K Fan, J Wang, and W Wang. Reduction of protein sequence complexity by residue grouping. *Protein Eng*, 16(5):323–30, 2003.

[27] M S Johnson and J P Overington. A structural basis for sequence comparisons. An evaluation of scoring methodologies. *J Mol Biol*, 233(4):716–738, 1993.

[28] A Prlić, F S Domingues, and M J Sippl. Structure-derived substitution matrices for alignment of distantly related sequences. *Protein Eng*, 13(8):545–550, 2000.

[29] E J Gumbel. *Statistics of Extremes*. Columbia University Press, New York City, NY, 1958.

[30] S Karlin and S F Altschul. Applications and statistics for multiple high-scoring segments in molecular sequences. *Proc Natl Acad Sci USA*, 90(12):5873–5877, 1993.

[31] P D Thomas and K A Dill. An iterative method for extracting energy-like quantities from protein structures. *Proc Natl Acad Sci USA*, 93(21):11628–11633, 1996.

[32] L A Mirny and E I Shakhnovich. Universally conserved positions in protein folds: reading evolutionary signals about stability, folding kinetics and function. *J Mol Biol*, 291(1):177–196, 1999.

[33] V I Abkevich and E I Shakhnovich. What can disulfide bonds tell us about protein energetics, function and folding: Simulations and bioinformatics analysis. *J Mol Biol*, 300(4):975–985, 2000.

[34] A D Solis and S Rackovsky. Optimized representations and maximal information in proteins. *Proteins*, 38(2):149–164, 2000.

[35] C A F Andersen and S Brunak. Representation of protein-sequence information by amino acid subalphabets. *AI MAGAZINE*, 25(1):97–104, Spr 2004.

[36] M Cieplak, N S Holter, A Maritan, and J R Banavar. Amino acid classes and the protein folding problem. *J Chem Phys*, 114(3):1420–1423, 2001.

[37] S Miyazawa and R L Jernigan. Residue-residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading. *J Mol Biol*, 256(3):623–644, 1996.

[38] C Landes and J L Risler. Fast databank searching with a reduced amino-acid alphabet. *Comput Appl Biosci*, 10(4):453–454, 1994.

[39] F Melo and M A Marti-Renom. Accuracy of sequence alignment and fold assessment using reduced amino acid alphabets. *Proteins*, 63(4):986–995, 2006.

[40] X Liu, D Liu, J Qi, and W-M Zheng. Simplified amino acid alphabets based on deviation of conditional probability from random background. *Phys Rev E*, 66(2 Pt 1): 021906, 2002.

[41] J Wang and W Wang. A computational approach to simplifying the protein folding alphabet. *Nat Struct Biol*, 6(11):1033–1038, 1999.

[42] L Holm and C Sander. Protein structure comparison by alignment of distance matrices. *J Mol Biol*, 233(1):123–138, 1993.

[43] L Holm and C Sander. Dictionary of recurrent domains in protein structures. *Proteins*, 33(1):88–96, 1998.

[44] S Dietmann, J Park, C Notredame, A Heger, M Lappe, and L Holm. A fully automatic evolutionary classification of protein folds: Dali Domain Dictionary version 3. *Nucleic Acids Res*, 29(1):55–57, 2001.

[45] Dali downloads. URL `http://ekhidna.biocenter.helsinki.fi/dali/downloads`.

[46] W R Pearson. Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. *Genomics*, 11(3):635–650, 1991.

[47] R E Green and S E Brenner. Bootstrapping and normalization for enhanced evaluations of pairwise sequence comparison. *Proc of the IEEE*, 90(12):1834–1847, Dec 2002.

[48] J A Hanley and B J McNeil. The meaning and use of the area under a Receiver Operating Characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.

[49] A G Murzin, S E Brenner, T Hubbard, and C Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, 247 (4):536–540, 1995.

[50] G A Price, G E Crooks, R E Green, and S E Brenner. Statistical evaluation of pairwise protein sequence comparison with the Bayesian bootstrap. *Bioinformatics*, 21(20): 3824–3831, Oct 2005.

[51] E A Weathers, M E Paulaitis, T B Woolf, and J H Hoh. Reduced amino acid alphabet is sufficient to accurately recognize intrinsically disordered protein. *FEBS Lett*, 576(3): 348–352, 2004.

[52] I B Kuznetsov and S Rackovsky. Discriminative ability with respect to amino acid types: assessing the performance of knowledge-based potentials without threading. *Proteins*, 49(2):266–284, 2002.

[53] K Fan and W Wang. What is the minimum number of letters required to fold a protein? *J Mol Biol*, 328(4):921–926, May 2003.

[54] K Sjölander, K Karplus, M Brown, R Hughey, A Krogh, I S Mian, and D Haussler. Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Comput Appl Biosci*, 12(4):327–345, 1996.

[55] F J Kull, E P Sablin, R Lau, R J Fletterick, and R D Vale. Crystal structure of the kinesin motor domain reveals a structural similarity to myosin. *Nature*, 380(6574): 550–555, 1996.

Chapter 3

Membrane shape as a reporter for applied forces

3.1 Introduction

Phospholipid bilayers are responsible for forming the boundaries between cells and organelles. This partitioning function is particularly apparent in eukaryotic cells, where organelles like the endoplasmic reticulum, Golgi apparatus and mitochondrion adopt complex shapes. An example of the labyrinthine morphologies adopted by biomembranes is illustrated by the endomembrane system shown[1] in Fig. 3.1. The roughly spherical, double-membraned nucleus is continuous with the endoplasmic reticulum, an intricate tubular network which directs constant traffic into the Golgi apparatus which in turn trafficks with the outer plasma membrane and other organelles of a eukaryotic cell. Much more than a passive scaffolding element, membranes in cells are dynamic elements which are continually reshaped and exchanged between the various parts of a cell.

Many of the mechanisms responsible for the remodelling of cellular membranes have now been studied in molecular detail. In a recent review by McMahon and Gallop [3], five principal molecular methods for bending a membrane were described as shown[2] in Fig. 3.2. Some of the methods used by cells to restructure their membranous features include the aggregation of transmembrane proteins [4, 5], active remodeling by molecular motors [6–8], assembly and disassembly of cytoskeletal filaments [9–11], and direct or indirect scaffolding by proteins [12, 13].

Together with the recent molecular understanding of the mechanisms that remodel mem-

[1] Fig. 3.1B reprinted by permission from PNAS 98(5):2399, copyright 2000.

[2] Reprinted by permission from Macmillan Publishers Ltd: Nature 438(7068):590, copyright 2005.

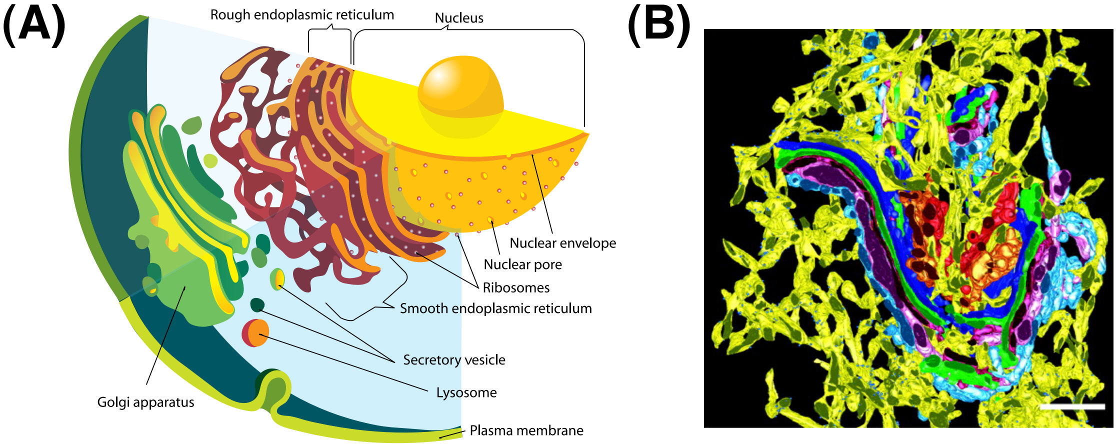


Figure 3.1: **The endomembrane system of a eukaryotic cell.** The membranes of eukaryotic cells adopt fascinating shapes such as tubes, vesicles, and flattened disks in support of their cellular function. **Panel A.** Schematic diagram of the endomembrane system [1]. **Panel B.** Three-dimensional reconstruction from electron cryotomography of the endoplasmic reticulum (yellow) surrounding the cisternae of the Golgi apparatus (multi-colored); scale bar = 500 nm. Figure reproduced, with permission, from Reference [2].

branes have developed exciting new technologies for imaging cell membranes at unprecendented resolution. Perhaps the most dramatic examples come from the technique of electron cryotomography which enables a detailed three-dimensional reconstruction of the contents of a cell at a resolution of a few nanometers from a "tilt series", images of a sample taken at various angles [14]. For instance, electron cryotomography has enabled the reconstruction of high-resolution, three-dimensional models of organelle structure (the mitochondrion [15], the Golgi apparatus [16], and endoplasmic reticulum [16]) as well as HIV viral entry [17], the architecture surrounding synaptic vesicles [18], and the endocytotic machinery [19]. An example of an image of a mitochondrion imaged with this technique is shown[3] in Fig. 3.3. Qualitative insights gained from such imaging has been substantial, and in the particular case of the mitochondrion has corrected a decades-old misperception of the structure of the inner mitochondrial membrane based on two-dimensional imaging (see Fig. 3.4).

Although these microscopy-based membrane structures have led to revolutions in our qualititative understanding of membrane morphology, no general technique exists as yet for their quantitative analysis. For instance, it has long been postulated that the mechanics

[3] Reprinted from Trends Biochem Sci, Frey and Mannella "The internal structure of mitochondria" 25:319–324 (2000) with permission from Elsevier.

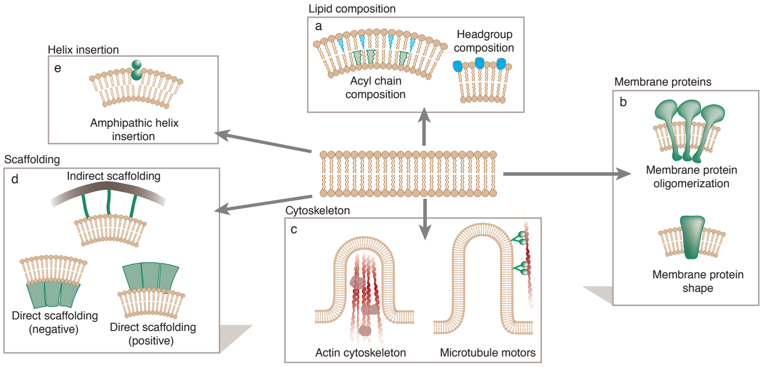


Figure 3.2: **The five ways to bend a membrane** as identified by McMahon and Gallop. A variety of methods are used by cells to induce positive and negative curvature and shape their membranes into the intricate structures observed under a microcope. Figure reproduced, with permission, from Reference [3].

of the membrane play an important role in the aggregation of membrane proteins [22], and recent simulations have also lent support to this idea [23]. In this case, knowledge of the mechanical forces applied by aggregating membrane proteins to an experimentally observed membrane structure would shed further light on their size, distribution, and mutual interaction. Further questions along these lines could include that of the nature of the scaffold, or forces, that give rise to the cristae structure of mitochondria seen in Fig. 3.3. The quantitative measurement of the location and magnitude of the forces applied by cellular machinery promises to yield important insights into the detailed mechanisms of membrane remodeling processes, from the life cycle of a membrane-bound virus to the scaffolding underlying organellar structure.

In this chapter we propose a general technique that may be applied to fluid phospholipid bilayers to directly measure the forces applied to the membrane from knowledge of its shape alone, as observed by, e.g., electron cryotomography. After developing the necessary theoretical and computational tools, we apply the method to experimental data from an optical tweezer microscope used to draw membrane tethers out of giant unilamellar vesicles or GUVs. This experiment provided a convenient test for validating the measurement of forces from an observed membrane conformation since the applied force is known independently by measurement from the optical tweezers. The technique applied here, which is easily adapted to the axisymmetric conformations observed in the optical tweezing experiments, is equally applicable to arbitrary membrane shapes and has been implemented in a

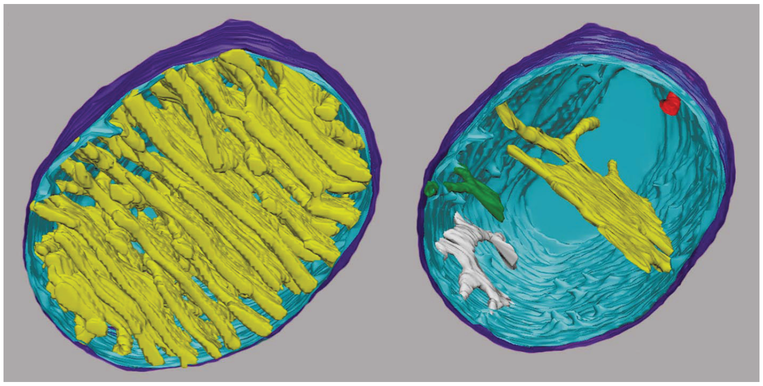


Figure 3.3: **Two views of a three-dimensional reconstruction of a mitochondrion** from a chick cerebellum cell. On the left the full reconstruction is shown; on the right, only a few cristae are shown to illustrate typical details of these structures. The outer membrane is shown in purple. The inner membrane of the mitochondrion is made up of cristae (yellow) interconnected by small tubules termed crista junctions to a surrounding envelope of inner membrane which encloses all the cristae (light blue). Figure reproduced, with permission, from Reference [15].

general finite element framework [24]. We envision that the measurement of applied force by analysis of membrane conformation offers exciting possibilities for the analysis of *in vivo* membranes, though the heterogeneity and extraction of reliable conformational data from tomograms are challenges that remain to be overcome in this area [25].

The remainder of this chapter is organized as follows. In Section 3.2 we review the necessary theoretical preliminaries to measure the force applied to a fluid lipid bilayer based on its observed shape. In Section 3.3 we review the experimental setup used to trap the optical beads and apply a known force with optical tweezers to giant unilamellar vesicles and in Section 3.4 elaborate the computational methods used to extract the membrane conformation and numerically calculate the applied force. Finally in Section 3.5 we show the results of the membrane force measurements and discuss the implications in Section 3.6. Additional material can be found in Appendix B on artifacts encountered during the experiments as well as detailed derivations of the normal force from the Helfrich energy and the dependence of membrane tether radius on the applied force.

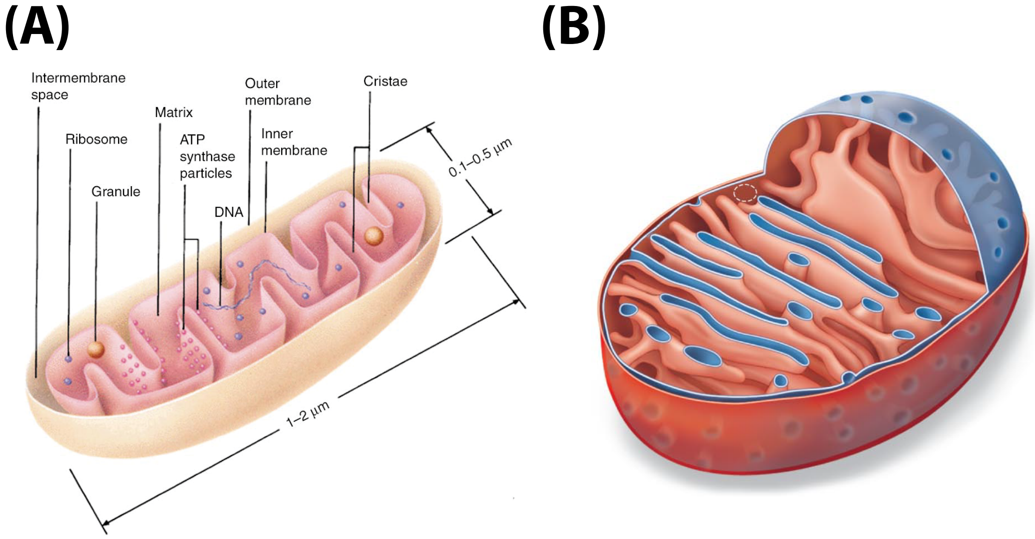


Figure 3.4: **Comparison of the old view of mitochondrial membrane structure with the new view informed by electron cryotomography. (A) A outdated textbook view of the inner mitochondrial membrane.** The so-called "baffle" model of the inner mitochondrial membrane which visualized the cristae as invaginations which form large folds continuous with the surrounding envelope of inner membrane. Figure reproduced from Reference [20], published in 2000. **(B) A current textbook view of the inner mitochondrial membrane.** The cristae are now known to connect to the surrounding inner membrane not through continuous folds but rather through small tubules termed crista junctions. Figure reproduced from Reference [21], published in 2008.

3.2 Theory

The mechanical theory of fluid-phase lipid membranes has been the subject of extensive study [25, 26] and experiments suggest that the response of membranes to the application of force is well-approximated by the theory of Helfrich, Canham, and Evans [27–29]. The Helfrich model describes the conformational energy of the membrane in terms of an areal energy density ε that is a function of the local shape of the membrane:

$$\varepsilon = \tfrac{1}{2} k_C \left(S - C\right)^2 + k_G K, \tag{3.1}$$

where K is the Gaussian curvature (the product of the principal curvatures) and S is the sum of the principal curvatures; C is the spontaneous curvature. The spontaneous curvature describes the spontaneous bending of the membrane induced by asymmetries in the lipid composition or chemical environment of the individual leaflets. The bending modulus k_C is typically 10–20 k_BT [30] and k_G is the Gaussian bending modulus. By the Gauss-Bonnet theorem, the k_GK term in the Helfrich energy contributes only a topological term to the free energy and is therefore irrelevant for describing closed membranes (see, e.g., Nakahara [31]). We also wish to note that the Helfrich energy is functionally identical to the Area-Difference Elasticity model (ADE) up to the first variation. The effect of an area difference between the two leaflets of the membrane is to produce an effective spontaneous curvature. Incorporation of the parameter C in the Helfrich model can therefore be interpreted to include the contribution of ADE. More detail on this can be found, e.g., in Seifert's review of the mechanics of fluid membranes [26].

Given that we may neglect the term proportional to K in the energy density, the functional integral which we will use to find local forces is:

$$E = \int \mathrm{d}A \left[\tfrac{1}{2}k_C\left(S^2 - 2SC\right) + \alpha\right] + \int \mathrm{d}V\, p, \tag{3.2}$$

where we have included the membrane tension α and osmotic pressure p as Lagrange multipliers. Note that we absorbed the constant $\frac{1}{2}k_CC^2$ term into the definition of the tension. The elastic force density of the membrane is determined by performing the first variation of the Helfrich energy [32]. Since the membrane is fluid in-plane [33, 34], the equilibrium elastic force per unit area is in the direction perpendicular to the membrane:

$$f_\perp = S\alpha - p - k_C\left[\nabla^2 S - 2K\left(S - C\right) + \tfrac{1}{2}S^3\right], \tag{3.3}$$

where the parameter p is the osmotic pressure. A detailed derivation of Eq. 3.3 for the elastic force density $f_\perp$ is given in Appendix B.2.

The force density depends on the local membrane geometry, which is directly observable, and three additional parameters: the pressure, tension, and spontaneous curvature, which must be determined. To compute these unknown parameters, we apply the Proximal Equilibrium Approximation which uses a maximum-likelihood principle to determine the values of pressure, tension, and spontaneous curvature which predict an equilibrium conformation closest to the observed conformation.

The total force applied by a region of the membrane is calculated as the integral over the force density in that region. Due to the particular mathematical form of the Helfrich energy, the total elastic force (applied by the membrane) can also equivalently be computed as a contour integral on the region boundary [35]. The concept of the equivalence of these two integrals is illustrated graphically in Fig. 3.5. Consider a fluid lipid bilayer membrane with forces applied as shown in the figure. The total external force $\vec{F}_1$ applied to the membrane can be equivalently calculated as an integral over an areal region $\mathcal{M}'$ or as an integral on the bounding contour Γ:

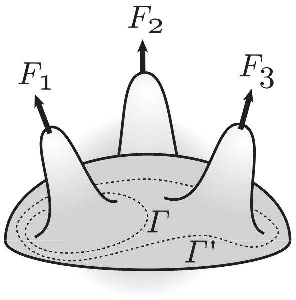


Figure 3.5: **Membrane with three external forces applied.** Dotted lines indicate contours Γ and Γ'. The external forces in these regions can be computed with line integrals. Figure by Paul Wiggins

$$\vec{F}_1 = -\int_{\mathcal{M}'} \mathrm{d}^2 s\, \sqrt{g}\; \vec{f}_{\text{int}} = \oint_{\Gamma} \mathrm{d}s\, \vec{\mathcal{F}}_{\text{int}}. \tag{3.4}$$

The total external force in the region $\mathcal{M}''$, bounded by contour Γ', can be computed similarly:

$$\vec{F}_1 + \vec{F}_3 = -\int_{\mathcal{M}''} \mathrm{d}^2 s\, \sqrt{g}\; \vec{f}_{\text{int}} = \oint_{\Gamma'} \mathrm{d}s\, \vec{\mathcal{F}}_{\text{int}}. \tag{3.5}$$

There are two advantages to calculating the external force as a contour integral rather than an integral over an areal region:

1. The internal force $\vec{\mathcal{F}}_{\text{int}}$ on the boundary has been integrated relative to $\vec{f}_{\text{int}}$ therefore it is one order lower in derivatives (third order instead of fourth order) making it more tractable numerically.

2. We may determine the external force without detailed knowledge of the structure of the membrane in the areal region; as long as the boundary has been well determined experimentally, we may measure the force on the region internal to that boundary.

With the necessary theoretical pieces now in place we now turn to the implementation of an experiment to test the predictions of the Helfrich theory outlined above.

3.3 Experimental materials and methods

3.3.1 Optical setup

The force measurements were performed using a custom apparatus comprised of two optical traps integrated with a fluorescence microscope. (See panel A of Fig. 3.6 for a schematic drawing of the vesicle force-extension apparatus.) Forces were applied to vesicles by binding streptavidin-coated 1 μm beads to DOPC giant unilamellar vesicles doped with biotin and TRITC. The vesicles were composed of 99% DOPC, 0.5% DOPE-biotin, and 0.5% DOPE-TRITC by molar fraction. The product numbers of the reagents used from Avanti Polar Lipids were: DOPC, 850375; DOPE-rhodamine, 810150; and DOPE-biotin, 870282. The beads used were Invitrogen catalog no. F-8777 FluoSpheres® NeutrAvidin®-labeled 1 μm nonfluorescent microspheres.

A 1064 nm trap was used as a force detector. The beam deflection was detected in the back focal plane by a position sensitive detector (PSD) [36]. A second 808 nm trap was employed to deform the vesicles. This second color trap was used to avoid crosstalk between the deflection and force detection beams at the PSD. The position of the 808 nm trap was varied using a beam steering telescope driven by a computer-controlled voltage actuator.

The radius of a typical vesicle was approximately 15 μm. In order to avoid membrane contact with the surface, experiments were performed at a distance of roughly 100 μm from both surfaces of the experimental chamber. To trap beads at these long working distances, it was necessary to employ a 60X water immersion objective. The trap stiffness was nearly independent of the working distance. The vesicles were imaged using TRITC fluoresence, excited by a mercury lamp. We used brightfield microscopy, lit by a white LED, to find and capture beads in the two optical traps ("bead fishing"). Vesicles were imaged using a Hamamatsu ORCA 285 camera.

The vesicle manipulations were performed in a two-input flow chamber. After blocking the chamber with BSA, the vesicle solution was pipetted into one of the input ports to fill the chamber. This input port was then sealed and the buffer was allowed to slowly evaporate from the unsealed input port, increasing the external osmotic pressure. When the vesicles became sufficiently soft, a charge of streptavidin-coated 1 μm beads was pipetted into the

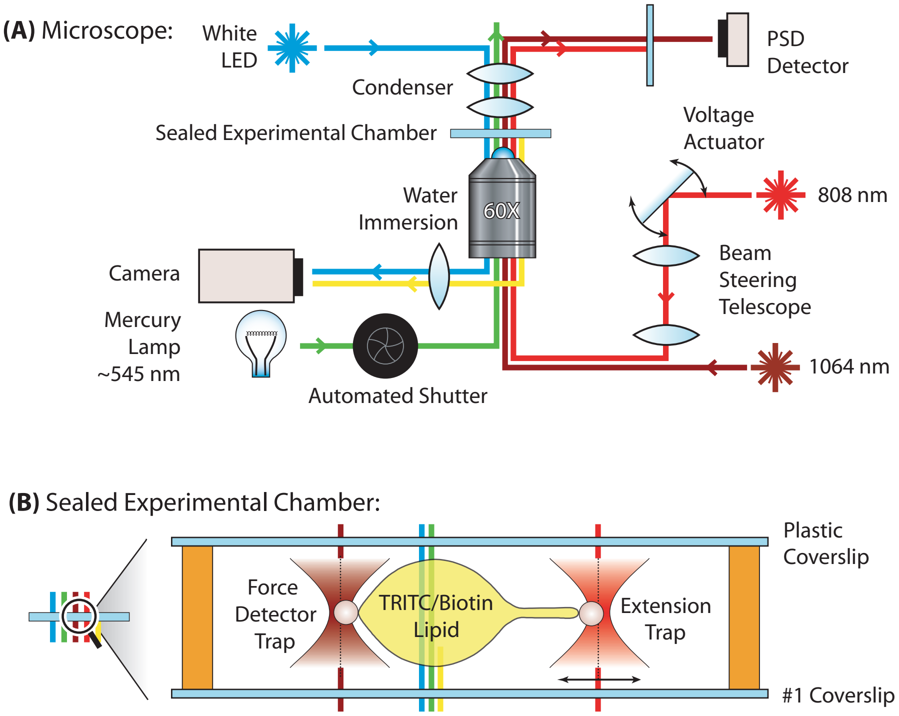


Figure 3.6: **Experimental apparatus schematic drawing and chamber detail. (A) Experimental apparatus schematic drawing.** The measurements were performed using a custom setup composed of two optical traps integrated with a fluorescence microcope. The beam deflection from the 1064 nm trap was detected in the back focal plane by a position sensitive detector (PSD). A second 808 nm trap with a beam-steering telescope was employed to deform the vesicles. **(B) Experimental chamber detail.** The experiments were performed in a sealed two-input flow chamber. Forces were applied to vesicles by binding streptavidin-coated 1 µm beads to DOPC giant unilamellar vesicles labeled with biotin and TRITC. Images of the fluorescently labeled lipid were analyzed to extract dynamic information about thermal membrane fluctuations as well as calculate the static elastic forces. Figure by Paul Wiggins

remaining input port and the port was sealed. The flow chamber is depicted schematically in panel B of Fig. 3.6.

3.3.2 Optical trap force measurement

Measurement of piconewton forces with optical traps is a well-studied subject, reviewed, e.g., in Neuman and Block [37]. A brief exposition of the method is also given here. The data that is directly read from the PSD is not position but voltage. In the limit of small deflections, one can apply Taylor's theorem to obtain a linear relation between voltage and position:

$$V(t) = EX(t), \tag{3.6}$$

where E has the units of an electric field.

We took voltage data from the PSD for each step in an extension series for a total of 10 seconds at a rate of 10 kHz, giving a time history of the movement of the trapped bead as a voltage time series $V(t)$. For the zero-force points, we took data for 10 seconds at 100 kHz. In terms of this voltage, the autocorrelation function is

$$\left\langle V(t)V(t')\right\rangle = C_V \exp(-\omega_0|t-t'|), \tag{3.7}$$

where C_V is the amplitude of the voltage autocorrelation function with corner frequency ω_0. Using the equipartition theorem, we have the following relation for the spatial autocorrelation function:

$$\frac{1}{2}\kappa\left\langle X(t)X(t')\right\rangle = \frac{1}{2}k_BT, \tag{3.8}$$

where κ is the effective spring constant of the optical trap. We may now use Eq. 3.6 to relate the voltage and spatial autocorrelation functions and obtain C_V as a function of known parameters:

$$C_V = \frac{k_BTE^2}{\kappa}. \tag{3.9}$$

We can therefore use the amplitude of the voltage autocorrelation function to determine E. The stiffness of the optical trap is a function of the corner frequency and the diffusion coefficient of the bead:

$$\kappa = \frac{\omega_0 k_BT}{D}. \tag{3.10}$$

In terms of experimental observables, the effective electric field E is

$$E = \sqrt{\frac{C_V \omega_0}{D}}. \tag{3.11}$$

We can write the force on the bead in terms of the voltage and E

$$F = -\kappa_V V = -\kappa X, \tag{3.12}$$

where the voltage spring constant is

$$\kappa_V = \frac{\kappa}{E} = k_B T \sqrt{\frac{\omega_0}{D C_V}}. \tag{3.13}$$

Assuming room temperature $T = 300\,\mathrm{K}$ and given values for D, ω_0 and C_V the trap stiffness κ_V is determined and the force exerted by the optical trap will be known. The diffusion constant D of beads in the sucrose solution used in this experiment was measured by tracking the diffusion of $1\,\mu\mathrm{m}$ beads, with no lipid present, in the sealed two-input flow chamber described in Section 3.3.1. A custom software package, kindly provided by Robert Bao, was used to analyze the time-series images of bead diffusion; the diffusion constant for the bead solution was measured to be $0.44\,\mu\mathrm{m}^2/\mathrm{s}$. The remaining parameters were determined by fitting the observed voltage autocorrelation to the exponential function given in Eq. 3.7, giving $\omega_0 = 6100\,\mathrm{s}^{-1}$ and $C_V = 0.0013\,\mathrm{V}^2$. Thus the trap stiffness was determined to be $13\,\mathrm{pN/V}$.

In our experiment, forces were applied via optically trapped beads to DOPC giant unilamellar vesicles. The streptavidin-coated beads were linked to the vesicles via biotin-labeled lipids. The bending modulus k_C for DOPC is $0.85 \times 10^{-19}\,\mathrm{J}$ [30].

In a typical experiment, fifty image frames were captured at each step in an extension series. An extension series consisted of about fifty steps, split between outward and inward movement, with a step size of roughly $1.5\,\mu\mathrm{m}$. In the next section we describe the procedure for tracing the shape of the vesicle from the captured images and the method used to estimate the unknown parameters p, α, and C.

3.4 Computational methods

In order to proceed with the calculation of the force applied to a membrane by analyzing its shape we must first determine its conformation. The vesicle conformation is represented as a cubic spline with control vertices traced from the images of the fluorescent membrane. A combination of tracing by hand together with fitting to an empirical model for the intensity profile of a vesicle was used to generate spline control vertices. At the close of this section we describe the Proximal Equilibrium Approximation which was used to determine the pressure, tension, and spontaneous curvature of the membrane, and discuss how the error on these parameters can be estimated.

3.4.1 Spline representation

We implemented the vesicle spline representation with the `MATLAB` command `interp1` and the `v5cubic` option. This interpolation scheme was chosen because it was convenient to implement in `MATLAB`. Using the `v5cubic` spline affords a number of desirable features:

1. The cubic splines were constructed with C^2 smoothness, i.e., continuous up to the second derivative. This property was necessary in order to produce well-defined derivatives of the curvature when computing forces.

2. The interpolation method implemented by the `v5cubic` option makes use of basis functions with modest support. The interpolated curve over each segment depends on the positions of the two nearest neighboring points on each side. Other spline options in `MATLAB` with larger or smaller supports produced curves that differed more noticeably from the vesicle contours.

3.4.2 Vesicle tracing

In Fig. 3.7 we show an image of a vesicle together with the control vertices traced from that image. The position of the vertices in the direction parallel to membrane surface was chosen by hand; the position in the direction normal to the surface was fit to the fluorescence profile independently in each frame. We attempted a number of automated procedures for choosing the parallel positions of the control points but these algorithms were difficult to implement in a robust manner. In this analysis, we assume that the vesicle is axially symmetric so that the observed cross-sectional structure determines the entire vesicle shape. This approximation is excellent at large extensions, but breaks down at low force.

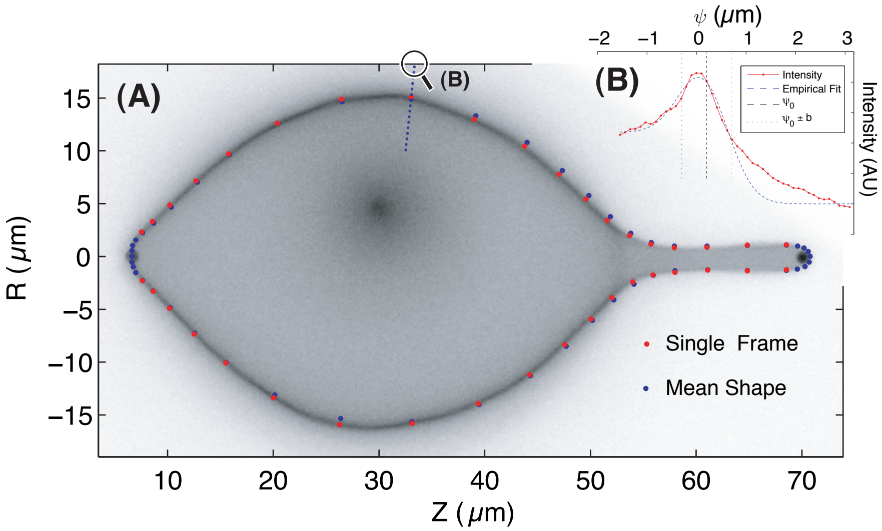


Figure 3.7: **Capturing the membrane conformation. (A) Fluorescence image of a vesicle from a force-extension series.** The vesicle conformation is represented as a cubic spline. The positions of the control vertices are fit from the fluorescence profile. The red vertices were traced from the fluorescence image shown in the figure; the blue vertices correspond to a trace of the mean of the fifty images at this step in the extension series. The localized fluorescence in the body of the vesicle is the result of an internal vesicle. The increase in external osmotic pressure often leads to a budding transition. Note also that the beads are not fluorescent themselves, they appear fluorescent due to the aggregation of lipid there. **(B) Fitting control point positions.** The positions of the control points normal to the contour were fit to the fluorescence profile (red curve), employing an empirical model (blue curve). The intensity values shown here were taken from a one dimensional slice of the image, shown as a blue dotted line in panel A.

The positions of the control points in the direction perpendicular to the membrane contour were fit to the fluorescence profile, employing an empirical model. A good model for the fluorescence profile was found to be a convolution of a structure function with a Gaussian point spread function (PSF):

$$I_j(\rho;\rho_0) = \Psi_{\mathrm{PSF}} \otimes I_0(\rho;\rho_0), \tag{3.14}$$

where I is the intensity of the grayscale fluorescence image for vertex j at position ρ, defined as the distance to the vesicle symmetry axis in a direction perpendicular to the membrane surface. ρ_0 is the value of ρ at the vertex j, and I_0 and Ψ_{PSF} are the structure function and Gaussian point spread function, respectively. We interpolated to find the intensity of the fluorescence image along a ray perpendicular to the membrane and then fit this intensity profile to the right half of the empirical function in Eq. 3.14 and depicted in Fig. 3.8. The structure function is

$$I_0(\rho;\rho_0) = A\delta(\rho-\rho_0) + A\delta(\rho+\rho_0) + B\Theta_H(\rho+\rho_0)\Theta_H(\rho_0-\rho), \tag{3.15}$$

where A and B are constants, δ is the Dirac delta function, and

$$\Theta_H(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0, \end{cases} \tag{3.16}$$

is the Heaviside step function. The Gaussian point spread function is defined to be

$$\Psi_{\mathrm{PSF}}(x) = \exp(-x^2/2b^2), \tag{3.17}$$

where b is the point spread width. The structure function I_0 and convolved empirical function are shown in Fig. 3.8. We have idealized the in-plane membrane as an infinitely thin surface, normal to the plane of focus, with a delta function source of fluorescence. The out-of-focus membrane is represented as a region of uniform intensity (the Heaviside function). A number of more elaborate (non-empirical) models were implemented, but this model led to the best approximation of the observed fluorescence profile. The fit to a typical fluorescence profile is shown in panel B of Fig. 3.7.

Mathematically, the force measurement should not depend on the vesicle representation (the parallel positions or number of the spline control vertices). However human, experimental, and numerical error could, in principle, make the conformational force measurement sensitive to these factors. We found however that the conformational force is independent of a particular choice of control vertices, as shown in Fig. 3.9. The close agreement of these two independent sets of tracings from the same vesicle extension series show that the conformational force does not depend sensitively on the position or number of the spline control points, as asserted.

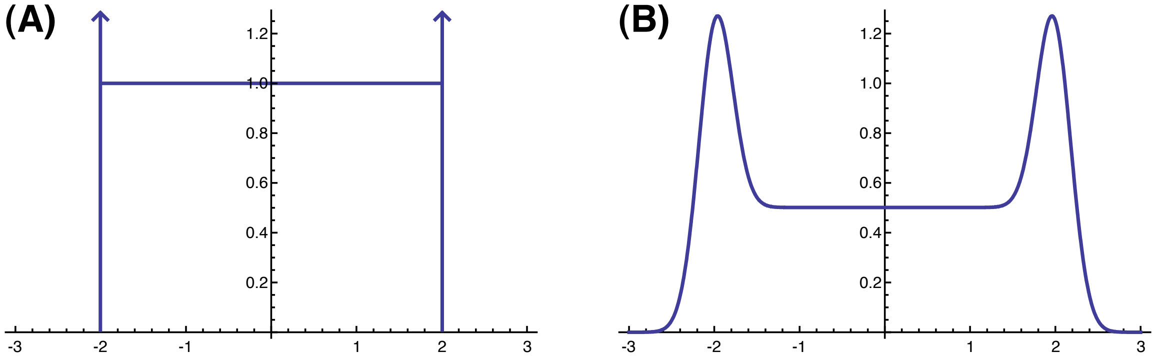


Figure 3.8: **Empirical model used to fit vertex normal positions. (A) Structure function $\mathbf{I_0}$.** Panel A on the left depicts the structure function, which consists of a delta function of intensity at the edge of the membrane with a uniform fluorescence source on the interior. **(B) The empirical fitting function.** Panel B shows the empirical fitting function, a convolution of the structure function I_0 and a Gaussian point spread function. The parameters used were $A = B = 1$, $b = 0.2$, and $\rho_0 = 2$.

3.4.3 Proximal Equilibrium Approximation

Analysis of the fifty image frames at each step in the extension and retraction series yielded a traced contour representing the axisymmetric shape of the vesicle. After determining the membrane structures, the unknown experimental parameters, pressure p, tension α, and spontaneous curvature C must still be determined. The condition of (approximate) mechanical equilibrium was used to calculate these three quantities.

In mechanics, there are at least three ways of expressing mechanical equilibrium; at every point on the membrane:

1. The energy associated with normal displacement of the membrane is minimized.

2. The total force is zero.

3. The normal displacement to the equilibrium position is zero.

Mathematically, all three of these statements are exactly equivalent. Therefore, for conformations that are dominated by deterministic mechanical forces rather than fluctuations, all three formulations of equilibrium should be equivalent.

Although these approaches for determining proximity to equilibrium appear equivalent, we argue that, from an experimental standpoint, using the normal displacement to the equilibrium position as the objective function (hereafter the "distance objective function")

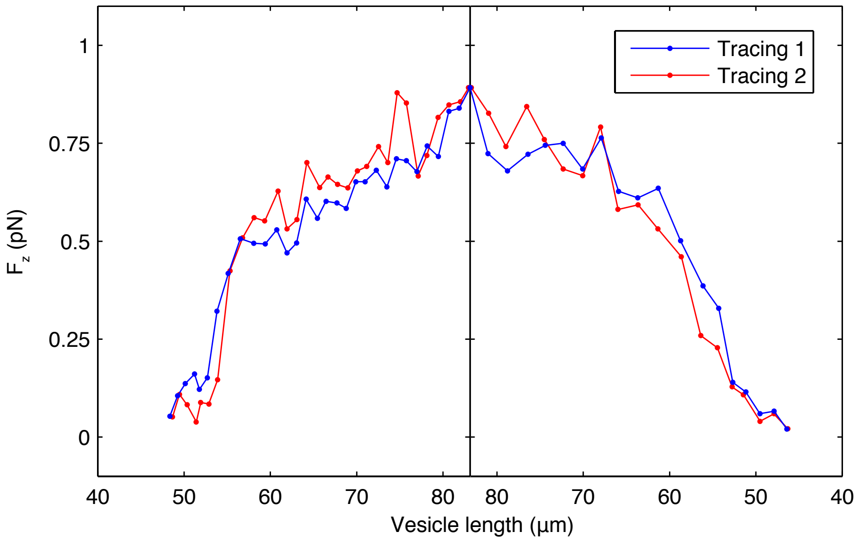


Figure 3.9: **The conformational force is independent of the tracing representation.** The figure shows two independent tracings of the vesicle extension series. It is clear that the two tracings are in quantitative agreement with one another, confirming that our results are not sensitive to a particular choice of spline control points.

is the most powerful. The first advantage of the distance objective function is that it is the position of vertices that is determined from the experimental fluorescence images, not the force or energy at each vertex. It is therefore most natural to describe the experiment in terms of the the observed quantities: the positions of the control vertices. But most importantly, the distance objective function is least susceptible to high frequency noise and gives the most robust results in the numerical minimization. For configurations that are close to equilibrium, the distance to equilibrium ψ can be computed by multiplying the vertex forces by the inverse stiffness matrix:

$$\psi = -K^{-1} f_{\perp}. \tag{3.18}$$

The stiffness matrix K is obtained by performing the second variation of Eq. 3.2 for the Helfrich energy and was derived in detail by Zhong-can and Helfrich [38].

With the displacement of each vertex to equilibrium chosen as the objective function, we are ready to apply the Proximal Equilibrium Approximation, as follows:

1. The observed membrane shape is represented as a cubic spline with an associated set

of vertices serving as control points. We fix the vertices near the points of contact with the trapped beads, where forces are applied and the membrane cannot be resolved with confidence.

2. We choose trial values for the unknown parameters of pressure p, tension α, and spontaneous curvature C.

3. We compute the vertex normal forces f_i and stiffness matrix K with the trial values of the parameters.

4. We calculate the estimated equilibrium displacement ψ_i at each vertex i with Eq. 3.18.

5. We minimize the sum of the squares of the equilibrium displacements for all non-fixed vertices with respect to the parameters p, α, and C using the `MATLAB` function `lsqnonlin`.

Particularly for conformations close to the transition, when a membrane tether forms it was necessary to carefully choose the initial values of the parameters for pressure, tension, and spontaneous curvature or the minimization would fail to find a satisfactory solution. Conformations not close to the transition were less sensitive to the initial values of the parameters and could be fit robustly without hand-tuning the initial parameter values. The Proximal Equilibrium Approximation is illustrated visually in Fig. 3.10 for the vesicle shown earlier in Fig. 3.7. The optimized parameters found were pressure: $3.4 \pm 0.2\,\mathrm{mPa}$, tension: $7.4 \pm 0.5\,\mathrm{k_BT/\mu m^2}$, and spontaneous curvature: $-0.27 \pm 0.10\,\mathrm{\mu m^{-1}}$.

One of the interesting outcomes of our experiments is the necessity for the spontaneous curvature parameter C. Intuitively one might suppose this parameter to be unnecessary since there is no asymmetry in the leaflets of the two membranes of the pure DOPC giant unilamellar vesicles we used in the experiments. Nevertheless, due to the finite thickness of the membrane there is an area difference between the two leaflets and in pulling a tether out of the GUV the effects of this difference are noticeable. In Fig. 3.11 we show how the parameter C is determined by the Proximal Equilibrium Approximation procedure and demonstrate its necessity in describing the conformation of the membrane.

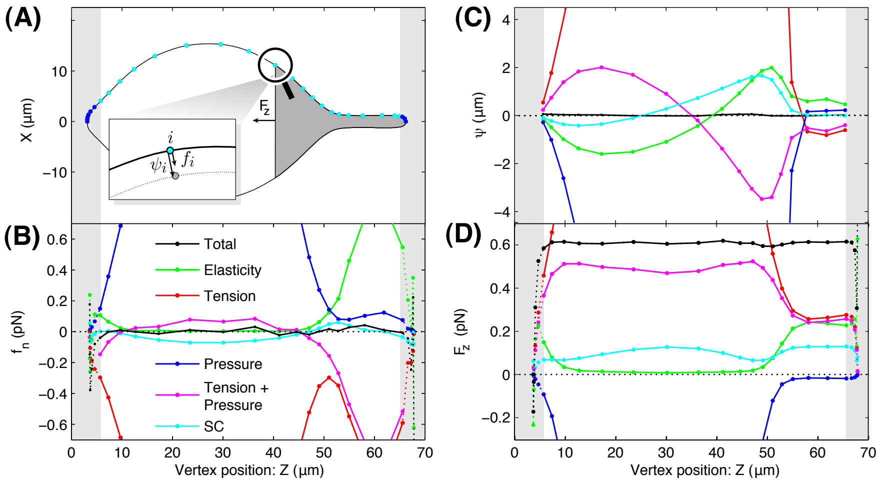


Figure 3.10: **The Proximal Equilibrium Approximation. (A) Vesicle conformation.** The axisymmetric vesicle conformation is represented as a cubic spline with control vertices shown as points along the contour. In order to find the pressure, tension, and spontaneous curvature of the membrane, we choose initial values of these parameters which produce a trial membrane configuration. On the right we show a magnified view of the observed membrane shape (solid curve) as well as the trial shape (dotted line) near control vertex i where the displacement between the vertex in the observed shape (light blue) and the trial shape (light gray) is labeled as ψ_i. We minimize the mean square displacements ψ_i with respect to the trial values of pressure, tension, and spontaneous curvature in order to form a best estimate for the values of these parameters. During the minimization we exclude the regions in the proximity of the bead attachment points, shaded in gray on the left and right, where the membrane shape cannot be resolved with sufficient precision to make reliable force estimates. **(B) Vertex force.** The vertex force f_i is the sum of components from membrane bending elasticity, tension, pressure, and spontaneous curvature. In the body of the vesicle, force balance is dominated by tension (red curve) acting to contract the vesicle radius, and pressure (blue curve) acting to expand the radius. In the vesicle tether, force balance is dominated by the competition between elasticity (green curve) acting to expand the tether radius, and tension (red curve) acting to contract the tether radius. The induced spontaneous curvature has its greatest effect in the neck of the vesicle where there is a transition between the two regions. **(C) Estimated displacement to equilibrium.** The distance to equilibrium along the normal to the membrane surface is shown in panel C. The optimal values for the parameters p, α, and C are found by minimizing the sum of squares of the vertex equilibrium displacements. **(D) Summed force:** The summed force at vertex i, F_z^i, is the sum of the z components of the vertex forces in the dark gray region which excludes vertices with $z < z_i$. Therefore F_z^i is the total force applied by the right-hand side of the vesicle in the z direction. In this panel we plot the total summed force (black) which, as expected, is approximately constant throughout the body of the vesicle, since the forces are applied only at the poles of the vesicle. The summed force is also shown decomposed into individual contributions from pressure, tension, spontaneous curvature, and bending elasticity.

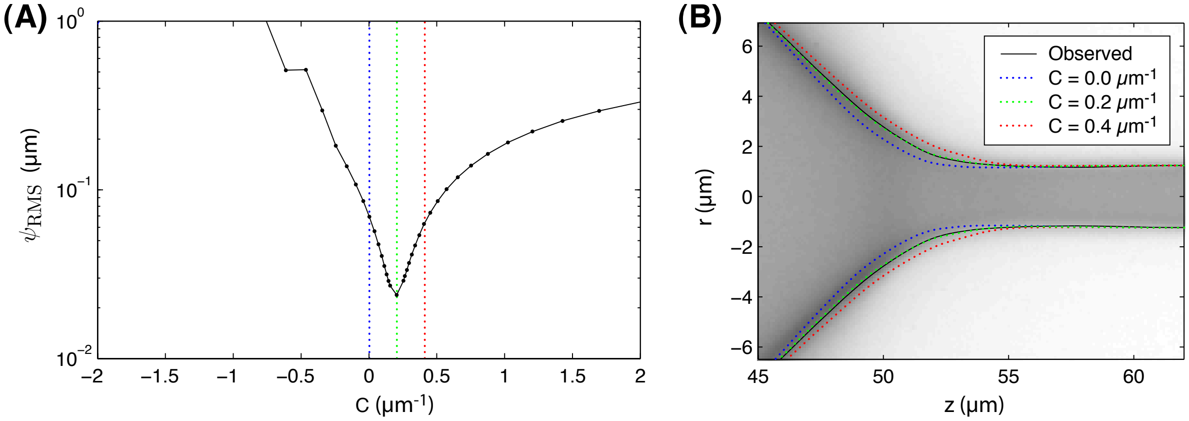


Figure 3.11: **The determination of C.** The unknown parameters are determined by the Proximal Equilibrium Approximation: the sum of the squared equilibrium displacements ψ_i are minimized with respect to a three-dimensional parameter space corresponding to (p, α, C). **Panel A.** The plot shows the $\sum_i \psi_i^2$ landscape along the direction of smallest curvature, parameterized by C. **Panel B.** The estimated equilibrium shape in the neck region for three C values corresponding to the vertical lines in Panel A is depicted. The green curve, corresponding to $C = 0.2\,\mu\mathrm{m}^{-1}$ (close to the optimal C value), qualitatively matches the observed membrane conformation (black).

3.4.4 Calculation of curvature and forces

We computed the curvatures from the spline interpolation by finite difference and then numerically integrated to calculate the membrane bending energy. Vertex normal forces were calculated using a centered finite difference technique: we deformed the spline contour by perturbing vertices by a small amount $\Delta N/2$ outward and inward in the direction of the surface normal, then took the difference of the outward and inward perturbed energies: $f_\perp = \Delta E/\Delta N$. The integrated force was computed by a forward finite difference, moving vertices in a region of the membrane by a slight amount Δz in the axial direction and computing the change in the energy ΔE, giving us a force $F_z = \Delta E/\Delta z$.

3.4.5 Error computation for the conformational force

The error in the force computed from the conformation of the membrane was estimated in two ways. One method consists of tracing each of the 50 images at each step in the extension series, applying the Proximal Equilibrium Approximation, and then analyzing the resulting distribution of force, pressure, tension, and spontaneous curvature measurements. This is the method used to generate the r.m.s. errors on the force shown in Section 3.5, Fig. 3.13.

Another method consists of calculating confidence intervals for the unknown parameters (p,α,C) and then deriving the error in the force based on those uncertainties. This can be done by standard methods; see, e.g., Bevington and Robinson [39], whose analysis we follow here. The Proximal Equilibrium Approximation amounts to a non-linear least-squares fit of the membrane contour given values for the pressure, tension, and spontaneous curvature to the observed membrane contour. The χ^2 goodness-of-fit parameter in the Proximal Equilibrium Approximation is:

$$\chi^2 = \sum_i \frac{1}{\sigma_i^2} \left[y_i - y_{\text{obs}}\right]^2, \tag{3.19}$$

where σ_i is the uncertainty in the position of vertex i in the direction normal to the membrane contour, y_{obs} is the observed position of the vertex in the normal direction, and y_i is the position of the ith vertex given parameter values p, α, and C. The uncertainty in the normal positions, σ_i, was essentially constant for each of the vertices in a given image from our experiments. In the neighborhood of the minimum χ^2 value found by the Proximal Equilibrium Approximation, we may use a Taylor expansion of our fitting function, y_i, to find:

$$\Delta a_j = \epsilon_{jk}\beta_k \tag{3.20}$$

where Δa_j is the uncertainty on the jth parameter, ϵ_{jk} is the error matrix, and β_k is the derivative of χ^2 with respect to the kth parameter. This was the technique used in generating the errors shown as dotted lines on the plots of pressure, tension, and spontaneous curvature in Fig. 3.14 below. More generally, after finding the optimal point in χ^2 space, we may calculate the χ^2 values in the neighborhood of this point and find contours in parameter space where χ^2 increases by a given amount to find arbitrary confidence intervals on the parameters. We found that using the χ^2 method of estimating confidence intervals on the parameters predicted larger errors than the r.m.s. method from the 50 images at each step in the experiment.

Applicability of the Proximal Equilibrium Approximation

In order to apply the Proximal Equilibirum Approximation, the conformation must be close to mechanical equilibrium rather than fluctuation dominated. The effect of thermal fluctuations as a source of error in the Proximal Equilibrium Approximation can be tested directly: we generated membrane tracings from both individual image frames, as well as from the mean of the fifty images at each step in the extension series. The analysis of traces from individual image frames results in a distribution of force measurements whose mean can be taken as the value of the force at that step in the experiment. The mean force from the 50 individual tracings varied by an average of 0.07 pN from the value of the force calculated from the traced average image for the results shown in Section 3.5, Fig. 3.13, and is small in comparison with other sources of error in the experiment.

A second limitation of the Proximal Equilibrium Approximation is the approximate degeneracy of some membrane conformations with respect to the estimated parameters p, α, and C. Perhaps the most obvious example is the sphere. The Proximal Equilibrium Approximation essentially solves for the unknown values of pressure, tension, and spontaneous curvature by balancing the constraint forces due to those parameters against the elastic force. In the case of a sphere, there are no elastic forces to balance. Since the elastic forces vanish, there is a two-dimensional continuum of pressure and tension values that will satisfy the force balance equations. For such a degenerate case, the Proximal Equilibrium Approximation cannot yield meaningful results and the numerical optimization will fail to find a good local minimum. Obviously, the sphere is a special case, but shapes near the transition point when a membrane tether forms appear to have only a weak dependence on the spontaneous curvature, and this near degeneracy makes it difficult for the numerical optimization to converge. Uncertainties in the spontaneous curvature dominate the error in conformation force computation, as can be seen in Fig. 3.14.

Finally, the Proximal Equilibrium Approximation is limited by the necessity of guessing and excluding points where external forces are applied. While the location of the applied forces is obvious in the analysis of deformed vesicles, for biological examples, the position of the applied forces may not be known. Indeed, in many examples, the contacts may be so frequent as to not permit large, force-free regions to be identified and analyzed. On the other hand, in biological applications the pressure can often be neglected and the tension and spontaneous curvature may be regulated by the cell or determined using other techniques [40]. If these parameters can be neglected or are otherwise known or determined, it is straightforward to compute membrane forces without applying the Proximal Equilibrium Approximation.

3.5 Results

Force-induced membrane tether formation [41, 42] provides a convenient test of the conformational force computation: a known force, applied to an optically trapped bead, can be directly compared to the conformational force. The progression of vesicle shapes in our experiment as the tether forms is illustrated by the membrane conformations shown in Fig. 3.12. As the axial force is increased, the vesicle first assumes an elongated shape before reaching a transition in which a cylindrical tether forms at one end of the vesicle. This transition is seen in reverse in the retraction phase of the experiment. In order to resolve the structure of vesicles with a membrane tether by light microscopy, the force at which tethers formed had to be less than 2 pN, implying that the undeformed vesicles had a large excess of area, i.e., the vesicles were in a low tension regime.

In Fig. 3.13, we compare the conformational forces to the applied force for the tether extension series shown in Fig. 3.12. We measure a mean relative error of 10% for forces in the range of 0.2 to 1 pN, showing that the applied and conformational force are in both qualitative and quantitative agreement in that range. The optical tweezing force measurements have two important shortcomings in the subpiconewton regime:

1. Subpiconewton force measurements are generically susceptible to drift.

2. The vesicles themselves were also found to weakly scatter the trapping beam, resulting in anomalous force signals during large-scale changes in vesicle structure. For instance, only data sets where the force sensor bead was not positioned on the end of the tether were found to result in consistent force traces. This effect is documented further in Appendix B.1.

The mismatch between conformational and applied forces appears to increase towards the end of the retraction, consistent with both trap drift and vesicle-induced beam scattering. Our analysis also revealed limitations in the conformational force technique. At forces

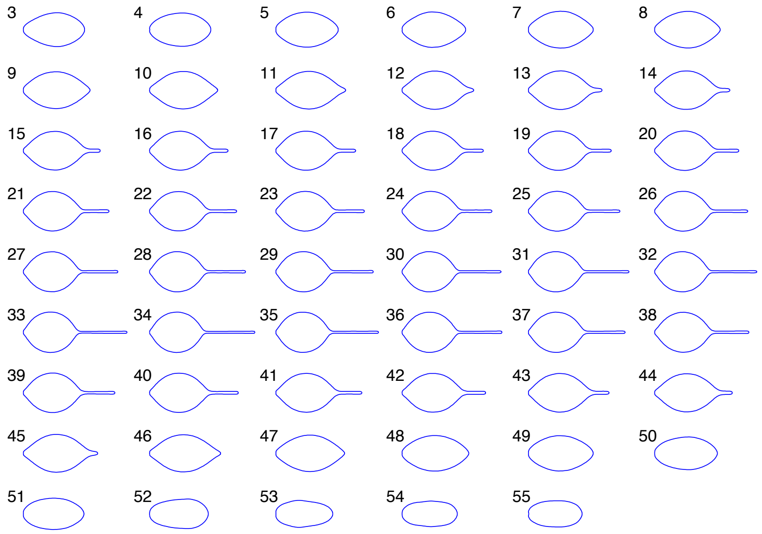


Figure 3.12: **Vesicle conformations.** In the figure above we show the vesicle conformations from an experimental extension series labeled by the corresponding step number. As force is applied to either end of the vesicle it first assumes an elogated shape before reaching a transition where a cylindrical tether is formed at one end. The maximal extension of 83.1 µm is reached at step number 34 in this series.

lower than about 0.2 pN, the conformational force is limited by the determination of the unknown parameters: pressure, tension, and spontaneous curvature. At high force, the conformational force is limited by diffraction as the typical length scale of the neck of the tether shrinks below that limit.

3.5.1 Results of the Proximal Equilibrium Method

As described in Section 3.4.3, the unknown parameters p, α, and C for each configuration are determined by applying the Proximal Equilibrium Approximation. We found that the values of all three of these parameters evolved with the vesicle conformation. These results are shown in Fig. 3.14 where these parameters are plotted as a function of the step number in the extension series. In Fig. 3.15, we plot the effective constitutive relations for the vesicle.

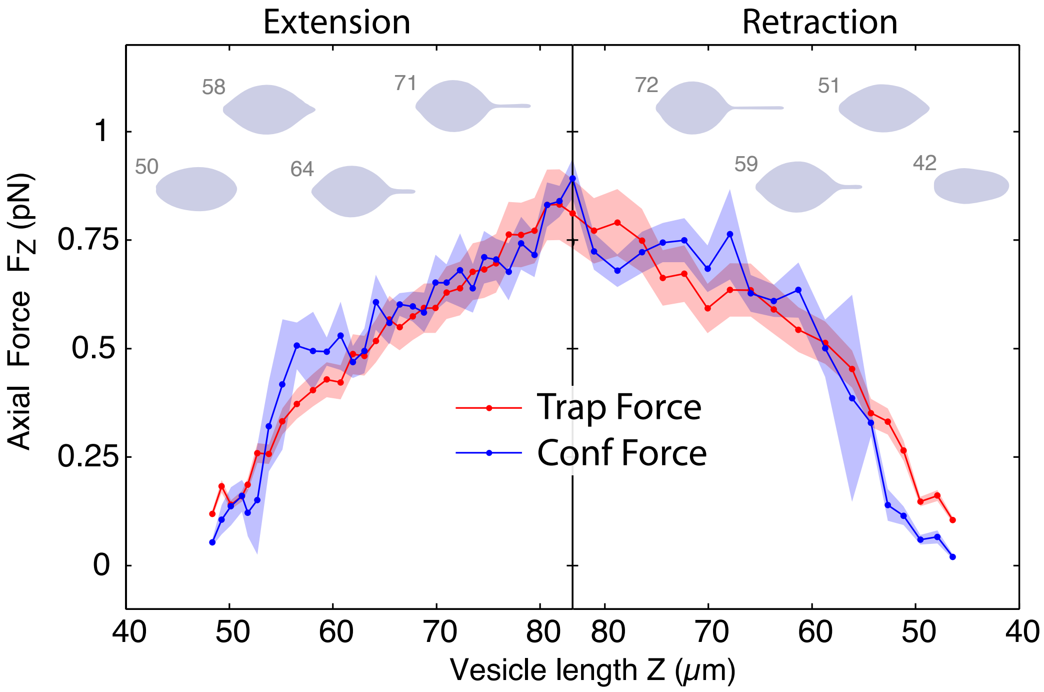


Figure 3.13: **Comparison of the applied and conformational force.** The computed conformational force (blue curve) and the applied trapping force (red curve) are in both qualitative and quantitative agreement from 0.2 to 1 pN. The red and blue shaded regions are the error in the trapping force and the r.m.s. variation of the conformational force, respectively. Note that the x-axis orientation has been inverted in the retraction data. **Inset above:** Membrane conformations as a function of axial length (μm).

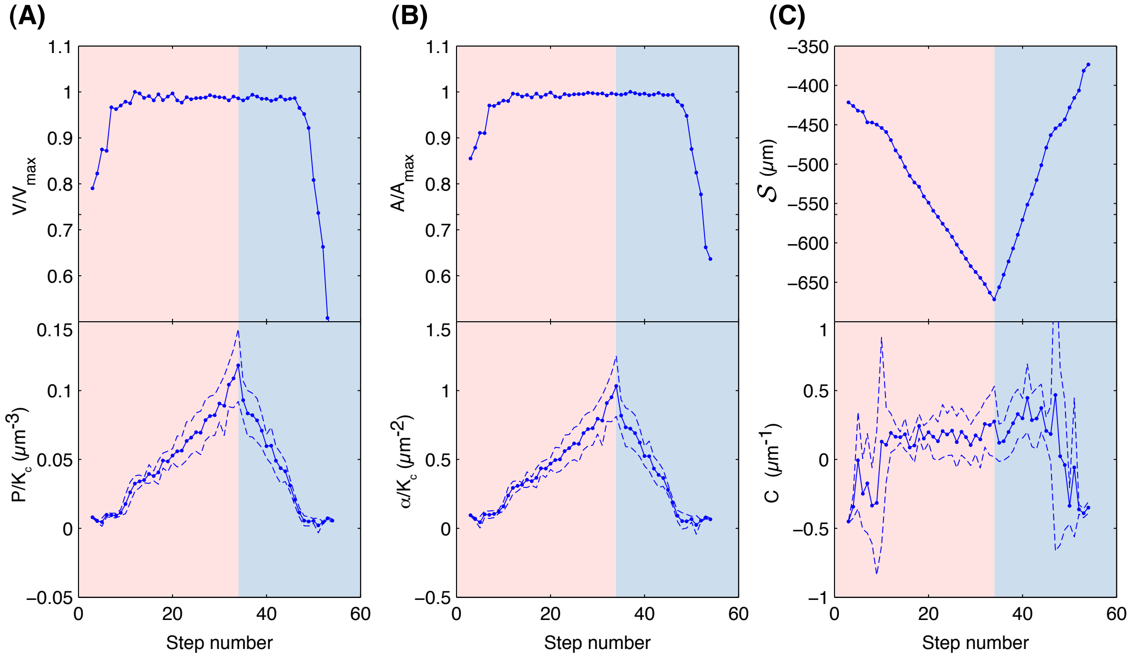


Figure 3.14: **Tension, pressure and spontaneous curvature of a vesicle with a membrane tether.** In this figure we plot the estimated values of pressure p, tension α, and spontaneous curvature C (blue solid lines) as a function of extension number for the extension (pink region) and contraction (blue region) of the vesicle featured in Fig. 3.12. These values and their r.m.s variation (dashed lines) were obtained by application of the Proximal Equilibrium Approximation method discussed in Section 3.4.3. The volume V, area A and integrated curvature $\mathcal{S}$ are also shown. **Panels A and B.** In the top panes of panels A and B, the apparent volume and area of the mean conformation are plotted vs. step number in the extension series. The true volume and area of the membrane are fixed, therefore the change in apparent volume and area upon extension and contraction is the result of volume and area hidden in fluctuations from the mean axisymmetric conformation. In the lower panes of panels A and B, the pressure and tension are seen to rise steadily with tether extension and are peaked, as expected, at the maximum extension. **Panel C.** The integrated curvature and spontaneous curvature as a function of step number in the extension series. The spontaneous curvature is predicted to be a linear function of the integrated curvature by the ADE theory. This linear relationship is not observed and some possible explanations for this discrepancy are discussed in Section 3.5.1.

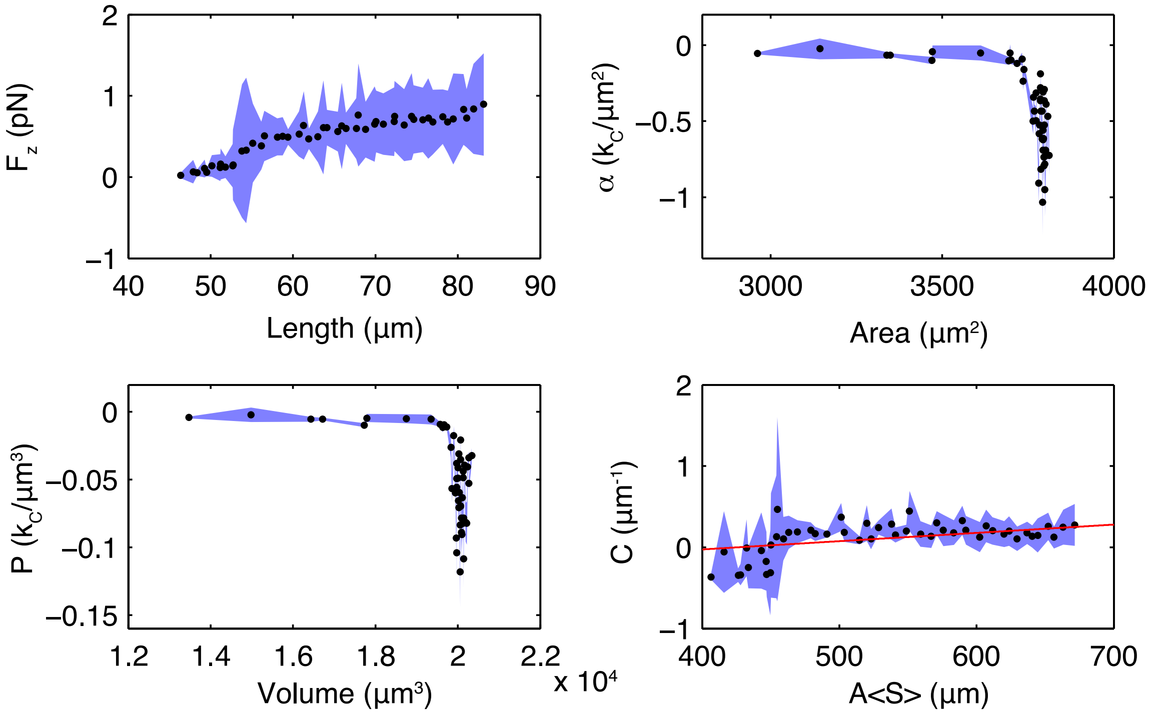


Figure 3.15: **Effective constitutive relations for a vesicle with a membrane tether.** In this figure we plot force vs. length, tension vs. area, pressure vs. volume, and spontaneous curvature vs. integrated curvature for the vesicle shown in Fig. 3.12. The black points indicate measured values and the blue regions show the corresponding errors in the ordinate variable. We note that the error in the right-most points on the the tension and pressure plots, which appear here to be small and approximately constant, are actually dominated by the errors in area and volume, respectively, which are not shown. The predicted theoretical curve according to Eq. 3.21 (which did not require a fit) is shown in red in the plot on the lower right.

Pressure and tension

At the beginning of the extension (the end of the retraction) in Fig. 3.14, there is a steep rise (drop) in the apparent volume and area of the membrane, seemingly contradicting the claim that the vesicle volume and area are incompressible. This is due to the fact that although the actual volume and area of the membrane are in fact nearly incompressible, some volume and area are hidden in the thermal membrane undulations. In Fig. 3.15, the fluctuation-dominated regime is clearly visible, corresponding to the (nearly) constant tension and pressure regimes. In the high-tension limit, little membrane remains in membrane undulations and the tension and pressure can be interpreted as constraint forces. This regime is characterized by the constant area and volume at high tension as shown in Fig. 3.15.

Spontaneous curvature

Area-difference elasticity theory predicts the dependence of spontaneous curvature on membrane deformation [26, 43, 44]:

$$C = -K_{\mathrm{ADE}} \left(\langle S \rangle - S_0 \right), \tag{3.21}$$

where K_{ADE} is a non-local bending modulus and

$$\langle S \rangle \equiv A^{-1} \mathcal{S} = A^{-1} \int \mathrm{d}A \, S, \tag{3.22}$$

is the integrated sum of the principal curvatures $\langle S \rangle$ over the deformed vesicle and S_0 is a constant. If the membrane is initially equilibrated by the slow processes of lipids flipping between leaflets, we would expect S_0 to be zero for the undeformed vesicle. In Fig. 3.15, this simple linear dependence on the mean summed curvature is not observed. Following are a few items to bear in mind when considering the lack of support in the data for a linear relationship between the spontaneous curvature and the integrated mean curvature:

1. The relative error in determining the spontaneous curvature C is larger than for the other two parameters. The optimization landscape for the distance objective function discussed in Section 3.4.3 is the most “flat” as C is varied. The higher error in C makes it difficult to clearly discern a linear relationship.

2. At low force, the effects of themal undulations become important and entropic effects dominate the results. Since we cannot observe the curvature (or area, or volume) stored in fluctuations in this regime we cannot make a good comparison with the theoretical prediction in the low-force regime.

3. Finally, the slow process of lipids flipping between the leaflets of the bilayer may also have an effect on our results. ADE theory assumes a fixed lipid number in each leaflet and so to the extent that lipids are able to equilibrate between the two leaflets, this condition is violated. The aggregation of lipid on the optically trapped beads, as well as the budding processes observed during the experiments, may all indicate sources of deviation from the assumption of fixed lipid number.

3.5.2 More vesicle experiments

In this section we show six additional force measurements from other experiments that were not analyzed in the full detail of the series shown in Fig. 3.13. In each plot, the red curve is the force measured from the optical trap and the blue curve represents the force measured from the conformation of the membrane using the Proximal Equilibrium Approximation to estimate the unknown parameters p, α, and C.

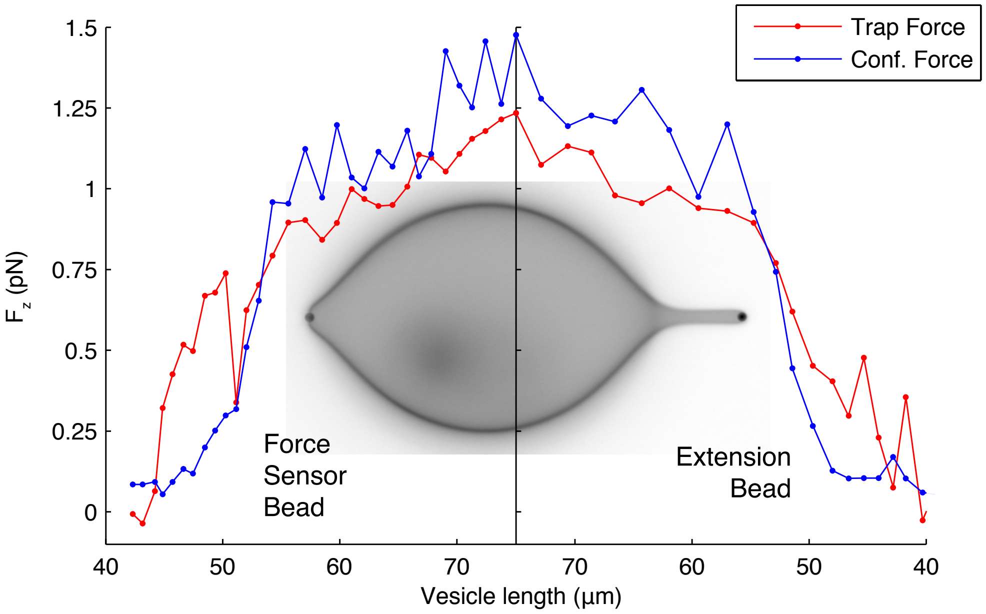


Figure 3.16: **Data set 2.** This extension series was executed immediately following the series shown in Fig. 3.12 on the same vesicle shown there. Anomalous forces due to light scattered by the membrane itself are apparent in the first few steps of this extension in the trap force; see Appendix B.1 for more discussion of this effect.

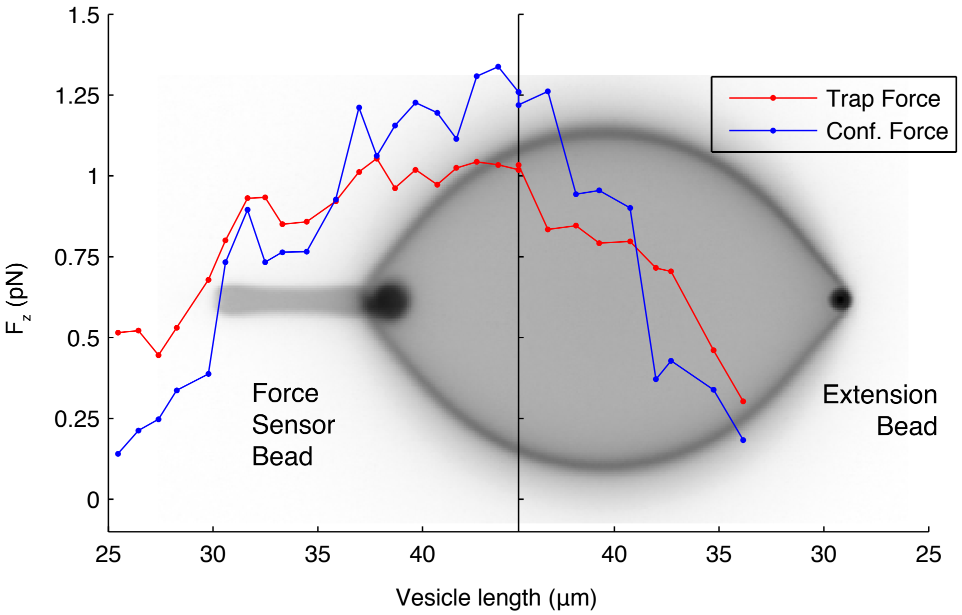


Figure 3.17: **Data set 3.** Again, anomalous forces due to light scattered by the membrane itself are apparent in the trap force. We found that data sets where the membrane tether formed on the extension side resulted in the best quantitative agreement with the conformational force. See Appendix B.1 for more details.

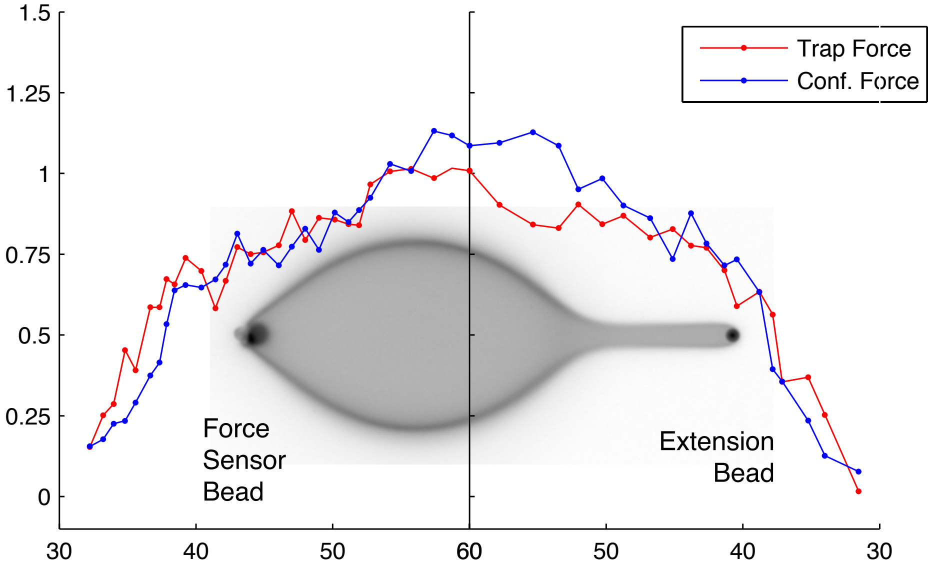


Figure 3.18: **Data set 4.** This extension series was conducted immediately following the series shown in Fig. 3.17 and shows excellent agreement between the trap and conformational force.

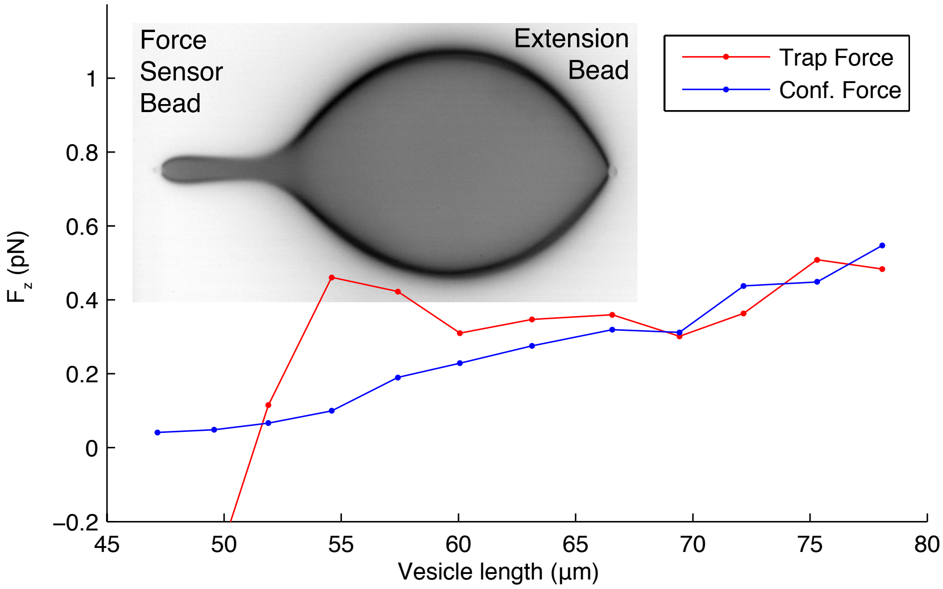


Figure 3.19: **Data set 5.** This experiment shows an extreme example of the anomalous trapping forces discussed in Appendix B.1 induced by large-scale membrane conformational changes at the force sensor bead.

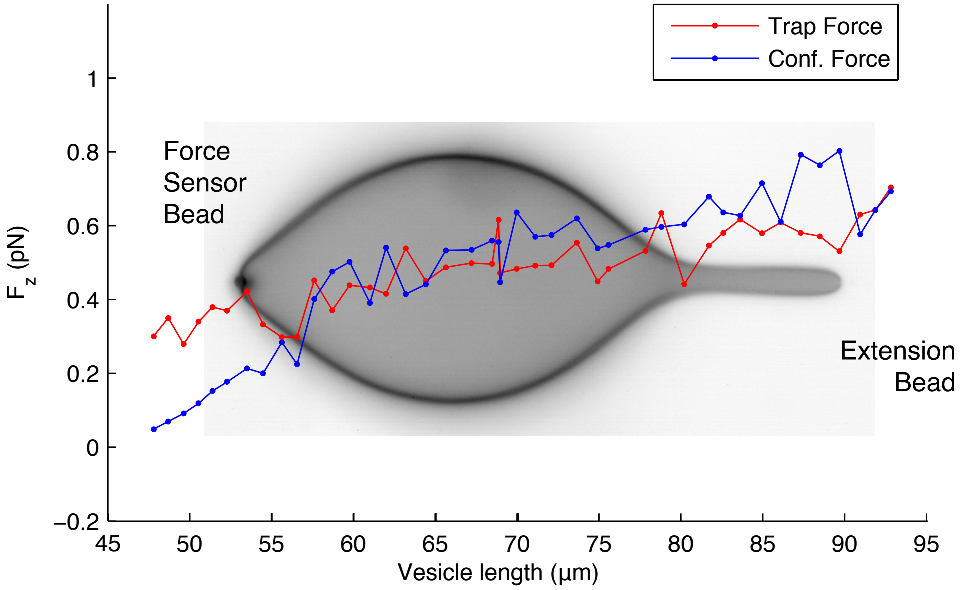


Figure 3.20: **Data set 6.** In this experiment we again observe excellent agreement between the trap and conformational force for higher forces.

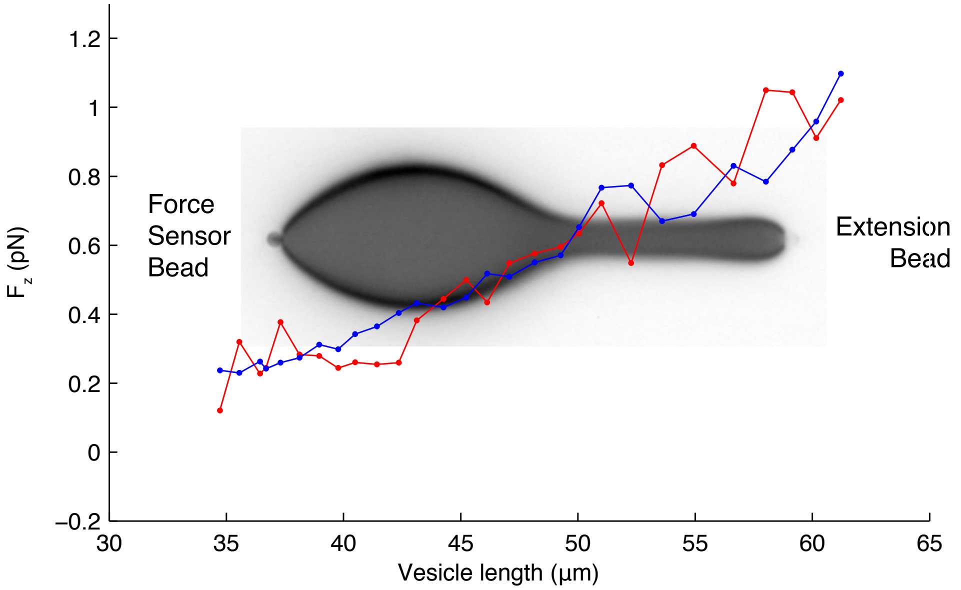


Figure 3.21: **Data set 7.** This final set of experimental data also demonstrates good agreement between the trap and conformational force.

3.6 Discussion

Our goal with this chapter was to develop a general method for calculating the forces applied to arbitrary membrane conformations observed by new microscopy techniques. We have validated the model with the axisymmetric shapes derived from an *in vitro* optical tweezing experiment pulling membrane tethers from GUVs, a well-studied system in the membrane mechanics literature [43, 45–47]. Nothing about the method however limits it to such symmetric membrane shapes and the full power of the conformational force technique will be unleashed when it is applied to *in vivo* data such as that obtained from, e.g., electron cryotomography or confocal microscopy. Computational techniques to analyze three-dimensional membrane conformations have been described elsewhere [24] and extend our framework in a natural way to general membrane structures. As of yet there are no other experimental tools which allow the measurement of forces applied to membranes *in vivo* such as the method described here.

We can estimate the spatial resolution required to image cellular structures of interest by examining the dependence of the force at tether formation on curvature. Given an applied force F, the approximate scale of membrane bending R is given by $R \sim 2\pi k_C/F = 0.4\ (1\ \text{pN}/F)\ \mu\text{m}$. (See Appendix B.3 for a derivation of this equation.) Since typical subcellular biological forces are on the order of several piconewtons, the typical scale R of membrane bending is on the order of 100 nm; thus, the conformational force technique will be most powerful in the analysis of structures captured at resolutions lower than the optical diffraction limit.

A number of challenges remain to be dealt with before this method can be applied generically to membrane conformations observed from, e.g., electron cryotomography. The first and perhaps most serious difficulty is the heterogeneity of biomembranes [25]. Though the condition of fluidity is probably often met in the *in vivo* setting, the presence of proteins in the membrane and associated cytoskeletal elements together with the presence of many species of lipid would, in general, necessitate a more sophisticated elasticity model than the Helfrich model our method is based on. All is not lost, however. Though this method would not be applicable to whole cell deformations, if a smaller area of membrane is considered where, for example, a lipid raft has concentrated one type of lipid around an embedded protein inclusion, the forces within that region may be computed by knowledge of the boundary of that region. In general, as long as a fluid membrane region can be found for which the phospholipid bilayer on the interior of the region can be described by a bending modulus, this technique allows a measurement of the forces applied within that region. This is particularly important for data from electron cryotomography, where it is often impossible to collect data from all angles and the resulting “missing wedge” results in a partially incomplete membrane surface.

One of the most difficult and time-consuming aspects of applying the conformational force measurement to the axisymmetric vesicle shapes analyzed in this work was the tracing of the membrane from fluorescence microscopy data into a cubic spline. The general process of extracting information about various features in digital images is termed “segmentation” and is one of the most difficult problems in image analysis. The combination of an extra dimension, the lack of symmetry and other complicating factors such as the “missing wedge” all combine to make the general problem of segmenting a membrane from three-dimensional data such as a tomogram still more difficult. In addition, the typically low signal-to-noise ratio of electron cryotomography data make the extraction of features by hand, as was done with the optical tweezing experiments, prone to bias: one may see only what one expects to see. The difficulty and danger of segmenting by hand make it desirable to find reliable automated methods for segmentation of membranes. Current automatic segmentation techniques such as isosurface extraction suffer limitations such as producing meshes that are ill conditioned for analysis by the presence of, e.g., elements with poor aspect ratios and inclusion of non-membranous features in the triangular mesh. Notwithstanding, some promising progress on automatic segmentation of membranes from tomograms has been made [48] and as microscopy techniques mature, so will the methods to reliably segment features from the data they yield.

A final challenge is the determination of the unknown parameters: pressure, tension, and spontaneous curvature. The determination of these parameters using the Proximal Equilibrium Approximation is, to a large degree, a peculiarity of the *in vitro* tether formation experiments where the membrane undergoes a dramatic conformational change on the same scale as the vesicle itself. For most biological membrane remodeling examples where the membrane mechanics is relevant, the osmotic pressure difference across the membrane is negligibly small compared to factors like bending and tension. Furthermore, studies suggest that the tension (and spontaneous curvature) may be a regulated property of many cytoplasmic and organelle membranes [40]. When these parameters are known or otherwise determined, the forces can be directly computed without resort to the Proximal Equilibrium Approximation.

If these challenges can be overcome, a whole array of interesting avenues of investigation are opened through use of the conformational force measurement method we have described. Is a protein scaffold necessary to maintain the cristae and crista junctions observed in the inner membrane of a mitochondrion? How are the fenestrae of the Golgi apparatus and endoplasmic reticulum formed and regulated? Is the aggregation of Gag protein sufficient to induce the HIV budding process, as it is for the viral capsids recently modeled by molecular dynamics [23]? The promise of measuring piconewton scale forces applied to *in vivo* phospholipid bilayers presents an exciting possibility for informing quantitative models of the processes underlying the beautiful and complex world of biomembranes.

Bibliography

[1] M Ruiz. Endomembrane system diagram. URL `http://commons.wikimedia.org/wiki/Image:Endomembrane_system_diagram.svg`.

[2] B J Marsh, D N Mastronarde, K F Buttle, K E Howell, and J R McIntosh. Organellar relationships in the Golgi region of the pancreatic beta cell line, HIT-T15, visualized by high resolution electron tomography. *Proc Natl Acad Sci USA*, 98(5):2399–2406, 2001.

[3] H T McMahon and J L Gallop. Membrane curvature and mechanisms of dynamic cell membrane remodelling. *Nature*, 438(7068):590–596, 2005.

[4] H C Fertuck and M M Salpeter. Localization of acetylcholine receptor by 125I-labeled alpha-bungarotoxin binding at mouse motor endplates. *Proc Natl Acad Sci USA*, 71 (4):1376–1378, 1974.

[5] N Unwin. Refined structure of the nicotinic acetylcholine receptor at 4 angstrom resolution. *J Mol Biol*, 346(4):967–989, 2005.

[6] G Koster, M VanDuijn, B Hofs, and M Dogterom. Membrane tube formation from giant vesicles by dynamic association of motor proteins. *Proc Natl Acad Sci USA*, 100 (26):15583–15588, 2003.

[7] E Rodriguez-Boulan, G Kreitzer, and A Musch. Organization of vesicular trafficking in epithelia. *Nat Rev Mol Cell Biol*, 6(3):233–247, 2005.

[8] R D Vale and H Hotani. Formation of membrane networks in vitro by kinesin-driven microtubule movement. *J Cell Biol*, 107(6 Pt 1):2233–2241, 1988.

[9] C M Waterman-Storer, R A Worthylake, B P Liu, K Burridge, and E D Salmon.

Microtubule growth activates Rac1 to promote lamellipodial protrusion in fibroblasts. *Nat Cell Biol*, 1(1):45–50, 1999.

[10] M P Sheetz. Cell control by membrane-cytoskeleton adhesion. *Nat Rev Mol Cell Biol*, 2(5):392–396, 2001.

[11] M D Ledesma and C G Dotti. Membrane and cytoskeleton dynamics during axonal elongation and stabilization. *Int Rev Cytol*, 227:183–219, 2003.

[12] B Antonny, P Gounon, R Schekman, and L Orci. Self-assembly of minimal COPII cages. *EMBO Rep*, 4(4):419–424, 2003.

[13] B Razani and M P Lisanti. Caveolins and caveolæ: molecular and functional relationships. *Exp Cell Res*, 271(1):36–44, 2001.

[14] K Grunewald, O Medalia, A Gross, A C Steven, and W Baumeister. Prospects of electron cryotomography to visualize macromolecular complexes inside cellular compartments: implications of crowding. *Biophys Chem*, 100(1–3):577–591, 2003.

[15] T G Frey and C A Mannella. The internal structure of mitochondria. *Trends Biochem Sci*, 25(7):319–324, 2000.

[16] M S Ladinsky, D N Mastronarde, J R McIntosh, K E Howell, and L A Staehelin. Golgi structure in three dimensions: functional insights from the normal rat kidney cell. *J Cell Biol*, 144(6):1135–1149, 1999.

[17] R Sougrat, A Bartesaghi, J D Lifson, A E Bennett, J W Bess, D J Zabransky, and S Subramaniam. Electron tomography of the contact between T cells and SIV/HIV-1: implications for viral entry. *PLoS Pathog*, 3(5):e63, 2007.

[18] M L Harlow, D Ress, A Stoschek, R M Marshall, and U J McMahan. The architecture of active zone material at the frog's neuromuscular junction. *Nature*, 409(6819):479–484, 2001.

[19] Y Cheng and T Walz. Reconstructing the endocytotic machinery. *Methods Cell Biol*, 79:463–487, 2007.

[20] H Lodish, A Berk, L S Zipursky, P Matsudaira, D Baltimore, and J Darnell. *Molecular Cell Biology*. W.H. Freeman and Company, 4th edition, 2000.

[21] S Freeman. *Biological Science*. Pearson/Benjamin Cummings, 3rd edition, 2008.

[22] R Bruinsma and P Pincus. Protein aggregation in membranes. *Curr Opin Solid St M*, 1(3):401–406, 1996.

[23] B J Reynwar, G Illya, V A Harmandaris, M M Muller, K Kremer, and M Deserno. Aggregation and vesiculation of membrane proteins by curvature-mediated interactions. *Nature*, 447(7143):461–464, 2007.

[24] F Feng and W S Klug. Finite element modeling of lipid bilayer membranes. *J Comput Phys*, 220(1):394–408, 2006.

[25] T Baumgart, S T Hess, and W W Webb. Imaging coexisting fluid domains in biomembrane models coupling curvature and line tension. *Nature*, 425(6960):821–824, 2003.

[26] U Seifert. Configurations of fluid membranes and vesicles. *Adv Phys*, 46(1):13–137, Jan–Feb 1997. ISSN 0001-8732.

[27] W Helfrich. Elastic properties of lipid bilayers: theory and possible experiments. *Z Naturforsch*, 28(11):693–703, 1973.

[28] P B Canham. The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell. *J Theor Biol*, 26(1):61–81, 1970.

[29] E A Evans. Bending resistance and chemically induced moments in membrane bilayers. *Biophys J*, 14:923–931, 1974.

[30] W Rawicz, K C Olbrich, T McIntosh, D Needham, and E Evans. Effect of chain length and unsaturation on elasticity of lipid bilayers. *Biophys J*, 79(1):328–339, 2000.

[31] M Nakahara. *Geometry, Topology, and Physics*. Institute of Physics Publishing, 1990.

[32] J T Jenkins. Static equilibrium configurations of a model red blood cell. *J Math Biol*, 4(2):149–69, 1977.

[33] S J Singer and G L Nicolson. The fluid mosaic model of the structure of cell membranes. *Science*, 175(23):720–731, 1972.

[34] A Kusumi, C Nakada, K Ritchie, K Murase, K Suzuki, H Murakoshi, R S Kasai, J Kondo, and T Fujiwara. Paradigm shift of the plasma membrane concept from the two-dimensional continuum fluid to the partitioned fluid: high-speed single-molecule tracking of membrane molecules. *Annu Rev Biophys Biomol Struct*, 34:351–378, 2005.

[35] R Capovilla and J Guven. Stress and geometry of lipid vesicles. *J Phys—Condens Mat*, 16(22):S2187–S2191, Jun 2004.

[36] K Visscher, S P Gross, and S M Block. Construction of multiple-beam optical traps with nanometer-resolution position sensing. *IEEE J Sel Top Quant*, 2(4):1066–1076, Dec 1996.

[37] K C Neuman and S M Block. Optical trapping. *Rev Sci Instrum*, 75(9):2787–2809, 2004.

[38] O-Y Zhong-can and W Helfrich. Bending energy of vesicle membranes: General expressions for the first, second, and third variation of the shape energy and applications to spheres and cylinders. *Phys Rev A*, 39(10):5280–5288, May 1989.

[39] P R Bevington and D K Robinson. *Data Reduction and Error Analysis for the Physical Sciences*. McGraw-Hill, 3rd edition, 2003.

[40] A Upadhyaya and M P Sheetz. Tension in tubulovesicular networks of Golgi and endoplasmic reticulum membranes. *Biophys J*, 86(5):2923–2928, 2004.

[41] D K Fygenson, J F Marko, and A Libchaber. Mechanics of microtubule-based membrane extension. *Phys Rev Lett*, 79(22):4497–4500, 1997.

[42] D Cuvelier, I Derenyi, P Bassereau, and P Nassoy. Coalescence of membrane tethers: experiments, theory, and applications. *Biophys J*, 88(4):2714–2726, 2005.

[43] R E Waugh, J Song, S Svetina, and B Žekš. Local and nonlocal curvature elasticity in bilayer membranes by tether formation from lecithin vesicles. *Biophys J*, 61(4): 974–982, 1992.

[44] L Miao, U Seifert, M Wortis, and H G Dobereiner. Budding transitions of fluid-bilayer vesicles: The effect of area-difference elasticity. *Phys Rev E*, 49(6):5389–5407, 1994.

[45] R E Waugh and R M Hochmuth. Mechanical equilibrium of thick, hollow, liquid membrane cylinders. *Biophys J*, 52(3):391–400, 1987.

[46] V Heinrich, B Božič, S Svetina, and B Žekš. Vesicle deformation by an axial load: from elongated shapes to tethered vesicles. *Biophys J*, 76(4):2056–2071, 1999.

[47] T R Powers, G Huber, and R E Goldstein. Fluid-membrane tethers: minimal surfaces and elastic boundary layers. *Phys Rev E*, 65(4 Pt 1):041901, 2002.

[48] A Bartesaghi, G Sapiro, and S Subramaniam. An energy-based three-dimensional segmentation approach for the quantitative interpretation of electron tomograms. *IEEE Trans Image Process*, 14(9):1314–1323, 2005.

Chapter 4

Cooperative gating and spatial organization of membrane proteins through elastic interactions

All of us have had the experience of staring, perhaps bleary-eyed, into the remnants of an early morning bowl of Cheerios and were perhaps intrigued to find that the few remaining Os had clumped together in the middle of the bowl and at the edges. Equally familiar is the experience with a glass of fizzy beverage: the bubbles floating at the top will tend to stick to one another and to the edges of the glass. This well-known effect, humourously dubbed "the Cheerios effect" and illustrated in Fig. 4.1, stems from capillary forces generated by the surface tension at the liquid-gas interface [1]. The bubbles, Cheerios, and container edges bend the surface of the liquid and the energy of bending that interface due to surface tension can be minimized when objects bending the surface draw nearer to one another.

The Cheerios effect is also leveraged by the living world. Water-dwelling insects use the effect to help them navigate their watery realm and even to reach sources of nourishment. In Fig. 4.2 we show the beetle larva *Pyrrhalta*, an inept swimmer, using this effect in order to reach its leafy green food source. The larva simply arches its back, creating a meniscus which is attracted to the meniscus created by the leaf. In this way it is able to move at speeds in excess of 10 cm/s towards its intended goal [2].

Interestingly enough, this effect is also observed on the nanoscale in biomembranes. A protein inclusion in a phospholipid bilayer also bends the membrane surrounding it and, as with Cheerios and water bugs, attractive forces between those proteins may arise depending on the particulars of the mechanics of the membrane and the geometry of the transmembrane protein. An interesting twist on the macroscopic examples cited above in the case

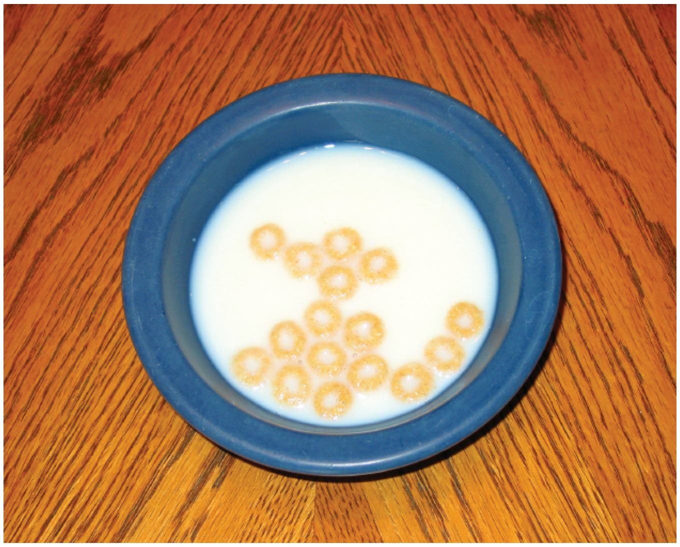


Figure 4.1: **Cheerios in a bowl of milk** are attracted to one another due to attractive forces stemming from the surface tension of the milk [1].

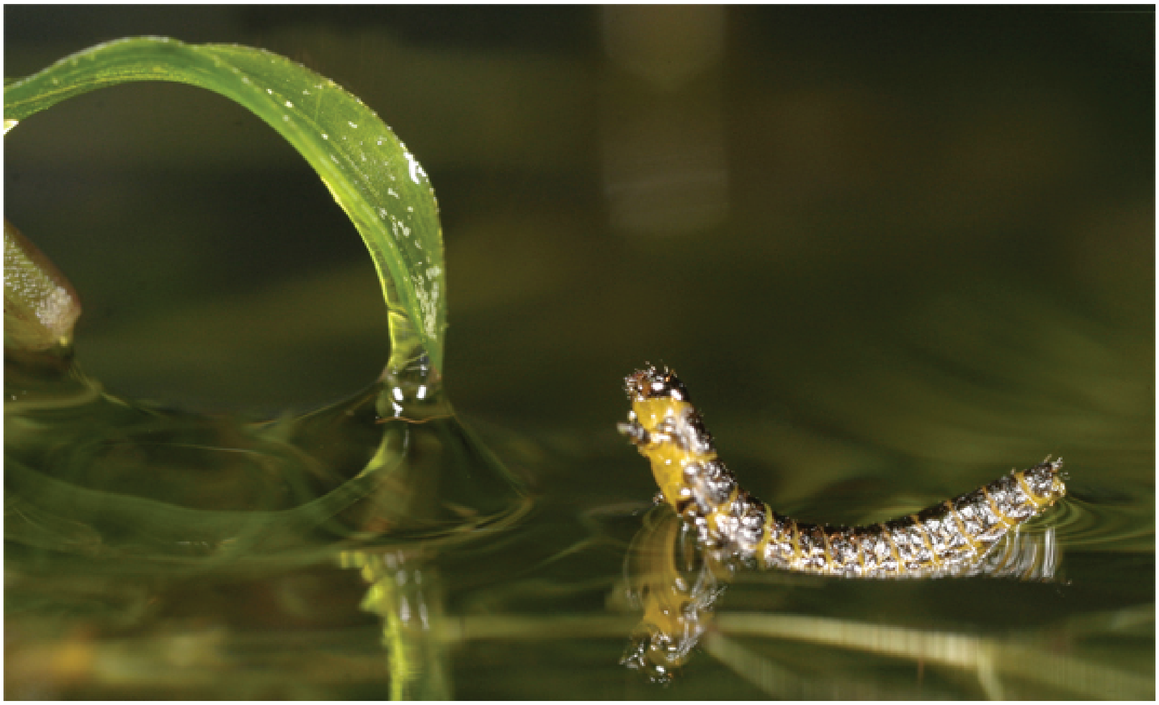


Figure 4.2: **The *Pyrrhalta* beetle larve using the "Cheerios effect" to propel itself towards a leafy snack.** By arching its back and pulling up the water in front of it, the larva is drawn by capillary forces towards the meniscus created by the leaf [2]. Photo courtesy of Hu and Bush.

of membrane protein channels is the effect of the channel's state. The membrane is bent differently when the protein is in an open as opposed to a closed state which creates an additional dependence on the *state* of the channel rather than its geometry alone. Finally, the state of the channel is coupled to the surface tension in the membrane: more tense membranes are more likely to gate the channel into an open conformation. In this chapter the interplay between tension, state, and geometry of the protein channel and the bending of a lipid bilayer are studied in detail. We find that these effects can combine to aggreate membrane channels and alter the statistics of their gating so that acting in aggregate the channels may more easily be found in the open conformation than when acting alone.

The text below was originally published as Ursell, Huang, Peterson, and Phillips, *PLoS Comput Biol* **3**(5):e81 (2007) with the exception of two additions: Markov Chain Monte Carlo (MCMC) simulations were conducted on a square lattice with periodic boundary conditions to confirm the theoretical results and are shown in Section 4.3.3; the computational method used to conduct the MCMC simulations is described in Section 4.5. Examples of input files and the source code for the MCMC simulation are given in Appendix C.

Biological membranes are elastic media in which the presence of a transmembrane protein leads to local bilayer deformation. The energetics of deformation allow two membrane proteins in close proximity to influence each other's equilibrium conformation via their local deformations, and spatially organize the proteins based on their geometry. We use the mechanosensitive channel of large conductance (MscL) as a case study to examine the implications of bilayer-mediated elastic interactions on protein conformational statistics and clustering. The deformations around MscL cost energy on the order of $10\,k_BT$ and extend $\sim 3\,\text{nm}$ from the protein edge, as such elastic forces induce cooperative gating, and we propose experiments to measure these effects. Additionally, since elastic interactions are coupled to protein conformation, we find that conformational changes can severely alter the average separation between two proteins. This has important implications for how conformational changes organize membrane proteins into functional groups within membranes.

4.1 Author summary

Membranes form flexible boundaries between the interior of a cell and its surrounding environment. Proteins that reside in the membrane are responsible for transporting materials and transmitting signals across these membranes to regulate processes crucial for cellular survival. These proteins respond to stimuli by altering their shape to perform specific tasks, such as channel proteins, which allow the flow of ions in only one conformation. However, the membrane is not just a substrate for these proteins, rather it is an elastic medium that bends and changes thickness to accommodate the proteins embedded in it. Thus, the membrane plays a role in the function of many proteins by affecting which conformation is energetically favorable. Using a physical model that combines membrane elastic properties with the structure of a typical membrane protein, we show that the membrane can communicate structural and hence conformational information between membrane proteins in close proximity. Hence, proteins can "talk" and "respond" to each other using the membrane as a generic "voice." We show that these membrane-mediated elastic forces can ultimately drive proteins of the same shape to cluster together, leading to spatial organization of proteins within the membrane.

4.2 Introduction

Biological membranes are active participants in the function and spatial organization of membrane proteins [3–5]. At the simplest level, the membrane positions proteins into a two-dimensional space, where they are often laterally organized into groups. These groups can serve specific purposes on the cell surface and within organelles, such as sensing, adhesion, and transport [6–11]. Electrostatic and van der Waals forces help drive lateral organization [12]; however, there is an additional class of purely bilayer-mediated elastic forces that can facilitate the formation of complexes of membrane proteins.

Conformational changes of membrane proteins result from a wide range of environmental factors including temperature, pH, ligand and small molecule binding, membrane voltage, and membrane tension. Likewise, conformational state is often tightly coupled with function (e.g., for ion channels) [13–15]. In this work, we demonstrate how elastic interactions can communicate information about protein conformation from one neighboring protein to another, coupling their conformational state. Additionally, we find that these interactions lead to spatial organization within the bilayer that is strongly dependent on protein conformation.

We suggest that elastic forces play a role in the function and spatial organization of many membrane proteins across many cell types, given the generically high areal density of membrane proteins [16] and the strength of these interactions. We use the mechanosensitive channel of large conductance (MscL) from *Escherichia coli* as the model protein for this study. MscL is a transmembrane homopentamer found in the plasma membrane of *E. coli* (and many other bacteria) serving as an emergency relief valve under hypo-osmotic shock [13, 17, 18]. As membrane tension increases, this nonselective ion channel changes conformation from a closed state to an open state, releasing water and osmolytes [19, 20]. Though several substates have been identified in this gating transition, the relatively short dwell-times in these substates as compared with the fully open or fully closed states allows us to approximate the protein as a simple two-state system [19, 21]. Crystal and electron-paramagnetic-resonance structures suggest the bilayer-spanning region is nearly cylindrical in both the open and closed conformations [17, 22, 23], making MscL particularly amenable to mechanical modeling. Electrophysiology of reconstituted channels allows measurement of the state of one or more of these proteins with excellent temporal and number resolution. Therefore, theoretical predictions for how elastic interactions change the gating behavior of a MscL protein can be readily tested using electrophysiology and other experimental techniques.

Following earlier work, we use continuum mechanics to break down the deformation caused by a cylindrical transmembrane protein into a term penalizing changes in bilayer thickness and a term penalizing bending of a bilayer leaflet [24–28], and we introduce a third term that preserves bilayer volume under deformation [29]. Due to its structural symmetry, MscL can be characterized by its radius and bilayer-spanning thickness in its two conformations (i.e., open and closed), neglecting any specific molecular detail (see Fig. 4.3). As these geometric parameters change with conformation, the bilayer-mediated interaction between two channels is altered. Using the interaction potentials in each combination of conformations, we explore how both the single-channel and interacting energetics affect the spatial and conformational behavior of two channels.

In the first section we cover the physical principles behind bilayer deformation due to the presence of membrane proteins. In the second section we explore the differences in

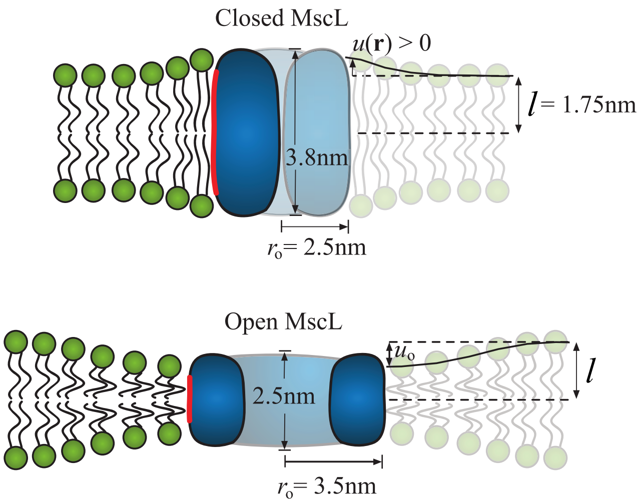


Figure 4.3: **Schematic of bilayer deformations due to MscL.** Mismatch between the hydrophobic regions of the lipid bilayer and an integral membrane protein gives rise to bending and compression deformations in each leaflet of the bilayer. The largest deformations occur at the protein-lipid interface, and over the scale of a few nanometers the bilayer returns to its unperturbed state. MscL is shown schematically at zero tension in its closed and open states with relevant dimensions. The red region of the protein indicates the hydrophobic zone. The hydrophobic mismatch at the protein-lipid interface is denoted by u_o. The deformation profile, denoted by $u(\mathbf{r})$, is measured with reference to the unperturbed leaflet thickness (l) from the protein center at $\mathbf{r} = 0$.

gating behavior of two MscL proteins when held at a fixed separation. In the third section we explore the conformational and spatial behavior of diffusing MscL proteins as a function of areal density. Finally, in the fourth section we discuss the relevance of these forces as compared with other classes of bilayer-mediated forces and support our hypotheses with results from previous experiments.

4.3 Results

4.3.1 Elastic deformation induced by membrane proteins

The bilayer is composed of discrete lipid molecules whose lateral diffusion ($D \sim 10\,\mu\text{m}^2/\text{s}$) [30] is faster than the diffusion of transmembrane proteins ($D \sim 0.1$–$1\,\mu\text{m}^2/\text{s}$) [31–33]. In the time it takes a transmembrane protein to diffuse one lipid diameter, many lipids will have exchanged places near the protein to average out the discreteness of the lipid molecules. Additionally, the transition time for protein conformational change ($\sim 5\,\mu\text{s}$) [34]

is slow compared with lipid diffusion. Hence, we argue the bilayer can be approximated as a continuous material in equilibrium with well-defined elastic properties [35]. Further, we choose to formulate our analysis in the language of continuum mechanics, rather than lateral pressure profiles [36]. In particular, each leaflet of the bilayer resists changes in the angle between adjacent lipid molecules, leading to bending stiffness of the bilayer [24, 37]. Likewise, the bilayer has a preferred spacing of the lipid molecules in-plane and will resist any changes in this spacing due to external tension [38]. Finally, experiments suggest that the volume per lipid is conserved [39, 40] such that changes in bilayer thickness are accompanied by changes in lipid spacing [4, 35].

Transmembrane proteins can compress and bend a bilayer leaflet via at least two mechanisms. The protein can force the bilayer to adopt a new thickness, matching the hydrophobic region of the protein to the hydrophobic core of the bilayer. Additionally, a noncylindrical protein can induce a slope in the leaflet at the protein-lipid interface [27, 41].

For transmembrane proteins such as MscL that can be approximated as cylindrical, symmetry dictates that the deformation energy of the bilayer is twice the deformation energy of one leaflet. Presuming the protein does not deform the bilayer too severely, we can write the bending and compression (thickness change) energies in a form analogous to Hooke's law, and account for external tension with a term analogous to PV work. We denote the deformation of the leaflet by the function $u(\mathbf{r})$, which measures the deviation of the lipid head-group from its unperturbed height as a function of the position $\mathbf{r}$ (see Fig. 4.3). In all the calculations that follow, the physical parameters chosen are representative of a typical phosphocholine (PC) lipid bilayer, and the number of lipids in this model bilayer is fixed. The energy penalizing compression of the bilayer is

$$G_{\text{comp}} = \frac{K_A}{2} \int \left(\frac{u(\mathbf{r})}{l} \right)^2 \mathrm{d}^2\mathbf{r}, \tag{4.1}$$

where K_A is the bilayer area stretch modulus ($\sim 58\, k_BT/\text{nm}^2$, k_BT is the thermal energy unit) and l is the unperturbed leaflet thickness ($\sim 1.75\,\text{nm}$) [38]. The bending energy of a leaflet is

$$G_{\text{bend}} = \frac{\kappa_b}{4} \int (\nabla^2 u(\mathbf{r}) - c_o)^2 \, \mathrm{d}^2\mathbf{r}, \tag{4.2}$$

where κ_b ($\sim 14\, k_BT$) is the bilayer bending modulus, c_o is the spontaneous curvature of the leaflet [35, 38, 42], and $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$ is the Laplacian operator.

Coupling external tension to bilayer deformations is more subtle than the previous two energetic contributions. We note that the bilayer is roughly forty times more resistant to volume change than area change [39, 40]; hence if a transmembrane protein locally thins the bilayer, lipids will expand in the area near the protein to conserve volume. Likewise, if the protein locally thickens the bilayer, lipids near the protein will condense (see Fig. 4.3). Therefore, the area change near the protein is proportional to the compression $u(\mathbf{r})$, and the work done on the bilayer is the integrated area change multiplied by tension

$$G_{\text{ten}} = \tau \int \frac{u(\mathbf{r})}{l}\, \mathrm{d}^2\mathbf{r}, \tag{4.3}$$

where τ is the externally applied bilayer tension [26, 29]. Slightly below bilayer rupture, and near the expected regime of MscL gating, $\tau \approx 2.6\, k_B T/\mathrm{nm}^2$ [19, 38]. In total, the bilayer deformation energy is

$$G = \frac{1}{2} \int \left(K_A \left(\frac{u}{l} + \frac{\tau}{K_A} \right)^2 + \kappa_b \left(\nabla^2 u - c_o \right)^2 \right) \mathrm{d}^2\mathbf{r}, \tag{4.4}$$

where we have made use of the constant bilayer area to elucidate the interplay between tension and compression. Specifically, we added a constant proportional to membrane area and τ^2, which is identically zero when calculating differences in free energy.

To obtain the length and energy scales of these deformations, we nondimensionalize the bilayer deformation energy, G. We scale both the position $\mathbf{r}$ and displacement $u(\mathbf{r})$ by $\lambda = (\kappa_b l^2/K_A)^{1/4} \approx 1\,\mathrm{nm}$, the natural length scale of deformation, to give the new variables $\boldsymbol{\rho}$ and $\eta(\boldsymbol{\rho})$, respectively, where $\boldsymbol{\rho} = \mathbf{r}/\lambda$ and $\eta(\boldsymbol{\rho}) = u(\mathbf{r})/\lambda$. Then G can be written as

$$G = \frac{\kappa_b}{2} \int \left((\eta + \chi)^2 + \left(\nabla^2 \eta - \nu_o \right)^2 \right) \mathrm{d}^2\boldsymbol{\rho}, \tag{4.5}$$

where $\nu_o = \lambda c_o$ is the dimensionless spontaneous curvature and $\chi = \tau l / K_A \lambda$ is the dimensionless tension, which is ≈ 0.09 in the regime of MscL gating. The energy scale is set by the bending modulus, κ_b.

Using the standard Euler-Lagrange equation from the calculus of variations [43], the functional for the deformation energy can be translated into the partial differential equation

$$\nabla^4 \eta + \eta + \chi = 0. \tag{4.6}$$

The deformation profile $u(\mathbf{r})$ that solves this partial differential equation depends on four boundary conditions. In the far-field, we expect the bilayer to be flat and slightly thinner in accordance with the applied tension, i.e., $|\nabla u(\infty)| = 0$ and $u(\infty) = -\tau l/K_A$, respectively. At the protein-lipid interface ($|\mathbf{r}| = r_o$) the hydrophobic regions of the protein and the bilayer must be matched, i.e., $u(r_o) = u_o$ (see Fig. 4.3), where u_o is one-half the mismatch between the hydrophobic region of the protein and the hydrophobic core of the bilayer. Finally, the slope of the bilayer at the protein-lipid interface is set to zero (i.e., $|\nabla u(r_o)| = 0$). The motivation for this last boundary condition is subtle and will be examined in more detail in the Discussion.

To understand how the deformation energy scales with hydrophobic mismatch (u_o), protein size (r_o), and tension (τ), we solve Eq. 4.6 analytically for a single cylindrical protein. The deformation energy is

$$G_{\text{single}} = \pi\kappa_b \int_{\rho_o}^{\infty} \left((\eta+\chi)^2 + \left(\nabla^2\eta - \nu_o\right)^2 \right) \rho \, \mathrm{d}\rho, \tag{4.7}$$

where $\rho_o = r_o/\lambda$ is the dimensionless radius of the protein. The leaflet deformation around a single protein is a linear combination of zeroth-order modified Bessel functions of the second kind (K_0) [27, 28]. For proteins such as MscL with a radius larger than λ (i.e., 1 nm), the deformation energy is well approximated by

$$G_{\text{single}} = \pi\kappa_b \left(\frac{u_o}{\lambda} + \frac{\tau}{K_A}\frac{l}{\lambda} \right)^2 \left(1 + \sqrt{2}\frac{r_o}{\lambda} \right). \tag{4.8}$$

The deformation energy scales linearly with protein radius and depends quadratically on the *combination* of hydrophobic mismatch (u_o) and tension (τ). This makes the overall deformation energy particularly sensitive to the hydrophobic mismatch, and hence leaflet thickness l. The deformation energy is fairly insensitive to changes in K_A (i.e., most terms in the energy are sublinear), and generally insensitive to changes in the bending modulus since $G \propto \kappa_b^{1/4}$.

Using our standard elastic bilayer parameters and the dimensions of a MscL channel (see Fig. 4.3), the change in deformation energy between the closed and open states is $\Delta G_{\text{single}} \approx 50\, k_BT$. The measured value for the free energy change of gating a MscL protein, including internal changes of the protein and deformation of surrounding lipids is $\approx 51\, k_BT$

[21]. This close correspondence does not indicate that bilayer deformation accounts for all of the free energy change of gating [44], but does suggest that it is a major contributor.

The gating energy of two channels in close proximity is a complex function of their conformations and the distance between them. As two proteins come within a few nanometers of each other (i.e., a few λ), the deformations that extend from their respective protein-lipid interfaces begin to overlap and interact. The bilayer adopts a new shape (i.e., a new $u(\mathbf{r})$), distinct from the deformation around two independent proteins, and hence the total deformation energy changes as well. This is the physical origin of the elastic interaction between two bilayer-deforming proteins [25, 26].

Each protein imposes its own local boundary conditions on the bilayer, which vary with conformation; hence, the deformation around a pair of proteins is a function of their individual conformation and the distance between them. A MscL protein has two distinct conformations, hence there can be pairwise interactions between two closed channels, an open and a closed channel, or two open channels (see Fig. 4.4). Tension also affects the deformations. The hydrophobic mismatch can be either positive or negative (i.e., the protein can be thicker or thinner than the bilayer), thus tension will strengthen the interaction of proteins that are thicker than the bilayer (e.g., the closed-closed interaction of two MscL proteins) and weaken the interaction of proteins that are thinner than the bilayer (e.g., the open-open interaction). This effect is demonstrated in Fig. 4.4. The interactions due to leaflet deformations have been explored before [25, 26], but our model elucidates the role that these interactions can play in communicating conformational information between proteins. Additionally, in our model, tension can play an important role in determining the overall deformation energy around a protein.

In a one-dimensional model, the interaction potentials can be solved for analytically. For two identical proteins in close proximity (e.g., closed-closed and open-open interactions), the approximate shape of the potential is linearly attractive $\kappa_b(u_o/\lambda)^2(d/2\lambda - \sqrt{2})$. Between two dissimilar proteins in close proximity (e.g., open-closed interaction), the potential is approximately $\kappa_b(u_o/\lambda)^2\pi^4/4(d/\lambda)^3$, where in both cases d is measured from the edges of the proteins. This illustrates the general principle that two similar proteins attractively interact, while two dissimilar proteins tend to repel each other. This one-dimensional model helps build intuition for what governs the strength of elastic interactions. Whether the interaction is attractive or repulsive, the strength of the interaction is dominated by its

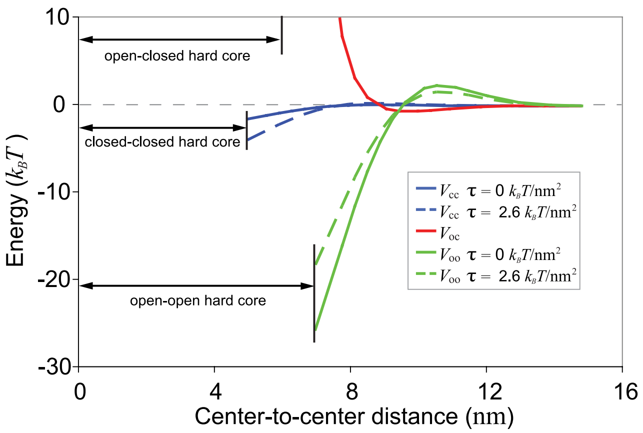


Figure 4.4: **Elastic potentials between two MscL proteins.** To minimize deformation energy, two transmembrane proteins exert elastic forces on each other. MscL has three distinct interaction potentials between its two distinct conformations. External tension weakens the interaction between two open channels (V_{oo}) and strengthens the interaction between two closed channels (V_{cc}), but has almost no effect on the interaction between an open and closed channel (V_{oc}). The open-open and closed-closed interactions are both more strongly attracting than the open-closed interaction, indicating that elastic potentials favor interactions between channels in the same state. The "hard core" distance is where the proteins' edges are in contact.

quadratic dependence on the combination of hydrophobic mismatch and tension-induced thinning. Hence, interactions between proteins that deform the membrane more severely are simultaneously more sensitive to tension. These effects are demonstrated in Fig. 4.4, where the closed-closed interaction, which has less hydrophobic mismatch, is both weaker and less sensitive to tension than the open-open interaction. In a two-dimensional bilayer, the geometry of the two proteins makes it difficult to solve for the interaction analytically, thus numerical techniques were used (see Section 4.5).

This theoretical framework provides a strong foundation for understanding how protein geometries and lipid properties give rise to elastic interactions. With this, we can investigate how elastic forces change the conformational statistics of a two-state protein population.

4.3.2 Gating behavior of two interacting channels

To probe the range of separations over which elastic interactions affect the gating of two MscL proteins, we need to account for the noninteracting energetics of gating a single channel in addition to the interactions between two channels. The noninteracting energy is the sum of three effects. First, there is some energetic cost to deform the surrounding membrane, which we already calculated as ΔG_{single}. Second, there is some cost to change the protein's internal conformation, independent of the membrane. Together, these first two effects are the gating energy $\Delta G_{\text{gate}} \approx 51\, k_B T$ for MscL [21]. Finally, there is an energetic mechanism that overcomes these costs and opens the channel as tension increases. This mechanism is provided by the bilayer tension working in concert with the conformational area change of the protein ($\Delta A \approx 20\,\text{nm}^2$ for MscL [21]). Given the experimentally determined values for ΔG_{gate} and the area change during gating, the critical tension, defined by $\Delta G_{\text{gate}} = \tau_c \Delta A$, is $\tau_c = 2.6\, k_B T/\text{nm}^2$.

In our thermodynamic treatment, we need to keep track of the conformations of each protein in a population in a way that allows us to tabulate the noninteracting and interacting contributions to the free energy. To this end, we assign a state variable, s_i, to each channel, indicating the conformational state of a protein, where $s_i = 0$ indicates that the ith channel is closed and $s_i = 1$ indicates that the ith channel is open. The noninteracting energy for two channels is then

$$H_{\text{non}}(s_1, s_2; \tau) = (\Delta G_{\text{gate}} - \tau \Delta A)\,(s_1 + s_2). \tag{4.9}$$

If both channels are closed ($s_1 = s_2 = 0$), the free energy is defined to be zero. If one channel is open and the other closed ($s_1 = 1, s_2 = 0$, or, $s_1 = 0, s_2 = 1$), this counts as the cost to gate one channel working against the benefit at a particular tension to opening the channel. Likewise, this counts twice if both channels are open ($s_1 = s_2 = 1$). We will measure all energies that follow in units of $k_B T$ ($\approx 4.14 \times 10^{-21} J$).

As we alluded to earlier, the interacting component of the free energy between two channels is a function of their states (s_1 and s_2), their edge separation (d), and the tension. Using a numerical relaxation technique to minimize the functional in Eq. 4.5 (see Section 4.5), we calculated the interaction potentials $H_{\text{int}}(s_1, s_2, d; \tau)$ for a range of tensions and separation distances (see Fig. 4.4). The total energy, $H_{\text{non}} + H_{\text{int}}$, is used to derive the Boltzmann weight for the three possible configurations of the two-channel system,

$$z(s_1, s_2) = e^{-(H_{\text{non}}(s_1, s_2; \tau) + H_{\text{int}}(s_1, s_2, d; \tau))}. \tag{4.10}$$

The probability that the system has two closed channels is

$$P_0 = \frac{z(0,0)}{Z}, \tag{4.11}$$

where the partition function Z is the sum of the Boltzmann weights for all possible two-channel configurations,

$$Z = \sum_{s_1,s_2=0}^{1} z(s_1,s_2) = z(0,0) + 2z(0,1) + z(1,1). \tag{4.12}$$

Likewise, the probabilities for the system to have exactly one or two open channels are

$$P_1 = \frac{2z(0,1)}{Z} \quad \text{and} \quad P_2 = \frac{z(1,1)}{Z}, \tag{4.13}$$

respectively. Finally, the probability for any one channel in this two channel system to be open is

$$P_{\text{open}}(\tau, d) = \frac{z(0,1) + z(1,1)}{Z}. \tag{4.14}$$

If the distance between two channels is much greater than λ, they will behave independently. As the channels get closer ($d \lesssim 5\lambda$) they begin to interact and their conformational statistics are altered. P_{open} as a function of tension for certain fixed separations is shown in Fig. 4.5. The open-open interaction is the most energetically favorable for most separations, hence the transition to the open state generally shifts to lower tensions as the distance between the two proteins is decreased. Though the edge spacing can be small, even fractions of the width of a lipid molecule, the two-dimensional nature of the interaction means that the majority of the interaction is mediated by lipids in the intervening region between the two proteins. Thus, a continuum model is still applicable, albeit less accurate, for very small protein separations.

Interactions also affect channel "sensitivity," defined as the derivative of P_{open} with respect to tension, which quantifies how responsive the channel is to changes in tension. The full-width at half-maximum of this peaked function is a measure of the range of tension over which the channel has an appreciable response. The area under the sensitivity curve is equal to 1, hence increases in sensitivity are always accompanied by decreases in range of response, as demonstrated by the effects of the beneficial open-open interaction on channel

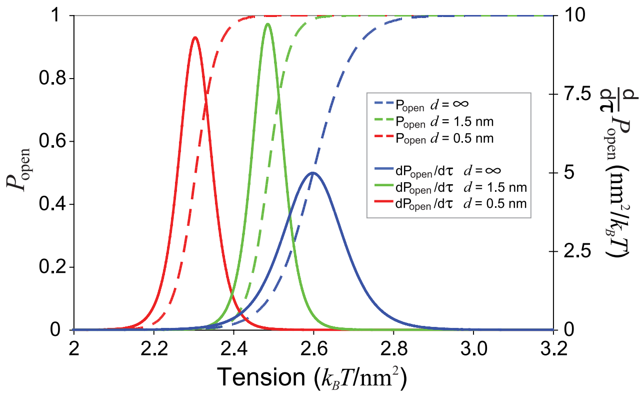


Figure 4.5: **Conformational statistics of interacting MscL proteins.** Interactions between neighboring channels lead to shifts in the probability that a channel will be in the open state (dashed lines). The sensitivity and range of response to tension, $\mathrm{d}P_{\text{open}}/\mathrm{d}\tau$, are also affected by bilayer deformations (solid lines). P_{open} and $\mathrm{d}P_{\text{open}}/\mathrm{d}\tau$ are shown for separations of 0.5 nm (red) and 1.5 nm (green) with reference to noninteracting channels at $d = \infty$ (blue). Interactions shift the critical gating tension for the closest separation by $\sim 12\%$. Additionally, the peak sensitivity is increased by $\sim 90\%$ from $\sim 5\,\text{nm}^2/k_BT$ to $\sim 9.5\,\text{nm}^2/k_BT$, indicating a Hill coefficient of ~ 2.

statistics (see Fig. 4.5).

In summary, we find that elastic interactions between two proteins have significant effects when the protein edges are closer than ~ 5 nm. At these separations the elastic interactions alter the critical gating tension and change the tension sensitivity of the channel (see Fig. 4.5). The critical gating tension and sensitivity are the key properties that define the transition to the open state, and are analogs to the properties that define the transition of *any* two-state membrane protein. Hence, we have shown that elastic interactions can affect protein function at a fundamental level.

4.3.3 Interactions between diffusing proteins

With an understanding of how two proteins will interact at a fixed distance, we now study the conformational statistics of two freely diffusing MscL proteins allowed to interact via their elastic potentials. In biological membranes, transmembrane proteins that are not rigidly attached to any cytoskeletal elements are often free to diffuse throughout the membrane and interact with various lipid species, as well as other membrane proteins. On average, the biological areal density of such proteins is high enough ($\sim 10^3$–10^4 μm^{-2} [16]) that elastic interactions should alter the conformational statistics and average protein separations.

We expect that if two MscL proteins are diffusing and interacting, the open probability will be a function of their areal density as well as the tension. It then follows that for a given areal density, elastic interactions will couple conformational changes to the average separation between the proteins. To calculate the open probability of two diffusing MscL proteins, the Boltzmann weight for these proteins to be in the conformations s_1 and s_2 must be summed at every possible position, giving

$$\langle z(s_1,s_2)\rangle = e^{-H_{\text{non}}} \int\int e^{-H_{\text{int}}}\, \mathrm{d}^2\mathbf{r}_1\, \mathrm{d}^2\mathbf{r}_2, \tag{4.15}$$

where $\langle \ldots \rangle$ indicates a sum over all positions. The distance between the proteins is measured center-to-center as $|\mathbf{r}_1 - \mathbf{r}_2|$ and only the absolute distance between the two proteins determines their interaction, hence we can rewrite the integrand as a function of $r = |\mathbf{r}_1 - \mathbf{r}_2|$. We then change the form of the integrand to

$$e^{-H_{\text{int}}(s_1,s_2,r;\tau)} = 1 + f_{12}(r), \tag{4.16}$$

which allows us to separate the interacting effects from the noninteracting effects (the function f_{12} is often called the Mayer-f function). Thus, the position-averaged Boltzmann weights are

$$\langle z(s_1,s_2)\rangle = e^{-H_{\text{non}}} \left(1 + \frac{2\pi}{A}\int_0^\infty f_{12}(r) r\, \mathrm{d}r\right), \tag{4.17}$$

where A is the total area occupied by the two proteins. Following our previous calculations, the probability that any one channel is open in this two-channel system is

$$P_{\text{open}}(\tau,\alpha) = \frac{\langle z(0,1)\rangle + \langle z(1,1)\rangle}{\langle Z\rangle}, \tag{4.18}$$

where α is the protein areal density (i.e., $\alpha = 2/A$) and $\langle Z\rangle = \sum_{s_1,s_2} \langle z(s_1,s_2)\rangle$.

In Fig. 4.6A, we plot $P_{\text{open}}(\tau,\alpha)$ over a wide range of areal density, from the area of ~ 100 lipids up to areas on the whole-cell scale. The more beneficial open-open interaction tends to shift the transition to the open state to lower tensions, with the most pronounced effect being when the two proteins are most tightly confined. For the estimated biological membrane protein density of $\sim 10^3$–10^4 μm^{-2} (or $\sim$ 10–30 nm spacing) [16], the gating tension is decreased by $\sim 13\%$, the sensitivity is increased by $\sim 85\%$, and the range of response is decreased by $\sim 55\%$. For the *in vivo* expression of MscL of ~ 1–$10\,\mu\text{m}^{-2}$ [45], the gating tension is reduced by $\sim 7\%$, the sensitivity is increased by $\sim 70\%$, and the range of response is decreased by $\sim 40\%$. These changes in gating behavior are accessible to electrophysiological experiments where MscL proteins can be reconstituted at a known areal density (~ 0.1–$10\,\mu\text{m}^{-2}$), and the open probability can be measured as a function of tension.

In addition to lowering the critical tension and augmenting channel sensitivity, the conformational states of channels are tightly coupled by their interaction. The probability that exactly one channel is open (P_1) decreases dramatically as areal density increases. For tensions above the critical tension, interacting channels ($\sim 10^3\,\mu\text{m}^{-2}$) are nearly three orders of magnitude less likely to gate as single channels than their noninteracting counterparts ($\sim 10^{-3}\,\mu\text{m}^{-2}$), as shown in Fig. 4.6B. Additionally, the tension at which it is more likely to have *both* channels open, rather than a single channel, is significantly lower for interacting channels, signaling that gating is a tightly coupled process. In addition to altering the open probability of two channels, the favorable open-open interaction provides an energetic barrier to leaving the open-open state. Based on a simple Arrhenius argument, the average open lifetime of two channels that are both open and interacting will be orders of magnitude longer than two open but noninteracting channels.

Having shown conformational coupling over a range of areal densities, it is reasonable to expect that elastic interactions will affect the separation between two proteins. We ask, how do interactions affect the average separation between proteins? How often will we find the two proteins separated by a distance small enough that we can consider them "dimerized"?

From Eq. 4.15 and 4.16, it follows that the Boltzmann weight for the two proteins to be separated by a distance r is

$$z(s_1, s_2, r) = e^{-H_{\text{non}}} \frac{2\pi}{A}(1 + f_{12})r. \tag{4.19}$$

The probability that the proteins are separated by a distance r, regardless of their conformation, is

$$P(r) = \frac{Z(r)}{\langle Z \rangle} = \frac{\sum_{s_1,s_2} z(s_1, s_2, r)}{\langle Z \rangle}, \tag{4.20}$$

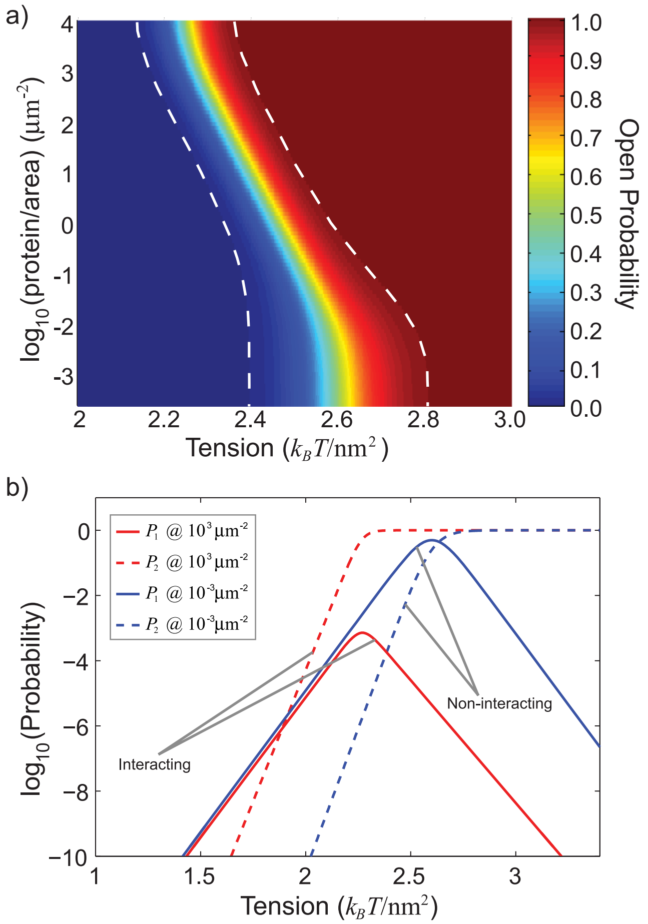


Figure 4.6: **Elastic interactions lower open probability transition and couple conformation changes.** Two MscL proteins in a square box of area A diffuse and interact via their elastic potentials. (A) At low areal density, the response to tension is the same as an independent channel. As the areal density increases, the more beneficial open-open interaction (see Fig. 4.4) shifts the open probability to lower tensions and decreases the range of response (dashed lines) while increasing the peak sensitivity, indicating that areal density can alter functional characteristics of a transmembrane protein. (B) The probability for exactly one channel to be open (P_1, solid lines) is shown at a low (blue) and high (red) areal density. For tensions past the critical tension, interacting channels are $\sim 1{,}000$ times less likely to gate individually. The probability for both channels to be open simultaneously (P_2, dashed lines) is shown for low (blue) and high (red) areal density. The tension at which two simultaneously open channels are favored is significantly lower for interacting channels. Together these facts signify a tight coupling of the conformational changes for two interacting channels.

from which we calculate the average separation

$$\begin{aligned}\langle r\rangle &= \frac{1}{\langle Z\rangle}\int Z(r)r\,\mathrm{d}r \\ &= \frac{1}{\langle Z\rangle}\sum_{s_1,s_2} e^{-H_{\text{non}}}\left(\delta\frac{\pi}{6}\sqrt{A}+\frac{2\pi}{A}\int_0^\infty f_{12}r^2\,\mathrm{d}r\right).\end{aligned} \tag{4.21}$$

This equation is valid as long as the area does not confine the proteins so severely that they are sterically forced to interact. The constant δ is an order-one quantity that is defined by the entropic component of average separation on a surface $\mathcal{S}$, given by

$$\int\int_{\mathcal{S}(A)} |\mathbf{r}_1-\mathbf{r}_2|\frac{\mathrm{d}^2\mathbf{r}_1}{A}\frac{\mathrm{d}^2\mathbf{r}_2}{A} = \delta\frac{\pi}{6}\sqrt{A} \tag{4.22}$$

and depends on the actual shape of the surface. For a square box, $\delta \approx 1$, and for a circle, $\delta \approx \sqrt{2}$. The average separation of two MscL proteins as a function of tension is plotted for various areal densities in Fig. 4.7. For certain densities, elastic interactions couple the conformational change from the closed to open state with a decrease in the average separation by more than two orders of magnitude. Our estimates of biological membranes yield fairly high membrane-protein densities ($\sim 10^3$–10^4 μm^{-2}) [16], which correspond to the more highly confined conditions on Fig. 4.6–4.8. In the native *E. coli* plasma membrane, MscL, with a copy number of ~ 5 [45], is present at a density of ~ 1–10 μm^{-2}, which means that even membrane proteins expressed at a low level are subject to the effects of elastic interactions.

To quantify the effects of interaction on the spatial organization of two channels, we define a "dimerized" state by the maximum separation below which two channels will favorably interact with an energy greater than k_BT (i.e., $H_{\text{int}}(s_1,s_2,\tau,r) < -1$). This defines a critical separation, $r_c(s_1,s_2,\tau)$, which depends on the conformations of each protein and the tension. The probability that the two proteins are found with a separation less than or equal to r_c is

$$P_{\text{dimer}}(\tau,\alpha) = \frac{1}{\langle Z\rangle}\sum_{s_1,s_2} e^{-H_{\text{non}}}\left(\frac{\pi r_c^2}{A}+\frac{2\pi}{A}\int_0^{r_c} f_{12}r\,\mathrm{d}r\right). \tag{4.23}$$

This "dimerization probability" is plotted as a function of tension and areal density in Fig. 4.8.

At low tension and high areal density, the channels are closed and near enough that

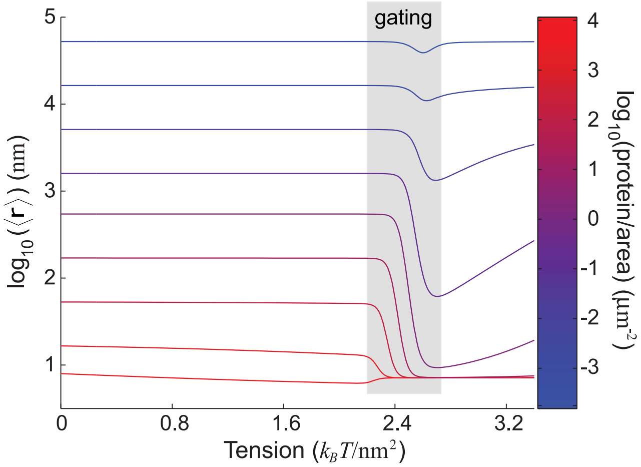


Figure 4.7: **Average separation between proteins drops significantly due to elastic interactions.** The average separation between two diffusing MscL proteins in a box of area A is plotted as a function of tension for a range of areal densities, each shown as a different line color. The grey region roughly indicates when gating is occurring. At low areal density (mostly blue), the conformational change does not draw the proteins significantly closer together. As the areal density increases, the conformational change is able to draw the proteins up to ~ 100 times closer than they would otherwise be. At the highest areal density (mostly red), the steric constraint of available area intrinsically positions the proteins close to one another regardless of their conformation. The average separation begins to increase again as higher tension weakens the open-open interaction.

the closed-closed interaction can dimerize them a fraction of the time. Keeping the areal density high, increasing tension strengthens the closed-closed interaction, and the dimerization probability increases until tension switches the channels to the open state, where the significantly stronger open-open interaction dimerizes them essentially 100% of the time. When the areal density decreases to moderate levels, as denoted by the white dashed lines in Fig. 4.8, the dimerization is strongly correlated with the conformational change to the open state. The zero tension separation between the two proteins for this one-to-one correlation is ~ 40 nm to ~ 2 µm. Finally, when the areal density is very low, entropy dominates, and neither the closed-closed, nor the open-open interaction is strong enough to dimerize the channels. Understanding the onset and stability of dimers is an important first step in understanding the formation of larger oligomers of membrane proteins. As the areal density of membrane proteins increases, clusters of more than two proteins become favorable and are energetically stabilized by their multibody interactions. For a rigorous theoretical treatment of these multibody interactions, we refer the interested reader to [46–48].

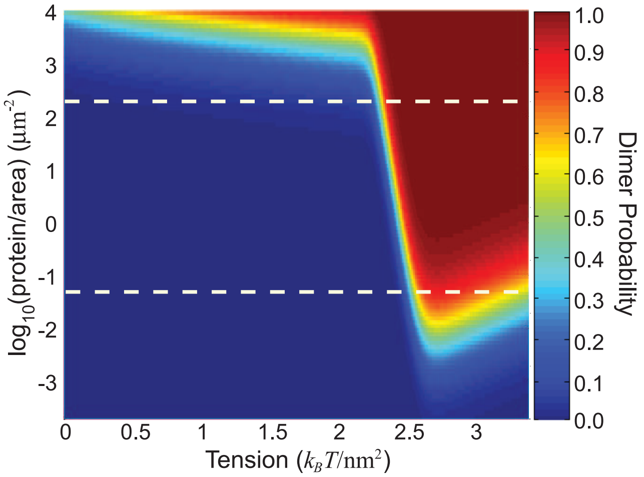


Figure 4.8: **Elastic interactions tightly couple conformational change with protein dimerization.** Diffusing MscL proteins are considered dimerized when they are close enough that they attract with an energy greater than k_BT. At high areal density, the net attractive closed-closed interaction is sufficient to dimerize the two channels part of the time. As the areal density decreases, the closed-closed interaction is not strong enough to dimerize the two channels—now dimerization only happens at higher tensions after both channels have switched to the open conformation. As the areal density decreases further, the open-open interaction is no longer strong enough to overcome entropy. This loss of dimerization is amplified by the fact that the open-open interaction is weaker at higher tensions (see Fig. 4.4). The white dashed lines roughly indicate the range of areal densities for which dimerization probability and open channel probability are equal to each other (see Fig. 4.6).

We tested the analytical results of this section with a Markov Chain Monte Carlo (MCMC) simulation, shown in Fig. 4.9 and described in more detail in Section 4.5. The MCMC simulations were run with two proteins on a square lattice with sides of length 49.5 nm. The simulation and theoretical results show excellent agreement and lend further confidence to the conclusions of this section with one exception: the average separation at low tension. In this regime, the channels are nearly always closed and interact only very weakly—the average separation is essentially just the average separation of two freely diffusing channels. On a square lattice with periodic boundary conditions this separation cannot exceed $L\sqrt{2}/2$ where L is the length of one side of the lattice; in the analytical calculations all separation lengths were integrated over. This artifact leads to the disagreement shown at low tension in Fig. 4.9C.

In summary, we have shown that over a broad range, areal density plays a nontrivial role in allowing two channels to communicate conformational information. This communication

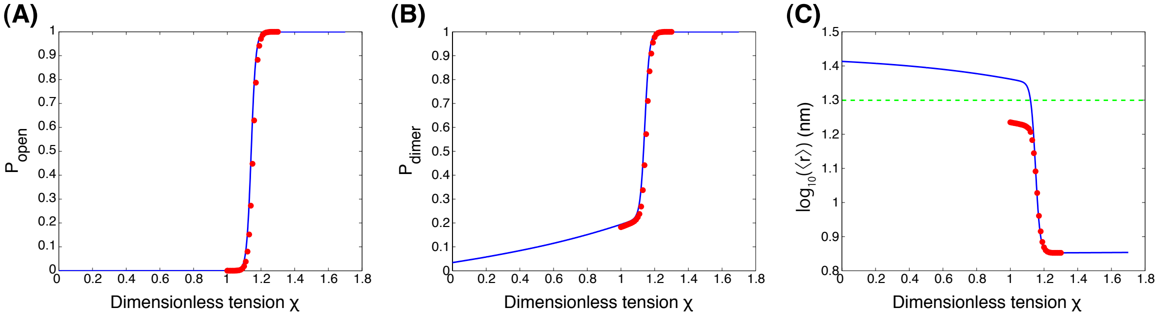


Figure 4.9: **Comparison of theoretical results (blue) with MCMC simulation (red).** Theory and simulation show excellent agreement for (A) P_{open}, (B) P_{dimer} and (C) $\langle r \rangle$ at higher tension values. The average separation at low tension does not match theory because the lattice disallows all configurations where the channels are separated by more than $L\sqrt{2}/2$ where L is the length of one side of the lattice. This artifact leads to the channels approaching the average separation for (nearly) freely diffusing proteins on a square lattice with periodic boundary conditions, shown as a dotted green line. This effect could be corrected for by increasing the area of the lattice at constant areal density of proteins.

can lead to large changes in the average separation between two proteins and the probability that they will be found together in a dimerized state. This may have implications for how conformational changes of transmembrane proteins in biological membranes are able to facilitate the formation of functional groups of specific proteins.

4.4 Discussion

In this section, we will perform a brief survey of other bilayer-mediated forces between proteins and make a comparison of their relative length and energy scales. We will also address some of the finer details of our model and how boundary conditions can affect deformation energy around a protein. Finally, we will suggest experiments using MscL to observe the predicted changes in conformational statistics, as well as provide evidence from previous experiments that leaflet interactions lead to significant changes in conformational statistics.

There are at least two other classes of purely bilayer mediated forces between membrane proteins. The first is a different type of bilayer deformation that bends the mid-plane of the bilayer. This arises from transmembrane proteins with a conical shape that impose a bilayer slope at the protein-lipid interface [41, 49]. If the protein does not deform the bilayer too severely, the mid-plane deformation energy of a bilayer is

$$G_{\text{mid}} = \int \left(\frac{\tau}{2} (\nabla h(\mathbf{r}))^2 + \frac{\kappa_b}{2} (\nabla^2 h(\mathbf{r}))^2 \right) \, \mathrm{d}^2\mathbf{r}, \tag{4.24}$$

where $h(\mathbf{r})$ is the deviation of the height of the mid-plane from a flat configuration [28, 37]. These kinds of interactions have been calculated for a variety of bilayer curvature environments and protein shapes at zero tension [49, 50]. Using a bilayer bending modulus of $\sim 100\,k_BT$, attractive interactions of order ~ 1–$5\,k_BT$ were found when the proteins were separated by one to two protein radii (which we estimate to be 5–10 nm measured center-to-center for a typical transmembrane protein). If we adjust the energy scale to be consistent with a PC bilayer bending modulus of $\sim 14\,k_BT$, this lowers the interaction energetics to ~ 0.4–$2\,k_BT$. Hence, although the length scale of appreciable interaction for mid-plane deformation is longer than for leaflet deformation, the interaction energies from leaflet deformation can be ten times greater depending on protein geometry. The deformation fields $h(\mathbf{r})$ and $u(\mathbf{r})$ exert their effects independent of one another [28], suggesting that while energetically weaker than leaflet deformation, mid-plane deformation probably also contributes to the spatial organization and conformational communication between transmembrane proteins. However, for the resting tension of many biological membranes [51], the interaction due to midplane deformation has a length scale ($\sqrt{\kappa_b/\tau} \approx 50\,\text{nm}$) longer than the nominal spacing of proteins (≈ 10–$30\,\text{nm}$ [16]). Thus, one protein can shield other proteins from feeling the deformation of a neighboring protein, and hence interactions are not (in general) pairwise additive. In fact, this is a general feature for both leaflet and midplane elastic interactions—they can be shielded by the presence of other proteins, and nonspecific protein interactions can couple to conformation and position within the membrane in the same manner as the specific interactions we have explored in the previous sections.

The second class of bilayer-mediated forces is a product of the thermal fluctuations of the bilayer. There is a small thermal force due to the excluded volume between two proteins, calculated via Monte Carlo methods to have a favorable $\sim 2\,k_BT$ interaction [52]. This force only exists when the proteins are separated by a fraction of the width of a lipid molecule. There is also a long-range thermal force, due to the surface fluctuations of the bilayer, which tends to drive two rigid proteins closer together [12, 53]. This force is proportional to $1/r^4$ and is generally attractive. Estimates using this power law indicate that the interaction is

$\sim 1\,k_BT$ when the center-to-center separation is roughly two protein radii. Though elegant, the derivation of this force is only valid in the far-field, thus how this force might contribute to conformational communication between proteins in close proximity is not entirely clear.

To gauge the overall importance of leaflet interactions, the virial coefficient used in Eq. 4.17,

$$C_V = 2\pi \int_0^\infty f_{12}(r) r \,\mathrm{d}r, \tag{4.25}$$

quantifies how the combination of length and energy scales leads to a deviation from noninteracting behavior; it is *exponentially* sensitive to the energy but only *quadratically* sensitive to the length scale. One can interpret the virial coefficient as the area per particle that makes the competing effects of entropy and interaction equivalent. Using this measure, we estimated the virial coefficients for all of these bilayer-mediated forces and found that leaflet deformations, while having a short length scale, actually lead to the most significant deviation from noninteracting behavior, due to their high energy scale. We estimate the virial coefficients from leaflet interactions to be $\sim 10^4$–$10^6\,\mathrm{nm}^2$, while mid-plane bending interactions are $\sim 10^3\,\mathrm{nm}^2$, and the thermal forces $\sim 10^2\,\mathrm{nm}^2$.

Examining our elastic model in greater detail, we have assumed that the slope of the leaflet at the protein-lipid interface is zero, which eliminates any dependence on the spontaneous and Gaussian curvatures of the leaflet. In a more general continuum-mechanical theory, the slope would be left as a free parameter with respect to which the energy could be minimized [27]. We examined this possibility and found that, at most, the energy was reduced by a factor of two. Spontaneous curvature couples to the slope of the leaflet at the protein-lipid interface; however, the spontaneous curvature of bilayer-forming lipids, such as phosphocholines, is small [54]. In addition, for proteins whose radius is larger than λ, if we assume the modulus associated with Gaussian curvature is of the same magnitude as the mean curvature modulus (κ_b) [55], the Gaussian contribution to the deformation energy is a second-order effect. We also examined the possibility of a term proportional to $(\nabla u)^2$, using the interfacial tension ($\sim 5\,k_BT/\mathrm{nm}^2$) as a modulus for this term; these effects were also second-order. Finally, we imposed the "strong hydrophobic matching" condition at the protein-lipid interface, assuming that the interaction of lipids with the hydrophobic zone of the protein is very favorable. Relaxing this condition would result in a decrease in the magnitude of the hydrophobic matching condition, u_o, and hence an overall decrease of interaction energetics [28].

There are experimental and mechanical reasons to believe the boundary slope on a cylindrical protein is small. The membrane protein gramicidin was used to comment on this so-called “contact angle” problem of lipid-protein boundary conditions [24, 56]. It was found that indeed the slope was nearly zero. From a mechanical standpoint, if the lipids are incompressible, a positive boundary slope that deviates significantly from zero would correspond to the creation of an energetically costly void at the protein-lipid interface when the protein is shorter than the bilayer. Conversely, lipids would have to penetrate the core of the protein to produce a negative slope when the protein is taller than the bilayer, again a very costly proposition.

We examined a roughly cylindrical protein and demonstrated the interesting effects elastic interactions would have in such cases. However, the scope of possible effects increases when noncylindrical proteins are considered. Most notably, noncylindrical cross-sections allow for orientational degrees of freedom in the interaction, hence such proteins do not just attract or repel each other, but would have preferred orientations in the membrane with respect to each other.

Measuring the changes in conformational statistics of two MscL proteins held at a fixed separation would allow for quantitative verification of our predictions. Electrophysiology is a common tool used to probe the conformation of ion channels, and is routinely used to measure the open probability of a single MscL protein *in vitro* [19, 21, 57]. Cysteine point mutations on the outer edges of two MscL proteins [22] could be covalently linked [58–61] by a polymer with a specific length (~ 0.5–$10\,\mathrm{nm}$) to control the separation distance [62, 63]. Linking stoichiometry could be controlled genetically [64] to ensure one channel interacts with only one other channel.

Similar experiments have been performed using gramicidin A channels [65]. The conducting form of gramicidin A is a cylindrical transmembrane protein which, like MscL, tends to compress the surrounding bilayer [24, 35, 66] and hence have a beneficial interaction. Electrophysiology of polypeptide-linked gramicidin channels [65] qualitatively supports our hypothesis that the beneficial interaction of the deformed lipids around two gramicidin channels significantly increases the lifetime of the conducting state [67]. As another example, recent FRET studies showed that oligomerization of rhodopsin is driven by precisely these kinds of elastic interactions, and exhibits a marked dependence on the severity of the deformation as modulated by bilayer thickness [68]. Additionally, recent experimental work has shown that the bacterial potassium channel KscA exhibits coupled gating and spatial clustering in artificial membranes [69].

In summary, we have demonstrated that leaflet deformations are one of the key mechanisms of bilayer-mediated protein-protein interactions. We provided support for our choice of boundary conditions at the protein-lipid interface, and suggested that extensions of our model have exciting possibilities for the specificity of elastic interaction. Finally, we suggested how one might measure the predicted changes in conformational statistics and drew an analogy to previous gramicidin channel experiments.

Conclusion

We have described the important role of an elastic bilayer in the function of, and communication between, membrane proteins. The interplay between the length scale of interaction (a few nanometers) and the energetics of interaction (on the order of $10\,k_BT$) mean elastic interactions are relevant over a wide range of areal densities, from protein separations on the order of nanometers up to a micron or more. Transmembrane proteins can communicate information about their conformational state via the deformations they cause in the surrounding bilayer. We demonstrated with a model protein, the tension-sensitive channel MscL, how deformations lead to elastic forces and result in cooperative channel gating. Additionally, we found that elastic interactions strongly correlate conformational changes to changes in spatial organization, aggregating two channels even at low areal densities, and hence bringing them together over very large distances relative to their size.

The elastic theory presented here can be easily expanded to include more complex deformation effects (such as spontaneous curvature) and protein shapes, and is applicable to any protein that causes thickness deformation in the membrane. Our calculations for the conformational statistics, average separation, and dimerization are insensitive to the actual stimulus triggering the conformational change. Hence, we suggest that elastic interactions are likely to play a role in the function and organization of many membrane proteins that respond to environmental stimuli by forming functional groups of multiple membrane proteins. Recent work suggests chemotactic receptors in *E. coli* function by precisely this kind of spatially clustered and conformationally coupled modality [70].

4.5 Materials and methods

To compute the pairwise elastic potentials in Fig. 4.4, we discretize the bilayer height, $\eta(\rho)$, and minimize the deformation energy in Eq. 4.5 using a preconditioned conjugate gradient approach. A separate minimization with the aforementioned boundary conditions, including the zero-slope boundary condition, was computed for each combination of channel configurations, protein-protein separation, and bilayer tension. Except in the regions of the bilayer nearby a protein at position (x_o, y_o), we use a Cartesian grid with spacing $dx = dy = 0.1\,\lambda = 0.093\,\text{nm}$. However, since deformations in the bilayer are largest at the circular membrane-protein interface, we interpolate between a polar grid at the interface at $|\mathbf{r}| = r_o$ and a Cartesian grid along the square $\mathcal{S}$ defined by $|x - x_o| < \Delta, |y - y_o| < \Delta$, where Δ is chosen to be an integral multiple of dx. This interpolation ensures an accurate estimate of the elastic deformation energy of a single protein and preserves the symmetry of the protein in its immediate vicinity.

The lines connecting the grid points along $\mathcal{S}$ define n_θ angular grid points θ_i $(i = 1, \ldots, n_\theta)$, and $n_r + 1$ grid points within the interpolation region are defined by the polar coordinates $(r_{ij}, \theta_i) = (r_o + \delta r_{ij}/n_r, \theta_i)$, where r_o is the radius of the protein and the distance from the center of the protein to $\mathcal{S}$ along θ_i is $r_o + \delta r_i$ (e.g., for $\theta_i = 0$, $\delta r_i = \Delta - r_o$; for $\theta_i = \pi/4$, $\delta r_i = \Delta\sqrt{2} - r_o$). For a protein in the open or closed configuration, Δ was chosen such that $n_\theta = 320$ or 224, respectively.

The deformation energy determined using this numerical relaxation method is converged with respect to dx, Δ, and the overall dimensions of the bilayer ($18.5\,\text{nm} \times 37.1\,\text{nm}$), and reproduces the analytic results for a single protein given by Eq. 4.8. The elastic potentials were determined over the relevant range of channel separations from 0 to $\sim 8\,\text{nm}$ (measured from protein edge to protein edge), and for a range of bilayer tensions from 0 to $3.4\,k_BT/\text{nm}^2$.

We compared the analytic results of Section 4.3.3 with the results of a Markov Chain Monte Carlo (MCMC) simulation utilizing the Metropolis algorithm. The simulation was set up on a square grid with two proteins, each occupying a single lattice site. Periodic boundary conditions were applied so that a channel protein at position (x, y) is periodically repeated at positions $(x+na, y+ma)$ where (n, m) take all integer values and a is the length of one side of the lattice. The values of the numerically calculated, two channel potential were tabulated at tension values of $\chi = 0$ and $\chi = 1.3$; linear interpolation was used to calculate the potential at tension values and spatial points not explicitly tabulated. The total energy of the lattice configuration as a function of the distance r between their centers is:

$$E = \begin{cases} K, & r < R \\ H_{\text{int}}(s_1, s_2, r; \tau) + H_{\text{non}}(s_1, s_2; \tau), & r > R \end{cases}, \tag{4.26}$$

where H_{int} and H_{non} are defined as in Section 4.3.3 and R is the sum of the radii of the two channel proteins. The constant K is chosen to be very large to avoid a situation where the two proteins would "overlap" one another. At each step in the Markov chain a channel and an empty lattice site are chosen at random. The current configuration with energy E_0 is compared with two new configurations having energies E_{fi} with the selected protein moved to the empty lattice site: one configuration leaves the internal state (open or closed) of the channel unaltered (E_{f1}) and the other switches the protein's internal state (E_{f2}). The Metropolis acceptance formula is calculated for each of the new configurations: a new configuration is accepted and becomes the current configuration always if $E_{fi} \leq E_0$ and with probability $\mathrm{e}^{(\mathrm{E_{fi}} - \mathrm{E_0})}$ if $E_{fi} > E_0$. In a situation where both of the new configurations would be accepted by the Metropolis condition one of them is chosen at random to become the new configuration. The open probability P_{open}, average channel separation $\langle r \rangle$ and dimerization probability P_{dimer} are then calculated as averages over all the configurations in the Markov chain.

4.6 Supporting information

Accession numbers

The primary accession numbers (in parentheses) from the Protein Data Bank (http://www.pdb.org) are: MscL (2OAR; formerly 1MSL), gramicidin A ion channel (1GRM), bacterial potassium ion channel KscA (1F6G), and bovine rhodopsin (1GZM).

4.7 Acknowledgements

We would like to thank Doug Rees, Olaf Andersen, Pierre Sens, Sergei Sukharev, Nily Dan, Jennifer Stockdill, and Ned Wingreen for their thoughtful comments on the manuscript, Chris Gandhi for his input into possible experiments, Ben Freund for useful discussion, and the useful comments of our anonymous reviewers. RP acknowledges the support of the US National Science Foundation (NSF) award CMS-0301657. TU and RP acknowledge the support of the NSF CIMMS award ACI-0204932 and NIRT award CMS-0404031, as well as the US National Institutes of Health (NIH) Director's Pioneer Award. EP was supported by the US Department of Homeland Security Graduate Fellowship program and the NIH Director's Pioneer award. KCH was supported by NIH award A1K25 GM75000. Part of this work took place at the Kavli Institute for Theoretical Physics, Santa Barbara, California, and the Aspen Center for Physics, Aspen, Colorado.

Author Contributions

TU conceived and designed the experiments. TU, KCH, and EP performed the experiments. TU, KCH, EP, and RP analyzed the data and contributed to writing the paper.

Bibliography

[1] D Vella and L Mahadevan. The “Cheerios effect”. *Am J Phys*, 73(9):817–825, 2005.

[2] D L Hu and J W M Bush. Meniscus-climbing insects. *Nature*, 437(7059):733–736, 2005.

[3] O G Mouritsen and M Bloom. Models of lipid-protein interactions in membranes. *Annu Rev Biophys Biomol Struct*, 22:145–171, 1993.

[4] A G Lee. Lipid-protein interactions in biological membranes: a structural perspective. *Biochim Biophys Acta*, 1612:1–40, 2003.

[5] M O Jensen and O G Mouritsen. Lipids do influence protein function—the hydrophobic matching hypothesis revisited. *Biochim Biophys Acta*, 1666:205–226, 2004.

[6] D Bray, M D Levin, and C J Morton-Firth. Receptor clustering as a cellular mechanism to control sensitivity. *Nature*, 393:85–88, 1998.

[7] V Sourjik. Receptor clustering and signal processing in E. coli chemotaxis. *Trends Microbiol*, 12:569–576, 2004.

[8] K A Gibbs, D D Isaac, J Xu, R W Hendrix, T J Silhavy, and J A Theriot. Complex spatial distribution and dynamics of an abundant Escherichia coli outer membrane protein, LamB. *Mol Microbiol*, 53:1771–1783, 2004.

[9] A D Douglass and R D Vale. Single-molecule microscopy reveals plasma membrane microdomains created by protein-protein networks that exclude or trap signaling molecules in T cells. *Cell*, 121:937–950, 2005.

[10] V B Shenoy and L B Freund. Growth and shape stability of a biological membrane adhesion complex in the diffusion-mediated regime. *Proc Natl Acad Sci USA*, 102: 3213–3218, 2005.

[11] D M Engelman. Membranes are more mosaic than fluid. *Nature*, 438:578–580, 2005.

[12] M Goulian, P Pincus, and R Bruinsma. Long-range forces in heterogenous fluid membranes. *Europhys Lett*, 22:145–150, 1993.

[13] S I Sukharev, P Blount, B Martinac, and C Kung. Mechanosensitive channels of Escherichia coli: the MscL gene, protein, and activities. *Annu Rev Physiol*, 59:633–657, 1997.

[14] D E Clapham, L W Runnels, and C Strubing. The TRP ion channel family. *Nat Rev Neurosci*, 2:387–396, 2001.

[15] P H Barry and J W Lynch. Ligand-gated channels. *IEEE Trans Nanobioscience*, 4: 70–80, 2005.

[16] K Mitra, I Ubarretxena-Belandia, T Taguchi, G Warren, and D M Engelman. Modulation of the bilayer thickness of exocytic pathway membranes by membrane proteins rather than cholesterol. *Proc Natl Acad Sci USA*, 101:4083–4088, 2004.

[17] G Chang, R H Spencer, A T Lee, M T Barclay, and D C Rees. Structure of the MscL homolog from Mycobacterium tuberculosis: a gated mechanosensitive ion channel. *Science*, 282:2220–2226, 1998.

[18] C D Pivetti, M R Yen, S Miller, W Busch, Y H Tseng, I R Booth, and M H Saier, Jr. Two families of mechanosensitive channel proteins. *Microbiol Mol Biol Rev*, 67:66–85, 2003.

[19] S I Sukharev, W J Sigurdson, C Kung, and F Sachs. Energetic and spatial parameters for gating of the bacterial large conductance mechanosensitive channel, MscL. *J Gen Physiol*, 113:525–540, 1999.

[20] S Sukharev, M Betanzos, C S Chiang, and H R Guy. The gating mechanism of the large mechanosensitive channel MscL. *Nature*, 409:720–724, 2001.

[21] C S Chiang, A Anishkin, and S Sukharev. Gating of the large mechanosensitive channel in situ: estimation of the spatial scale of the transition from channel population responses. *Biophys J*, 86:2846–2861, 2004.

[22] E Perozo, A Kloda, D M Cortes, and B Martinac. Site-directed spin-labeling analysis of reconstituted MscL in the closed state. *J Gen Physiol*, 118:193–206, 2001.

[23] E Perozo, D M Cortes, P Sompornpisut, A Kloda, and B Martinac. Open channel structure of MscL and the gating mechanism of mechanosensitive channels. *Nature*, 418:942–948, 2002.

[24] H W Huang. Deformation free energy of bilayer membrane and its effect on gramicidin channel lifetime. *Biophys J*, 50:1061–1070, 1986.

[25] N Dan, P Pincus, and S Safran. Membrane-induced interactions between inclusions. *Langmuir*, 9:2768–2771, 1993.

[26] H Aranda-Espinoza, A Berman, N Dan, P Pincus, and S Safran. Interaction between inclusions embedded in membranes. *Biophys J*, 71:648–656, 1996.

[27] C Nielsen, M Goulian, and O S Andersen. Energetics of inclusion-induced bilayer deformations. *Biophys J*, 74:1966–1983, 1998.

[28] P Wiggins and R Phillips. Membrane-protein interactions in mechanosensitive channels. *Biophys J*, 88:880–902, 2005.

[29] V S Markin and F Sachs. Thermodynamics of mechanosensitivity. *Phys Biol*, 1:110–124, 2004.

[30] N Kahya, D Scherfeld, K Bacia, B Poolman, and P Schwille. Probing lipid mobility of raft-exhibiting model membranes by fluorescence correlation spectroscopy. *J Biol Chem*, 278:28109–28115, 2003.

[31] M K Doeven, J H Folgering, V Krasnikov, E R Geertsma, G van den Bogaart, and B Poolman. Distribution, lateral mobility and function of membrane proteins incorporated into giant unilamellar vesicles. *Biophys J*, 88:1134–1142, 2005.

[32] Y Gambin, R Lopez-Esparza, M Reffay, E Sierecki, N S Gov, M Genest, R S Hodges, and W Urbach. Lateral mobility of proteins in liquid membranes revisited. *Proc Natl Acad Sci USA*, 103:2098–2102, 2006.

[33] G Guigas and M Weiss. Size-dependent diffusion of membrane inclusions. *Biophys J*, 91:2393–2398, 2006.

[34] G Shapovalov and H A Lester. Gating transitions in bacterial ion channels measured at 3 μs resolution. *J Gen Physiol*, 124:151–161, 2004.

[35] T A Harroun, W T Heller, T M Weiss, L Yang, and H W Huang. Theoretical analysis of hydrophobic matching and membrane-mediated interactions in lipid bilayers containing gramicidin. *Biophys J*, 76:3176–3185, 1999.

[36] R S Cantor. Lipid composition and the lateral pressure profile in bilayers. *Biophys J*, 76:2625–2639, 1999.

[37] W Helfrich. Elastic properties of lipid bilayers: theory and possible experiments. *Z Naturforsch*, 28:693–703, 1973.

[38] W Rawicz, K C Olbrich, T McIntosh, D Needham, and E Evans. Effect of chain length and unsaturation on elasticity of lipid bilayers. *Biophys J*, 79:328–339, 2000.

[39] R E Tosh and P J Collings. High pressure volumetric measurements in dipalmitoylphosphatidylcholine bilayers. *Biochim Biophys Acta*, 859:10–14, 1986.

[40] H Seemann and R Winter. Volumetric properties, compressibilities and volume fluctuations in phospholipid-cholesterol bilayers. *Z Phys Chem*, 217:831–846, 2003.

[41] N Dan and S A Safran. Effect of lipid characteristics on the structure of transmembrane proteins. *Biophys J*, 75:1410–1414, 1998.

[42] G Niggemann, M Kummrow, and W Helfrich. The bending rigidity of phosphatidylcholine bilayers: Dependences on experimental method, sample cell sealing and temperature. *J Phys II France*, 5:413–425, 1995.

[43] G B Arfken and H J Weber. *Mathematical Methods for Physicists*. Harcourt Academic Press, 5th edition, 2001.

[44] K Yoshimura, A Batiza, M Schroeder, P Blount, and C Kung. Hydrophilicity of a single residue within MscL correlates with increased channel mechanosensitivity. *Biophys J*, 77:1960–1972, 1999.

[45] N R Stokes, H D Murray, C Subramaniam, R L Gourse, P Louis, W Bartlett, S Miller, and I R Booth. A role for mechanosensitive channels in survival of stationary phase:

regulation of channel expression by RpoS. *Proc Natl Acad Sci USA*, 100:15959–15964, 2003.

[46] J B Fournier. Microscopic membrane elasticity and interactions among membrane inclusions: Interplay between the shape, dilation, tilt and tilt-difference modes. *Eur Phys J B*, 11:261–272, 1999.

[47] P G Dommersnes and J B Fournier. N-body study of anisotropic membrane inclusions: Membrane mediated interactions and ordered aggregation. *Eur Phys J B*, 12:9–12, 1999.

[48] D Bartolo and J B Fournier. Elastic interaction between "hard" or "soft" pointwise inclusions on biological membranes. *Eur Phys J E*, 11:141–146, 2003.

[49] T Chou, K S Kim, and G Oster. Statistical thermodynamics of membrane bending-mediated protein-protein attractions. *Biophys J*, 80:1075–1087, 2001.

[50] A R Evans, M S Turner, and P Sens. Interactions between proteins bound to biomembranes. *Phys Rev E*, 67:041907, 2003.

[51] C E Morris and U Homann. Cell surface area regulation and membrane tension. *J Membr Biol*, 179:79–102, 2001.

[52] T Sintes and A Baumgartner. Protein attraction in membranes induced by lipid fluctuations. *Biophys J*, 73:2251–2259, 1997.

[53] J M Park and T C Lubensky. Interactions between membrane inclusions on fluctuating membranes. *J Phys I France*, 6:1217–1235, 1996.

[54] D Boal. *Mechanics of the Cell.* Cambridge University Press, 1st edition, 2002.

[55] D P Siegel and M M Kozlov. The Gaussian curvature elastic modulus of n-monomethylated dioleoylphosphatidylethanolamine: relevance to membrane fusion and lipid phase behavior. *Biophys J*, 87:366–374, 2004.

[56] J R Elliott, D Needham, J P Dilger, and D A Haydon. The effects of bilayer thickness and tension on gramicidin single-channel lifetime. *Biochim Biophys Acta*, 735:95–103, 1983.

[57] E Perozo, A Kloda, D M Cortes, and B Martinac. Physical principles underlying the transduction of bilayer deformation forces during mechanosensitive channel gating. *Nat Struct Biol*, 9:696–703, 2002.

[58] A Karlin and M H Akabas. Substituted-cysteine accessibility method. *Methods Enzymol*, 293:123–145, 1998.

[59] G Wilson and A Karlin. Acetylcholine receptor channel structure in the resting, open, and desensitized states probed with the substituted-cysteine-accessibility method. *Proc Natl Acad Sci USA*, 98:1241–1248, 2001.

[60] H Hastrup, N Sen, and J A Javitch. The human dopamine transporter forms a tetramer in the plasma membrane: cross-linking of a cysteine in the fourth transmembrane segment is sensitive to cocaine analogs. *J Biol Chem*, 278:45045–45048, 2003.

[61] M Bogdanov, W Zhang, J Xie, and W Dowhan. Transmembrane protein topology mapping by the substituted cysteine accessibility method (SCAM^{TM}): application to lipid-specific membrane protein topogenesis. *Methods*, 36:148–171, 2005.

[62] T Haselgrubler, A Amerstorfer, H Schindler, and H J Gruber. Synthesis and applications of a new poly(ethylene glycol) derivative for the crosslinking of amines with thiols. *Bioconjug Chem*, 6:242–248, 1995.

[63] R O Blaustein, P A Cole, C Williams, and C Miller. Tethered blockers as molecular “tape measures” for a voltage-gated K+ channel. *Nat Struct Biol*, 7:309–311, 2000.

[64] S I Sukharev, M J Schroeder, and D R McCaslin. Stoichiometry of the large conductance bacterial mechanosensitive channel of E. coli. A biochemical study. *J Membr Biol*, 171:183–193, 1999.

[65] R L Goforth, A K Chi, D V Greathouse, L L Providence, R E Koeppe, II, and OS Andersen. Hydrophobic coupling of lipid bilayer energetics to channel function. *J Gen Physiol*, 121:477–493, 2003.

[66] J A Lundbaek and O S Andersen. Spring constants for channel-induced lipid bilayer deformations. *Biophys J*, 76:889–895, 1999.

[67] M B Partenskii, G V Miloshevsky, and P C Jordan. Stabilization of ion channels due to membrane-mediated elastic interaction. *J Chem Phys*, 118:10306–10312, 2003.

[68] A V Botelho, T Huber, T P Sakmar, and M F Brown. Curvature and hydrophobic forces drive oligomerization and modulate activity of rhodopsin in membranes. *Biophys J*, 91:4464–4477, 2006.

[69] M L Molina, F N Barrera, A M Fernandez, J A Poveda, M L Renart, J A Encinar, G Riquelme, and J M Gonzalez-Ros. Clustering and coupled gating modulate the activity in KcsA, a potassium channel model. *J Biol Chem*, 281:18837–18848, 2006.

[70] M L Skoge, R G Endres, and N S Wingreen. Receptor-receptor coupling in bacterial chemotaxis: evidence for strongly coupled clusters. *Biophys J*, 90:4317–4326, 2006.

Appendix A

Reduced amino acid alphabets can improve the sensitivity and selectivity of pairwise sequence alignments

A.1 Performance of substitution matrices vs. family size

In order to tease out the reason for the improved performance of some reduced alphabets over the full alphabet, we conducted a study of how the additional performance in mean pooled precision, area under the ROC curve, and recall at 0.01 EPQ was distributed as a function of family size. The results are shown in Fig. A.1 through Fig. A.5. In the case of mean pooled precision and area under the ROC curve the three larger alphabets perform comparably as a function of family size; in mean pooled precision GMBR4 does markedly worse. In recall at 0.01 EPQ under all three normalizations it is clear that the additional gains made by GBMR4 are accumulated in slight increments among the smaller families, which tend to be more diverse.

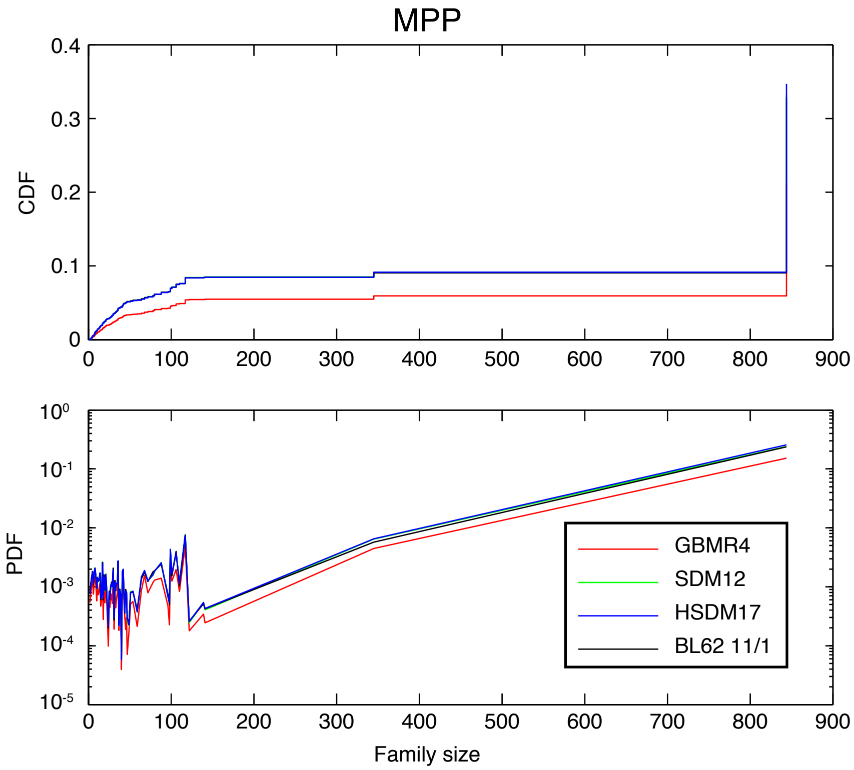


Figure A.1: **Performance of GBMR4, SDM12, HSDM17, and BL62 11/1 in mean pooled precision as a function of family size.** The distribution over family size is shown on the bottom and the integrated distribution is shown above. The three larger alphabets perform comparably to each other; the smaller GBMR4 alphabet performs more poorly.

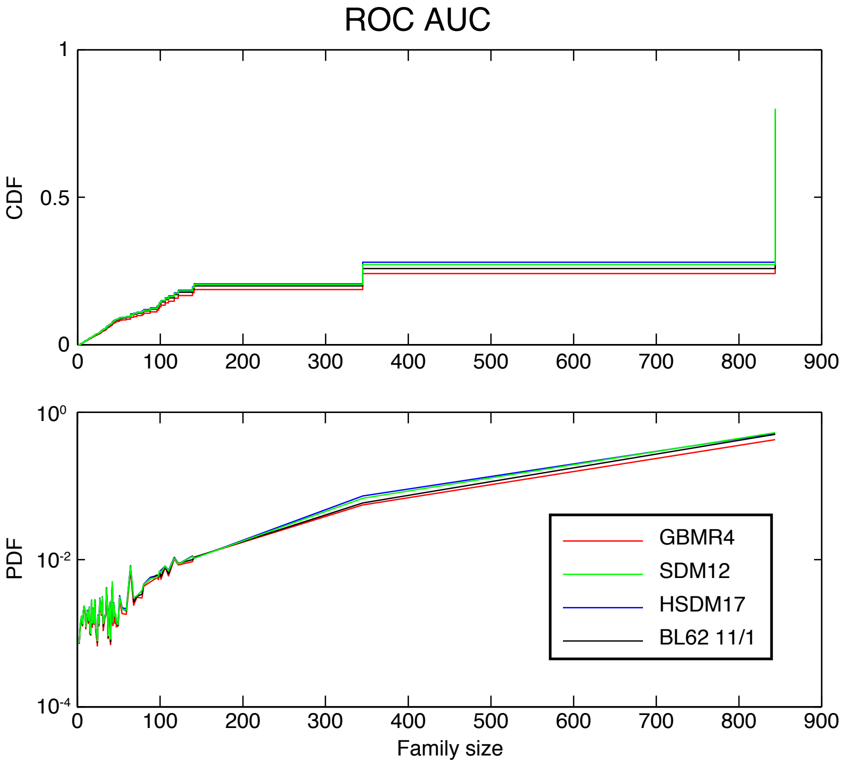


Figure A.2: **Performance of GBMR4, SDM12, HSDM17, and BL62 11/1 in area under the ROC curve as a function of family size.** The distribution over family size is shown on the bottom and the integrated distribution is shown above. In this metric all four alphabets perform comparably across most of the range of family sizes.

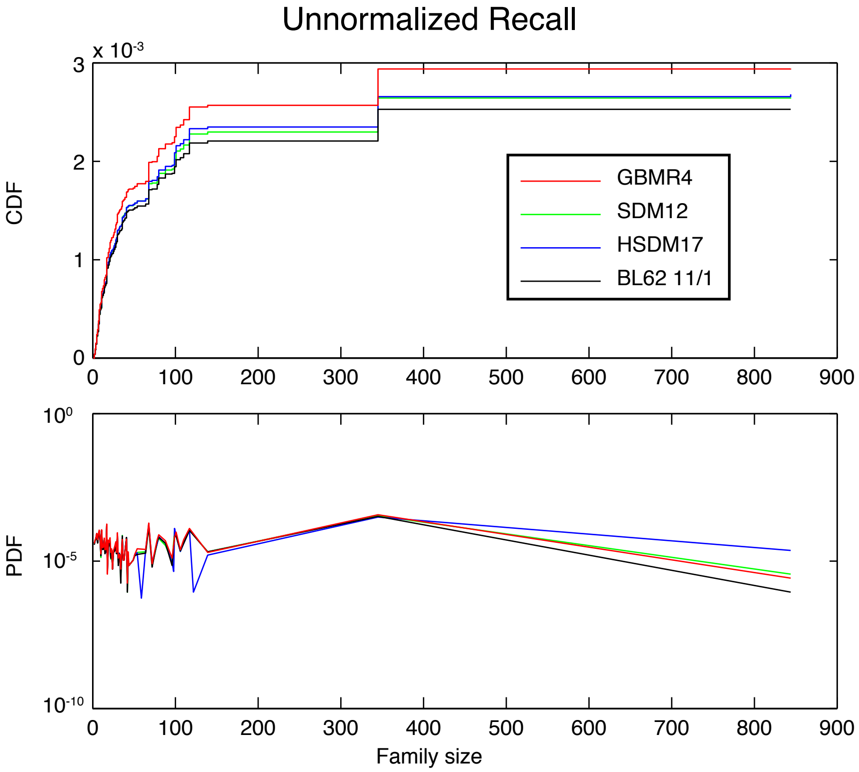


Figure A.3: **Performance of GBMR4, SDM12, HSDM17, and BL62 11/1 in unnormalized recall at 0.01 EPQ as a function of family size.** The distribution over family size is shown on the bottom and the integrated distribution is shown above. The superior performance of GBMR4 is gained chiefly at smaller family sizes.

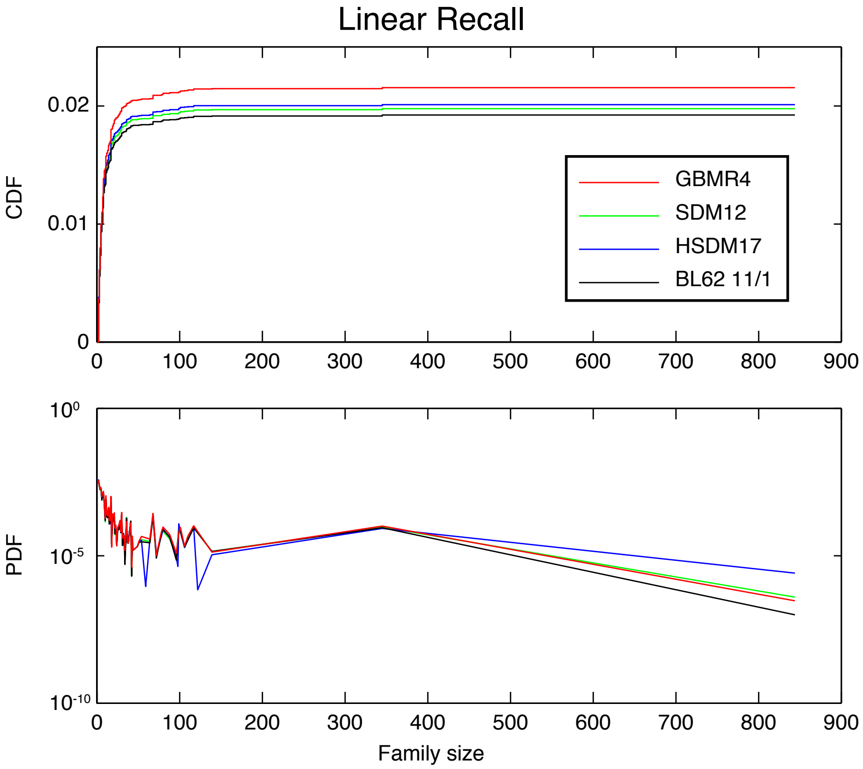


Figure A.4: **Performance of GBMR4, SDM12, HSDM17, and BL62 11/1 in linearly normalized recall at 0.01 EPQ as a function of family size.** The distribution over family size is shown on the bottom and the integrated distribution is shown above. The superior performance of GBMR4 is gained chiefly at smaller family sizes.

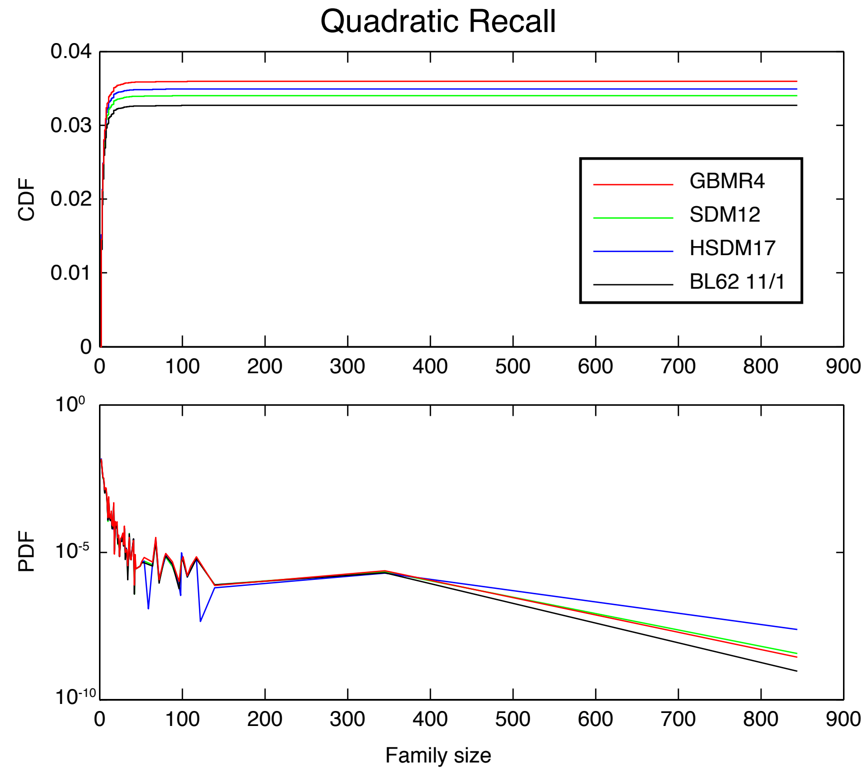


Figure A.5: **Performance of GBMR4, SDM12, HSDM17, and BL62 11/1 in quadratically normalized recall at 0.01 EPQ as a function of family size.** The distribution over family size is shown on the bottom and the integrated distribution is shown above. The superior performance of GBMR4 is gained chiefly at smaller family sizes.

A.2 Table of all substitution matrices tested

Table A.1: Results for all alphabets and matrices tested

					Recall at 0.01 EPQ		
Scheme	**Letters**	**AUC**	**MPP**	**Align**	**None**	**Linear**	**Quadratic**
AB	2	0.665	0.159	0.04844	0.0026	0.020	0.034
AB	3	0.674	0.178	0.04968	0.0025	0.019	0.033
AB	4	0.682	0.194	0.05828	0.0026	0.020	0.034
AB	5	0.702	0.206	0.06143	0.0025	0.019	0.033
AB	6	0.719	0.210	0.05877	0.0024	0.019	0.034
AB	7	0.731	0.234	0.06827	0.0025	0.019	0.033
AB	8	0.721	0.252	0.07310	0.0026	0.020	0.034
AB	9	0.728	0.294	0.07554	0.0026	0.019	0.033
AB	10	0.740	0.297	0.07509	0.0026	0.020	0.034
AB	11	0.749	0.310	0.07690	0.0027	0.020	0.034
AB	12	0.749	0.313	0.07736	0.0027	0.020	0.034
AB	13	0.750	0.312	0.07610	0.0027	0.020	0.034
AB	14	0.755	0.314	0.07599	0.0026	0.020	0.034
AB	15	0.753	0.319	0.07603	0.0026	0.020	0.033
AB	16	0.754	0.320	0.07648	0.0026	0.019	0.033
AB	17	0.752	0.322	0.07702	0.0026	0.019	0.033
AB	18	0.756	0.326	0.07876	0.0026	0.020	0.034
AB	19	0.757	0.323	0.07828	0.0026	0.019	0.033
BL50 11/1	20	0.779	0.346	0.08476	0.0027	0.021	0.035
BL50 12/2	20	0.762	0.334	0.08111	0.0026	0.020	0.034
BL62 11/1	20	0.759	0.329	0.08322	0.0025	0.019	0.033
CB	2	0.705	0.188	0.06774	0.0029	0.021	0.035
CB	5	0.674	0.191	0.05904	0.0024	0.018	0.031
DSSP	2	0.679	0.154	0.05393	0.0044	0.019	0.033
DSSP	3	0.731	0.209	0.07479	0.0025	0.020	0.034
DSSP	4	0.709	0.222	0.07996	0.0027	0.020	0.033
DSSP	5	0.723	0.219	0.07736	0.0026	0.019	0.032
DSSP	6	0.729	0.230	0.07913	0.0025	0.019	0.033
DSSP	7	0.738	0.246	0.08042	0.0025	0.019	0.032
DSSP	8	0.730	0.233	0.07572	0.0024	0.018	0.031
DSSP	9	0.731	0.244	0.07631	0.0024	0.019	0.033
DSSP	10	0.733	0.253	0.07740	0.0025	0.019	0.032
DSSP	11	0.757	0.282	0.07829	0.0026	0.019	0.033
DSSP	12	0.759	0.287	0.07942	0.0026	0.019	0.033
DSSP	13	0.758	0.290	0.08084	0.0026	0.019	0.033
DSSP	14	0.768	0.297	0.08252	0.0026	0.020	0.034
GBMR	2	0.605	0.091	0.02423	0.0029	0.018	0.032
GBMR	3	0.614	0.122	0.03261	0.0025	0.020	0.035

Table A.1: Results for all alphabets and matrices tested (continued)

					Recall at 0.01 EPQ		
Scheme	**Letters**	**AUC**	**MPP**	**Align**	**None**	**Linear**	**Quadratic**
GBMR	4	0.667	0.212	0.07676	0.0029	0.022	0.036
GBMR	5	0.678	0.220	0.07549	0.0027	0.020	0.033
GBMR	6	0.691	0.196	0.06778	0.0097	0.019	0.031
GBMR	7	0.709	0.202	0.06784	0.0089	0.019	0.032
GBMR	8	0.716	0.205	0.06857	0.0085	0.020	0.032
GBMR	9	0.719	0.206	0.06871	0.0067	0.018	0.030
GBMR	10	0.726	0.217	0.07052	0.0038	0.020	0.033
GBMR	11	0.735	0.240	0.07264	0.0028	0.019	0.032
GBMR	12	0.738	0.240	0.07098	0.0027	0.020	0.035
GBMR	13	0.742	0.250	0.07096	0.0026	0.020	0.034
GBMR	14	0.742	0.249	0.07022	0.0026	0.020	0.034
HSDM	2	0.725	0.199	0.07214	0.0029	0.021	0.036
HSDM	3	0.732	0.203	0.07266	0.0026	0.020	0.034
HSDM	4	0.726	0.210	0.07306	0.0026	0.019	0.034
HSDM	5	0.751	0.234	0.07562	0.0027	0.019	0.034
HSDM	6	0.751	0.262	0.07827	0.0027	0.020	0.035
HSDM	7	0.751	0.279	0.08154	0.0028	0.020	0.033
HSDM	8	0.759	0.295	0.08235	0.0027	0.020	0.034
HSDM	9	0.784	0.319	0.08714	0.0028	0.020	0.033
HSDM	10	0.776	0.325	0.08691	0.0027	0.020	0.034
HSDM	12	0.771	0.323	0.08676	0.0026	0.020	0.035
HSDM	14	0.783	0.341	0.08862	0.0027	0.020	0.035
HSDM	15	0.783	0.343	0.08857	0.0026	0.020	0.035
HSDM	16	0.789	0.344	0.08849	0.0026	0.020	0.035
HSDM	17	0.796	0.347	0.08887	0.0027	0.020	0.035
HSDM	20	0.791	0.345	0.08882	0.0026	0.020	0.036
JO20	20	0.725	0.274	0.05900	0.0024	0.019	0.033
LR	10	0.719	0.280	0.07019	0.0026	0.020	0.034
LW-I	2	0.705	0.188	0.06774	0.0029	0.021	0.035
LW-I	3	0.720	0.203	0.06403	0.0027	0.020	0.034
LW-I	4	0.697	0.232	0.07038	0.0026	0.019	0.032
LW-I	5	0.701	0.227	0.06388	0.0027	0.020	0.034
LW-I	6	0.695	0.241	0.06369	0.0027	0.020	0.034
LW-I	7	0.688	0.240	0.06681	0.0026	0.020	0.033
LW-I	8	0.737	0.286	0.07320	0.0026	0.020	0.034
LW-I	9	0.740	0.290	0.07389	0.0026	0.020	0.035
LW-I	10	0.728	0.292	0.07218	0.0025	0.019	0.033
LW-I	11	0.735	0.303	0.07579	0.0026	0.020	0.035
LW-I	12	0.740	0.303	0.07605	0.0026	0.020	0.034
LW-I	13	0.754	0.310	0.07662	0.0026	0.020	0.034

Table A.1: Results for all alphabets and matrices tested (continued)

					Recall at 0.01 EPQ		
Scheme	**Letters**	**AUC**	**MPP**	**Align**	**None**	**Linear**	**Quadratic**
LW-I	14	0.756	0.307	0.07569	0.0025	0.020	0.034
LW-I	15	0.757	0.308	0.07593	0.0025	0.020	0.034
LW-I	16	0.754	0.314	0.07599	0.0026	0.020	0.034
LW-I	17	0.752	0.317	0.07619	0.0026	0.020	0.034
LW-I	18	0.753	0.318	0.07663	0.0026	0.020	0.034
LW-I	19	0.755	0.321	0.07693	0.0026	0.020	0.034
LW-NI	2	0.705	0.188	0.06774	0.0029	0.021	0.035
LW-NI	3	0.720	0.203	0.06403	0.0027	0.020	0.034
LW-NI	4	0.723	0.224	0.06361	0.0025	0.019	0.032
LW-NI	5	0.702	0.229	0.06417	0.0026	0.019	0.032
LW-NI	6	0.707	0.243	0.06453	0.0026	0.019	0.033
LW-NI	7	0.698	0.245	0.06795	0.0026	0.020	0.033
LW-NI	8	0.698	0.244	0.06509	0.0025	0.020	0.034
LW-NI	9	0.696	0.249	0.06569	0.0025	0.019	0.034
LW-NI	10	0.706	0.263	0.06816	0.0025	0.020	0.034
LW-NI	11	0.739	0.292	0.07406	0.0025	0.020	0.034
LW-NI	12	0.740	0.303	0.07605	0.0026	0.020	0.034
LW-NI	13	0.754	0.310	0.07662	0.0026	0.020	0.034
LW-NI	14	0.756	0.312	0.07688	0.0027	0.020	0.034
LW-NI	15	0.757	0.308	0.07593	0.0025	0.020	0.034
LW-NI	16	0.754	0.314	0.07599	0.0026	0.020	0.034
LW-NI	17	0.752	0.317	0.07619	0.0026	0.020	0.034
LW-NI	18	0.753	0.318	0.07663	0.0026	0.020	0.034
LW-NI	19	0.755	0.321	0.07693	0.0026	0.020	0.034
LZ-BL	2	0.690	0.194	0.06969	0.0026	0.020	0.033
LZ-BL	3	0.719	0.217	0.07751	0.0026	0.019	0.033
LZ-BL	4	0.736	0.242	0.08056	0.0027	0.020	0.033
LZ-BL	5	0.734	0.282	0.08134	0.0028	0.020	0.034
LZ-BL	6	0.742	0.299	0.08337	0.0028	0.020	0.035
LZ-BL	7	0.744	0.297	0.08129	0.0027	0.021	0.036
LZ-BL	8	0.736	0.299	0.08115	0.0026	0.020	0.036
LZ-BL	9	0.744	0.300	0.08092	0.0026	0.020	0.034
LZ-BL	10	0.749	0.327	0.08417	0.0026	0.020	0.034
LZ-BL	11	0.750	0.325	0.08184	0.0026	0.020	0.035
LZ-BL	12	0.769	0.328	0.08326	0.0025	0.019	0.033
LZ-BL	13	0.774	0.331	0.08380	0.0026	0.020	0.034
LZ-BL	14	0.774	0.334	0.08391	0.0025	0.019	0.033
LZ-BL	15	0.777	0.339	0.08413	0.0026	0.020	0.034
LZ-BL	16	0.783	0.346	0.08451	0.0027	0.020	0.034
LZ-MJ	2	0.700	0.165	0.05816	0.0026	0.019	0.033

Table A.1: Results for all alphabets and matrices tested (continued)

					Recall at 0.01 EPQ		
Scheme	**Letters**	**AUC**	**MPP**	**Align**	**None**	**Linear**	**Quadratic**
LZ-MJ	3	0.666	0.173	0.05389	0.0023	0.018	0.032
LZ-MJ	4	0.722	0.203	0.06992	0.0024	0.019	0.032
LZ-MJ	5	0.779	0.221	0.07165	0.0022	0.017	0.031
LZ-MJ	6	0.793	0.220	0.07124	0.0022	0.018	0.031
LZ-MJ	7	0.770	0.246	0.07434	0.0023	0.018	0.030
LZ-MJ	8	0.750	0.250	0.07749	0.0025	0.019	0.032
LZ-MJ	9	0.757	0.261	0.08010	0.0024	0.018	0.031
LZ-MJ	10	0.764	0.266	0.08065	0.0024	0.018	0.031
LZ-MJ	11	0.759	0.265	0.07871	0.0023	0.018	0.030
LZ-MJ	12	0.782	0.279	0.07966	0.0023	0.018	0.031
LZ-MJ	13	0.782	0.285	0.08017	0.0024	0.018	0.031
LZ-MJ	14	0.783	0.285	0.08082	0.0023	0.018	0.031
LZ-MJ	15	0.773	0.311	0.08207	0.0024	0.019	0.032
LZ-MJ	16	0.773	0.312	0.08259	0.0024	0.019	0.032
ML	4	0.693	0.236	0.07146	0.0026	0.019	0.032
ML	8	0.753	0.294	0.07733	0.0027	0.020	0.035
ML	10	0.757	0.304	0.08087	0.0027	0.020	0.033
ML	15	0.762	0.331	0.08063	0.0027	0.020	0.034
MM	5	0.691	0.210	0.06894	0.0023	0.017	0.030
MS	6	0.715	0.232	0.06853	0.0027	0.020	0.035
SDM	2	0.740	0.145	0.06695	0.0022	0.013	0.020
SDM	3	0.775	0.184	0.07099	0.0022	0.016	0.026
SDM	4	0.767	0.212	0.07450	0.0026	0.015	0.024
SDM	6	0.766	0.232	0.07591	0.0030	0.016	0.029
SDM	7	0.773	0.255	0.08029	0.0028	0.020	0.034
SDM	8	0.759	0.268	0.07952	0.0028	0.021	0.035
SDM	10	0.764	0.310	0.08642	0.0026	0.019	0.031
SDM	11	0.800	0.329	0.08686	0.0027	0.019	0.033
SDM	12	0.801	0.332	0.08670	0.0026	0.020	0.034
SDM	13	0.800	0.335	0.08675	0.0027	0.020	0.034
SDM	14	0.795	0.332	0.08603	0.0027	0.020	0.034
SDM	20	0.770	0.331	0.08594	0.0026	0.019	0.033
TD	2	0.678	0.162	0.05673	0.0027	0.021	0.035
TD	3	0.679	0.162	0.05631	0.0025	0.020	0.034
TD	4	0.704	0.175	0.06090	0.0024	0.019	0.031
TD	5	0.718	0.185	0.06316	0.0023	0.018	0.032
TD	6	0.768	0.224	0.07772	0.0023	0.018	0.031
TD	7	0.740	0.237	0.08096	0.0023	0.018	0.031
TD	8	0.748	0.265	0.08159	0.0024	0.018	0.031
TD	9	0.737	0.278	0.08153	0.0025	0.019	0.032

Table A.1: Results for all alphabets and matrices tested (continued)

					Recall at 0.01 EPQ		
Scheme	**Letters**	**AUC**	**MPP**	**Align**	**None**	**Linear**	**Quadratic**
TD	10	0.743	0.283	0.08033	0.0025	0.019	0.032
TD	14	0.743	0.306	0.07984	0.0026	0.020	0.034
WW	5	0.709	0.219	0.07172	0.0026	0.019	0.033

Appendix B

Membrane shape as a reporter for applied forces

B.1 Vesicle-induced force artifacts

We checked that the size of the fluctations (the spring constant) remains nearly constant during vesicle extension experiments, implying that the presence of the vesicle does not affect the trap stiffness and the trap remains close to the linear regime. The mean and variance of the voltage are shown in Fig. B.1.

Although the vesicle does not significantly affect the trap stiffness, it does affect the DC offset of the PSD. If the trap is zeroed without contact with the vesicle, the presence of the vesicle in the beam scatters light, leading to an offset voltage at the PSD. It was therefore necessary to assume a zero on bead contact. The beads were bound to the membrane by squeezing the vesicle between the trapped beads.

We also discovered that changes in the vesicle conformation in proximity to the bead could also lead to force artifacts. The evidence for this effect is discussed in Fig. B.2. Data sets where the vesicle tethered on the pole opposite from the force sensor did not appear to show these anomalous forces, presumably because the local vesicle conformation was not appreciably changed by deformation.

We also note that the trapping force is susceptible to drift in the subpiconewton regime. Due to the competing effects of drift and systematic uncertainties related to membrane-induced beam scattering, the most meaningful comparisons between the conformational force and the trapping forces are differences between conformations separated by a few steps in the extension series, since the membrane-induced scattering is similar and the

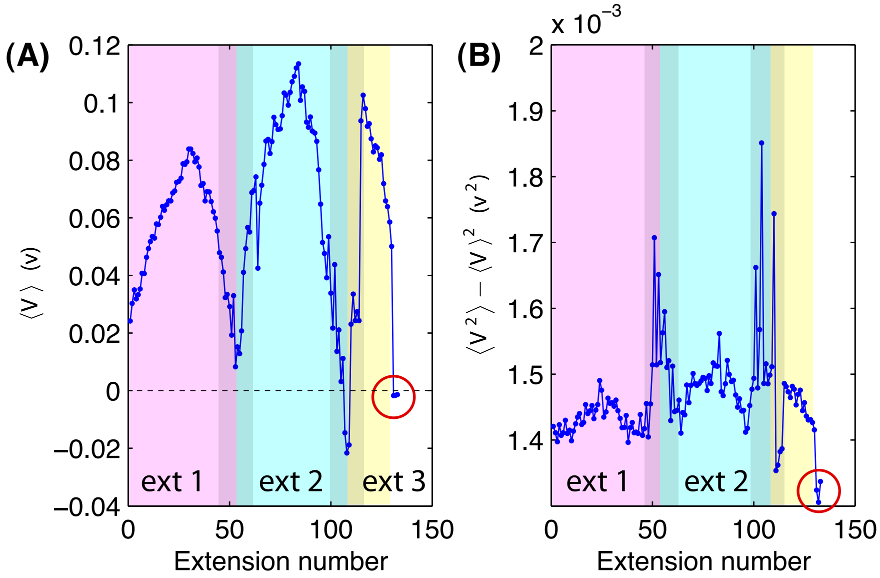


Figure B.1: **Issues in determining the zero of the PSD force measurements. Panel A.** The mean PSD voltage plotted vs. step number for a data set where a membrane tether was extended and retracted three times. The trap deflection (and force) are proportional to the PSD voltage. The last three circled points are the zeros for the experiment with beads unattached to the vesicle. The initial points on the trace correspond to beads bound to the vesicle, but without extension. Binding the vesicle causes an anomalous force due to scattering of light. See discussion in the caption of Fig. B.2. **Panel B.** The voltage variance vs. step number. The voltage variance is inversely proportional to the trap stiffness. Note first that the voltage variance is stable relative to its magnitude throughout the extension. (The data has been plotted to accentuate variations by not including zero on the y axis.) As the force is increased in the first tether extraction, the voltage variance is increased slightly, corresponding to a slight reduction in the trap stiffness, as would be expected due to the large-deflection non-linearity in the optical trap. Two larger features are visible in the voltage variance at step number 65 and 110. These features are the result of changes in the local structure of the vesicle surrounding the bead. See discussion in the caption of Fig. B.2.

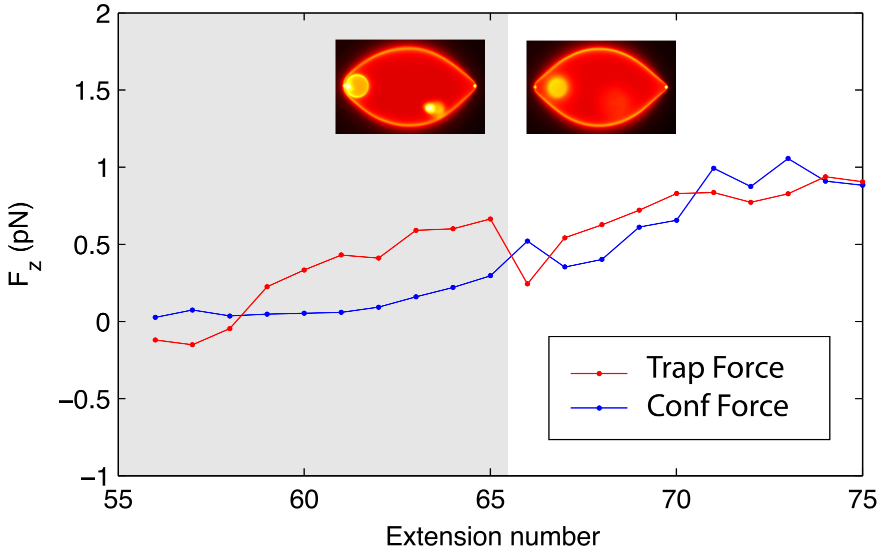


Figure B.2: **A limitation of measuring trap forces in the presence of a vesicle.** The vesicles themselves scatter the trapping beam, which can result in the detection of anomalous forces. In the figure above, we depict the most compelling evidence for this effect. At the beginning of this data set, an internal vesicle is bound to the lipid aggregate surrounding the bead. Between step number 65 and 66, the vesicle is released and diffuses into the body of the outer vesicle. This release causes a precipitous drop in the detected trap force (red curve). For reference purposes, we have also plotted the force calculated from the membrane conformation (blue curve). A similar offset in force is observed upon attachment of the force detection bead to the vesicle. Rigorously, this implies that at best we can detect relative forces, not absolute forces. Similar anomalies occur when the vesicle becomes tethered at the detection bead, due to a dramatic change in local vesicle structure. The most reliable force traces resulted from vesicles tethering at the extension bead. In spite of these complications, the trap stiffness was not significantly affected by the vesicle.

drift between adjacent steps is smaller than between measurements taken ten to twenty minutes apart. As a practical matter of reducing the influence of drift and low-tension vesicle structure induced force anomalies, the DC offset was calculated by minimizing the difference between the trapping force and conformational force at high force.

B.2 Derivation of the local force density

B.2.1 The manifold

In a coordinate system (s_1, s_2) on a two-manifold M the canonical energy of a fluid lipid bilayer membrane is (valid for small displacements from equilibrium) [1–3]:

$$E = \int_{\mathcal{M}} \mathrm{d}s^1 \mathrm{d}s^2 \sqrt{g} \left[\alpha + \frac{1}{2} k_C \left(S - C_0 \right)^2 \right] . \tag{B.1}$$

In this expression for the energy we omit the Gaussian curvature term, since it is purely topological for closed membranes and does not contribute to the local membrane forces. The Lagrange multiplier α is introduced to constrain the area of the membrane to be constant. The other factors appearing are k_C, the bending modulus of the membrane; S, the sum of the principal curvatures; and C_0, the spontaneous curvature. It is useful to embed the membrane in a three-dimensional Cartesian space; vectors in this space will be denoted X. Greek indices will be used in the Cartesian embedding space, and Latin indices for the manifold. Note that we don't have to be careful about raising and lowering Greek indices since the metric for the Cartesian space is just the identity tensor.

B.2.2 The metric

The metric on the membrane in terms of the vectors X^α of the embedding space is:

$$\begin{aligned} g_{ij} &= \delta_{\alpha\beta} \frac{\partial X^\alpha}{\partial s^i} \frac{\partial X^\beta}{\partial s^j} && \text{(B.2)} \\ &= \partial_i X^\alpha \, \partial_j X^\alpha . && \text{(B.3)} \end{aligned}$$

B.2.3 The tangent space

A right-handed coordinate system can be formed for the tangent space of the membrane by using the tangent vectors $\partial_i X^\alpha$. These two vectors span the subspace of the tangent space parallel to the membrane; we can form the normal vector by taking the antisymmetric product of the tangent vectors:

$$\epsilon_{ij} n^{\alpha} = A \, \epsilon_{\alpha\beta\gamma} \, \partial_i X^{\beta} \, \partial_j X^{\gamma}. \tag{B.4}$$

The ϵ_{ij} is present since we are taking the right-handed set of basis vectors to be $\{\partial_1 X^{\alpha}, \partial_2 X^{\alpha}, n^{\alpha}\}$. Now we need to work out the normalization factor A of n^{α}:

$$\begin{aligned}
1 &= n^{\alpha} n^{\alpha} \\
&= A^2 \, \epsilon_{\alpha\beta\gamma} \, \epsilon_{\alpha\delta\xi} \, \partial_1 X^{\beta} \, \partial_2 X^{\gamma} \, \partial_1 X^{\delta} \, \partial_2 X^{\xi} \\
&= A^2 \, (\delta_{\beta\delta}\delta_{\gamma\xi} - \delta_{\beta\xi}\delta_{\gamma\delta}) \, \partial_1 X^{\beta} \, \partial_2 X^{\gamma} \, \partial_1 X^{\delta} \, \partial_2 X^{\xi} \\
&= A^2 \, (g_{11} g_{22} - g_{12} g_{21}) \\
A &= \frac{1}{\sqrt{g}}.
\end{aligned}$$

So the normal vector in the membrane tangent space is:

$$n^{\alpha} = \frac{1}{\sqrt{g}} \, \epsilon_{\alpha\beta\gamma} \, \partial_1 X^{\beta} \, \partial_2 X^{\gamma}. \tag{B.5}$$

B.2.4 The shape operator

The shape operator (also called the second fundamental form) says something about the curvature of the membrane. Given a normal vector and a tangent direction, the shape operator tells one how the normal vector changes in that direction. Following are some identities that will be needed in the calculation:

$$S_{ij} = - \, \partial_i X^{\alpha} \, \partial_j n^{\alpha} \tag{B.6}$$

$$S_{ij} = n^{\alpha} \, \nabla_i \nabla_j X^{\alpha} \tag{B.7}$$

$$\nabla^i S_{ij} = \nabla_j S. \tag{B.8}$$

Let us also define what we mean by S and K which are the trace and determinant of the shape operator:

$$\begin{aligned} S &= \text{tr } \mathsf{S} && \text{(B.9)} \\ &= g^{ij} S_{ij} && \text{(B.10)} \\ K &= \det \mathsf{S} && \text{(B.11)} \\ &= \frac{1}{2} \epsilon^{ij} \epsilon^{lm} S_{il} S_{jm}. && \text{(B.12)} \end{aligned}$$

Another useful identity that we will use is:

$$K = \frac{1}{2} \left(S^2 + \text{tr } \mathsf{S}^2 \right), \tag{B.13}$$

where tr $\mathsf{S}^2 = S^{ij} S_{ij}$.

B.2.5 Finding the normal force

We will find the forces needed to maintain the membrane in mechanical equilibrium with a virtual work principle. Before launching into the variation of the whole membrane energy expression it will be useful to derive a few preliminary results.

B.2.5.1 Preliminary variations

When we vary the membrane energy, the δ operator will find two terms to operate on: $\sqrt{g}$ and $(S - C_0)^2$. Why only these terms? First, there is no dependence on the embedding coordinates X^α in C_0 or k_C. Also, there is also no dependence in α on the coordinates; it is a property of the membrane lipids (total surface area) and is independent of the embedding. First let us work out $\delta \left(\sqrt{g} \right)$ in a locally diagonal basis, so that $g = g_{11} g_{22}$:

$$\begin{aligned} \delta \left(\sqrt{g} \right) &= \delta \sqrt{g_{11} g_{22}} \\ &= \frac{1}{2} \left(\sqrt{\frac{g_{11}}{g_{22}}} \, \delta g_{22} + \sqrt{\frac{g_{22}}{g_{11}}} \, \delta g_{11} \right) \\ &= \frac{1}{2} \sqrt{g} \left(\frac{1}{g_{22}} \, \delta g_{22} + \frac{1}{g_{11}} \, \delta g_{11} \right). \end{aligned}$$

From this we can surmise the following identity:

$$\delta\left(\sqrt{g}\right) = \frac{1}{2}\sqrt{g}\; g^{ij}\; \delta g_{ij}. \tag{B.14}$$

This identity is also derived in Reference [4] (see Eq. E.1.17 in Appendix E).

Now for the variation of the quadratic curvature term, using Eq. B.10:

$$\begin{aligned} \delta\left(\left(S - C_0\right)^2\right) &= 2\left(S - C_0\right)\delta\left(g^{ij}S_{ij}\right) \\ &= 2\left(S - C_0\right)\left(S_{ij}\delta g^{ij} + g^{ij}\delta S_{ij}\right). \end{aligned}$$

So we need the variation of the inverse of the metric and the variation of the shape operator itself. The variation of the inverse metric is most easily obtained from the orthogonality condition of the metric:

$$\begin{aligned} \delta^{i}{}_{k} &= g^{ij}g_{jk} \\ 0 &= \delta\left(g^{ij}g_{jk}\right) \\ 0 &= g_{jk}\delta g^{ij} + g^{ij}\delta g_{jk} \\ g_{jk}\delta g^{ij} &= -g^{ij}\delta g_{jk} \\ g^{lk}g_{kj}\delta g^{ij} &= -g^{lk}g^{ij}\delta g_{jk} \\ \delta^{l}{}_{j}\delta g^{ij} &= -g^{lk}g^{ij}\delta g_{jk}. \end{aligned}$$

Now if we rename indices and simplify we obtain:

$$\delta g^{ij} = -g^{il}g^{jk}\delta g_{lk}. \tag{B.15}$$

The variation of the shape operator is:

$$\begin{aligned} \delta S_{ij} &= \delta\left(n^{\alpha}\nabla_i\nabla_j X^{\alpha}\right) \\ &= \nabla_i\nabla_j X^{\alpha}\; \delta n^{\alpha} + n^{\alpha}\delta\left(\nabla_i\nabla_j X^{\alpha}\right) \\ &= S_{ij}n^{\alpha}\; \delta n^{\alpha} + n^{\alpha}\delta\left(\nabla_i\nabla_j X^{\alpha}\right). \end{aligned}$$

The first term on the RHS vanishes because of the normalization of the normal vector:

$$\begin{aligned} n^\alpha n^\alpha &= 1 \\ \delta\left(n^\alpha n^\alpha\right) &= 0 \\ n^\alpha \ \delta n^\alpha &= 0. \end{aligned}$$

Now for the other term.The ∇_j can be converted to a ∂_j since X^α is a Cartesian vector:

$$\begin{aligned} n^\alpha \delta\left(\nabla_i \nabla_j X^\alpha\right) &= n^\alpha \delta\left(\nabla_i \partial_j X^\alpha\right) \\ &= n^\alpha \delta\left(\partial_i \partial_j X^\alpha + \Gamma^m{}_{ij} \partial_m X^\alpha\right) \\ &= n^\alpha \partial_i \partial_j \delta X^\alpha + n^\alpha \partial_m X^\alpha \delta \Gamma^m{}_{ij} + n^\alpha \Gamma^m{}_{ij} \partial_m \delta X^\alpha. \end{aligned}$$

The second term on the RHS is zero because n^α and $\partial_m X^\alpha$ are othogonal. If we choose a local frame where the connection vanishes, then the third term is also zero, leaving us with the final result:

$$\delta S_{ij} = n^\alpha \nabla_i \nabla_j \delta X^\alpha. \tag{B.16}$$

Going back to the variation of $\delta\left(\left(S - C_0\right)^2\right)$ we find:

$$\delta\left(\left(S - C_0\right)^2\right) = 2\left(S - C_0\right)\left(n^\alpha \nabla^2 X^\alpha - S^{ij} \delta g_{ij}\right). \tag{B.17}$$

B.2.5.2 Variation of the membrane energy

With these expressions in place, we are now in a position to evaluate the variation of the membrane energy.

$$\begin{aligned} E &= \int_M \mathrm{d}s^1 \, \mathrm{d}s^2 \, \sqrt{g} \left[\alpha + \frac{1}{2} k_C \left(S - C_0\right)^2\right] \\ \delta E &= \int_M \mathrm{d}s_1 \, \mathrm{d}s_2 \left\{\delta\left(\sqrt{g}\right)\left[\alpha + \frac{1}{2} k_C \left(S - C_0\right)\right] + \sqrt{g} \, \frac{1}{2} k_C \delta\left(\left(S - C_0\right)^2\right)\right\}. \end{aligned}$$

Now using Eqs. B.14 and B.17 we have:

$$\delta E = \int_{\mathcal{M}} \mathrm{d}s_1\,\mathrm{d}s_2\,\sqrt{g}\,\Bigg\{ \underbrace{g^{ij}\delta g_{ij}\frac{1}{2}\left[\alpha + \frac{1}{2}k_C\left(S - C_0\right)\right]}_{1} - \underbrace{k_C\left(S - C_0\right)S^{ij}\delta g_{ij}}_{2} + \underbrace{k_C\left(S - C_0\right)n^{\alpha}\nabla^2\delta X^{\alpha}}_{3}\Bigg\}. \tag{B.18}$$

We must now integrate by parts and then collect terms to find the force. For simplicity let us assume the membrane is closed so that we may ignore the boundary terms (total derivatives) that arise.

B.2.5.3 Term 1

The integration by parts for the first term will use this result:

$$\nabla_i\left(\delta X^{\alpha}V^i\right) = V^i\nabla_i\delta X^{\alpha} + \delta X^{\alpha}\nabla_i V^i \tag{B.19}$$

$$V^i\nabla_i\delta X^{\alpha} = -\delta X^{\alpha}\nabla_i V^i + \cdots \tag{B.20}$$

where $\cdots$ is a total derivative that we may neglect for a closed surface and V^i represents all prefactors of $\nabla_i\delta X^{\alpha}$. Term 1 looks like $\Phi g^{ij}\delta g_{ij}$ where Φ is a scalar function; using the above result we obtain:

$$\begin{aligned}\Phi g^{ij}\delta g_{ij} &= \Phi g^{ij}\left(\nabla_i\delta X^{\alpha}\nabla_j X^{\alpha} + \nabla_i X^{\alpha}\nabla_j\delta X^{\alpha}\right)\\ &= 2\Phi\nabla_i X^{\alpha}\nabla^i\delta X^{\alpha}\\ &= -2\delta X^{\alpha}\left(\nabla_i X^{\alpha}\nabla^i\Phi + \Phi\nabla^2 X^{\alpha}\right)\\ &= -2\delta X^{\alpha}\left(\nabla_i X^{\alpha}\nabla^i\Phi + n^{\alpha}S\Phi\right).\end{aligned}$$

The first term on the RHS corresponds to a force in the place of the membrane since it is proportional to ∇X^{α}, and the second is normal to it. In this derivation we will only collect terms multiplying the normal vector, since it is the normal force we are most interested in.

The final result for term 1 with $\Phi = \frac{1}{2}\left[\alpha + \frac{1}{2}k_C\left(S - C_0\right)^2\right]$ is:

$$g^{ij}\delta g_{ij}\Phi = -\delta X^\alpha\left\{\nabla^i X^\alpha\left(\cdots\right) - n^\alpha S\left[\alpha + \frac{1}{2}k_C\left(S - C_0\right)^2\right]\right\}. \tag{B.21}$$

B.2.5.4 Term 2

For term 2 we have, using Eq. B.20 with $V^i = -2k_C\left(S - C_0\right)S^{ij}\nabla_j X^\alpha$:

$$\begin{aligned}
-k_C\left(S - C_0\right)S^{ij}\delta g_{ij} &= -2k_C\left(S - C_0\right)S^{ij}\nabla_j X^\alpha\nabla_i\delta X^\alpha \\
&= 2k_C\ \delta X^\alpha\nabla_i\left[\left(S - C_0\right)S^{ij}\nabla_j X^\alpha\right] \\
&= 2k_C\ \delta X^\alpha\left[\nabla_j X^\alpha\left(\cdots\right) + \left(S - C_0\right)S^{ij}\nabla_i\nabla_j X^\alpha\right].
\end{aligned}$$

Now using Eq. B.7 the final result for term 2 is:

$$-k_C\left(S - C_0\right)S^{ij}\delta g_{ij} = 2k_C\ \delta X^\alpha\left[\nabla_j X^\alpha\left(\cdots\right) + n^\alpha\left(S - C_0\right)S^{ij}S_{ij}\right]. \tag{B.22}$$

B.2.5.5 Term 3

For term 3 we apply Eq. B.20 twice to find:

$$\begin{aligned}
\left(S - C_0\right)n^\alpha\nabla^2\delta X^\alpha &= k_C\ \delta X^\alpha\nabla^2\left[\left(S - C_0\right)n^\alpha\right] \\
&= k_C\ \delta X^\alpha\left[n^\alpha\nabla^2 S + \left(S - C_0\right)\nabla^2 n^\alpha\right].
\end{aligned}$$

We can use the definition of the shape operator (Eq. B.6) along with Eqs. B.7 and B.8 to calculate $\nabla^2 n^\alpha$:

$$\begin{aligned}
\nabla^2 n^\alpha &= \nabla^i\nabla_i n^\alpha \\
&= \nabla^i\left[S^{ij}\left(-\nabla_j X^\alpha\right)\right] \\
&= \nabla^i S_{ij}\left(-\nabla^j X^\alpha\right) - S^{ij}\nabla_i\nabla_j X^\alpha \\
&= \nabla_j S\left(-\nabla^j X^\alpha\right) - n^\alpha S^{ij}S_{ij}.
\end{aligned}$$

Putting this all together the final form of term 3 is:

$$\begin{aligned}(S - C_0)\, n^\alpha \nabla^2 \delta X^\alpha = k_C \delta X^\alpha \left[\nabla^i X^\alpha (\cdots) + \right. \\ \left. n^\alpha \left(\nabla^2 S - S^{ij} S_{ij} (S - C_0)\right)\right].\end{aligned} \tag{B.23}$$

B.2.5.6 Combining terms

Going back to Eq. B.18 and putting in the results for the three terms worked out above gives:

$$\begin{aligned}\delta E = \int_{\mathcal{M}} \mathrm{d}s_1\, \mathrm{d}s_2\, \sqrt{g} \left\{ \nabla^i X^\alpha (\cdots) + \right. \\ \left. n^\alpha \left[-S\alpha + k_C \nabla^2 S + k_C (S - C_0) S^{ij} S_{ij} - \frac{1}{2} k_C S (S - C_0)^2 \right] \right\}.\end{aligned} \tag{B.24}$$

Now using Eq. B.13 and the equilibrium condition $\delta E = 0$ we arrive at an expression for the external force density needed to maintain mechanical equilibirum for our membrane model:

$$\begin{aligned} f_\perp &= -S\alpha + k_C \left[\nabla^2 S + (S - C_0) S^{ij} S_{ij} - \frac{1}{2} S (S - C_0)^2 \right] \\ &= -S\alpha + k_C \left[\nabla^2 S + (S - C_0)\left(S^2 - 2K\right) - \frac{1}{2} S (S - C_0)^2 \right]. \end{aligned}$$

After a few lines of algebra to simplify things we arrive, finally, at the expression:

$$f_\perp = -S\alpha + k_C \left[\nabla^2 S - 2K (S - C_0) + \frac{1}{2} S \left(S^2 - {C_0}^2\right) \right]. \tag{B.25}$$

B.3 Derivation of the tether equation

Let us write down the energy of a spherical vesicle of radius R having a cylidrical tether of length L and radius r pulled out of it, with the tether being closed by a hemispherical cap:

$$E = E_{\text{bend}} - f\, L - \alpha\, A - p\, V. \tag{B.26}$$

The energy of bending is:

$$\begin{aligned} E_{\text{bend}} &= 8\pi\kappa_B + \frac{1}{2}\kappa_B \left(\frac{1}{r}\right)^2 2\pi r L + 4\pi\kappa_B && \text{(B.27)} \\ &= 12\pi\kappa_B + \pi\kappa_B \frac{L}{r}. && \text{(B.28)} \end{aligned}$$

The constant bending terms aren't actually going to contribute anything useful. The area and volume are:

$$\begin{aligned} A &= 4\pi R^2 + 2\pi r L + 2\pi r^2 && \text{(B.29)} \\ V &= \frac{4}{3}\pi R^3 + \pi r^2 L + \frac{2}{3}\pi r^3. && \text{(B.30)} \end{aligned}$$

Now we need to find out what values of L, r, and R will minimize the energy. We have 3 equations:

$$\begin{aligned} \frac{\partial E(R,L,r)}{\partial R} &= 0 && \text{(B.31)} \\ \frac{\partial E(R,L,r)}{\partial L} &= 0 && \text{(B.32)} \\ \frac{\partial E(R,L,r)}{\partial r} &= 0. && \text{(B.33)} \end{aligned}$$

A membrane tether's length is typically long compared to its radius, and the radius of the tether is also small compared to the radius of a typical vesicle from which it is drawn. Let us assume our geometry meets these approximations and simplify the calculation by neglecting any terms that go like r/L or r/R.

Tackling the first equation, we have

$$\begin{aligned} \frac{\partial E}{\partial R} &= -8\pi\alpha R - 4\pi p R^2 = 0 && \text{(B.34)} \\ -2\alpha R &= p R^2 && \text{(B.35)} \\ p &= -\frac{2\alpha}{R}, && \text{(B.36)} \end{aligned}$$

which is just the Laplace law familiar from the study of soap bubbles. In the body of the vesicle, away from the tether, this law still applies.

Now for the second equation:

$$\begin{aligned}
\frac{\partial E}{\partial L} &= \frac{\pi\kappa_B}{r} - f - 2\pi\alpha r - \pi p r^2 = 0 && \text{(B.37)}\\
f &= \frac{\pi\kappa_B}{r} - 2\pi\alpha r - \pi\left(-\frac{2\alpha}{R}\right) r^2 && \text{(B.38)}\\
f &= \frac{\pi\kappa_B}{r} - 2\pi\alpha r\left(1 - \frac{r}{R}\right) && \text{(B.39)}\\
f &\approx \frac{\pi\kappa_B}{r} - 2\pi\alpha r. && \text{(B.40)}
\end{aligned}$$

And the final equation:

$$\begin{aligned}
\frac{\partial E}{\partial r} &= -\pi\kappa_B\frac{L}{r^2} - 2\pi\alpha L - 4\pi\alpha r - 2\pi p r L - 2\pi p r^2 = 0 && \text{(B.41)}\\
\frac{\pi\kappa_B L}{r^2} &= -2\pi\alpha L - 4\pi\alpha r - 2\pi\left(-\frac{2\alpha}{R}\right) rL - 2\pi\left(-\frac{2\alpha}{R}\right) r^2 && \text{(B.42)}\\
\frac{\pi\kappa_B}{r^2} &= -2\pi\alpha - 4\pi\alpha\left(\frac{r}{L}\right) + 4\pi\alpha\left(\frac{r}{R}\right) + 4\pi\alpha\left(\frac{r}{R}\right)\left(\frac{r}{L}\right) && \text{(B.43)}\\
\frac{\pi\kappa_B}{r^2} &\approx -2\pi\alpha r. && \text{(B.44)}
\end{aligned}$$

Using this last result we may eliminate the term with α from Eq. B.40 and obtain a very simple equation relating the tether force and radius:

$$f = \frac{2\pi\kappa_B}{r}. \tag{B.45}$$

So as we increase the force on the end of the tether, its radius shrinks. Other ways of writing this result are [5]:

$$\begin{aligned}
f &= \frac{2\pi\kappa_B}{R}, && \text{(B.46)}\\
&= 4\pi R\alpha, && \text{(B.47)}\\
&= 2\pi\sqrt{2\kappa_B\alpha}. && \text{(B.48)}
\end{aligned}$$

Bibliography

[1] P B Canham. The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell. *J Theor Biol*, 26(1):61–81, 1970.

[2] E A Evans. Bending resistance and chemically induced moments in membrane bilayers. *Biophys J*, 14:923–931, 1974.

[3] W Helfrich. Elastic properties of lipid bilayers: theory and possible experiments. *Z Naturforsch*, 28(11):693–703, 1973.

[4] R M Wald. *General Relativity*. The University of Chicago Press, Chicago, 1984.

[5] A Upadhyaya and M P Sheetz. Tension in tubulovesicular networks of Golgi and endoplasmic reticulum membranes. *Biophys J*, 86(5):2923–2928, 2004.

Appendix C

Cooperative gating and spatial organization of membrane proteins through elastic interactions

C.1 Simulation input files

Two input files are required for the Monte Carlo simulation: one with values of physical parameters and another with values for simulation parameters. The physical parameter file we used followed by an example of a simulation parameter file are given below.

C.1.1 Explanation of `physics.inp`

The values of various necessary physical parameters are input through this file. Values are given one per line, in the following order and units:

1. Area stretch modulus K_A in $\mathrm{k_B T/nm^2}$

2. Single leaflet bending modulus κ_b in $\mathrm{k_B T}$

3. Single leaflet thickness l in nm

4. Area change of the channel upon gating ΔA in $\mathrm{nm^2}$

5. Radius of the open channel r_o in nm

6. Radius of the closed channel r_c in nm

7. Internal conformational energy change upon gating ΔG_{gate} in $\mathrm{k_B T}$.

C.1.2 physics.inp

```
29
7
1.75
20
3.5
2.5
52
```

C.1.3 Explanation of sim.inp

The parameters for the simulation itself are read in from this input file. Just as with `physics.inp` the values are given one per line, as follows:

1. Path to the text file with closed-closed potentials V_{cc}
2. Path to the text file with open-closed potentials V_{oc}
3. Path to the text file with open-open potentials V_{oo}
4. Resolution `res` of the lattice
5. Side length `side_len` of the lattice in nm
6. Total number of channel proteins `M` on the lattice
7. Total number of steps `NMC` to be taken in the Markov chain
8. Random number generator seed
9. Initial value of (dimensionless) tension τ_i
10. Final value of (dimensionless) tension τ_f
11. Optional output directive(s), one per line:

    (a) `!history`

    Write out a file `history.txt` with a detailed log of simulation activity. The first line in the file is the number of Markov chain steps to be taken at each tension value, `NMC`. At each new tension value from τ_i to τ_f, one line is written out with the current value of τ. Following the line with the value of τ, there follow lines with tab-separated values each time a gating change occurs: `M` values with the open/closed state of each protein on the lattice, the current number of interacting dimers and the current Markov chain step number.

(b) `!channel`

Write out a file `channel.txt` with data calculated at each tension value. The tab-separated values on each line are: tension value τ, probability of being open P_{open}, probability of being in an interacting dimer P_{dimer}, and average separation of the channels $\langle r \rangle$ in nm.

(c) `!dimer`

Write out a file `dimer.txt` having the lifetimes of interacting dimers given, one per line, in units of Markov chain steps.

(d) `!gating`

Write out a file `gating.txt` having the lifetimes of open channels given, one per line, in units of Markov chain steps.

The parameters `res` and `side_len` are used to determine the total number of lattice sites N as

$$N = \text{res} \times \text{side\_len}/\lambda, \tag{C.1}$$

with the natural deformation length scale given by $\lambda = (\kappa_b l^2/K_A)^{1/4} \approx 1\,\text{nm}$. The simulation is run with `M` channel proteins on an $N \times N$ lattice from an initial tension value of τ_i to a final value of τ_f in increments of 0.01 with `NMC` Markov steps taken at each increment.

The example input file shown below specifies that two channel proteins be placed on a $45\,\text{nm} \times 45\,\text{nm}$ grid with 10 million Markov chain steps generated at each of the tension values from $\tau_i = 1.10$ to $\tau_f = 1.30$ in increments of $\Delta\tau = 0.01$ (the increment is fixed in the source code and not specified in the input file). The random number seed is set to 92478 and output is requested for the lifetimes of interacting dimers (`!dimer`) as well as the standard statistics (`!channel`) at each tension value: P_{open}, P_{dimer}, and $\langle r \rangle$. The potentials are found in the current directory in the files `Vcc.txt`, `Voc.txt`, and `Voo.txt`. The lattice will be set up so that each typical interaction length λ is split into 10 lattice sites.

C.1.4 `sim.inp`

```
Vcc.txt
Voc.txt
Voo.txt
```

```
10
45
2
10000000
92478
1.10
1.30
!dimer
!channel
```

C.2 MCMC source code

Below are listings for the source code to the Markov Chain Monte Carlo (MCMC) simulation.

C.2.1 main.cc

```
#include "channels.hh"

int main( int argc, void **argv ) {
   // error check
   if ( argc < 2 ) {
      cout << "Usage: channels physics.inp sim.inp" << endl;
      return -1;
   }

   // input file names
   const string p( (char *)argv[1] );
   const string s( (char *)argv[2] );

   // run simulation
   channel_KMC *sim = new channel_KMC( p, s );
   sim->PrintParams();
   sim->DoSeries();
   delete sim;

   return 0;
}
```

C.2.2 channels.hh

```
#include <climits>
#include <cmath>
#include <string>
#include <iostream>
#include <iomanip>
#include <fstream>

#include "RNGXCI.h"
#include "potential.hh"

using std::string;
using std::cout;
using std::cerr;
using std::endl;
using std::ifstream;
using std::ofstream;
using std::ios;
using std::setprecision;
using std::setiosflags;

// (fast?) integer absolute value
// from http://graphics.stanford.edu/~seander/bithacks.html#IntegerAbs
#define ABS(x) ((x) ^ ((x) >> (sizeof(int) * CHAR_BIT - 1))) -
               ((x) >> (sizeof(int) * CHAR_BIT - 1));

/*********************/
/* class channel_KMC */
/*********************/

class channel_KMC {
public:
   channel_KMC( const string, const string );
   ~channel_KMC( void );

   // print parameter values
   void PrintParams( void );

   // calculate the order parameter and dimer probabilty
   // for one tension value
   double DoTension( const float, double&, double&, double& );

   // calculate a series of tension values
   void DoSeries( void );

   // print out potential
   void PrintV( int, float, const string );
```

```
private:
   enum { FALSE, TRUE };

   enum { MATCH }; // for comparing strings

   // move types in metropolis()
   enum { NO_CHANGE, LATTICE_HOP, HOP_AND_SWITCH };
   // NOTE: any type of accepted move must have values > 0
   // DoTension() depends on this

   enum { VCC, VOC, VOO };          // potentials

   enum { L_UNOCC, L_CLOSED, L_OPEN }; // lattice states

   enum { CLOSED, OPEN };           // channel states
   // NOTE: the following must hold:
   //  OPEN  +  OPEN  = VOO
   //  OPEN  + CLOSED = VOC
   // CLOSED + CLOSED = VCC
   // interaction() depends on this
   //  !OPEN  = CLOSED
   // !CLOSED =  OPEN
   // metropolis() depends on this

   // physical constants, read from file
   //-----------------------------------

   float Ka;    // leaflet strech modulus (kT/nm^2)
   float kappa; // leaflet bending modulus (kT)
   float l;  // leaflet thickness (nm)
   float da;    // measured gating area change (nm^2)

   float ro; //  open  channel radius (nm)
   float rc; // closed channel radius (nm)

   float gating; // gating energy (kT)

   // simulation constants, read from file
   //-------------------------------------

   string fnames[3]; // file names with potential data

   int res;    // lattice resolution
   float side_len; // length of box side (nm)

   int M;       // number of proteins on lattice
   int NMC;  // number of (accepted) MC steps to be taken
   RNlong seed; // random number generator seed

   // derived constants
   //------------------

   float lambda; // natural length scale (nm)

   float delta; // dimensionless tension

   float dalpha; // dimensionless gating area change

   float rhoo; // scaled open channel radius
   float rhoc; // scaled closed channel radius

   float h;    // lattice spacing
   int N;      // number of lattice sites is N x N
   int N2;     // = N/2
   long unocc; // number of unoccupied lattice sites

   // excluded radius; hard core potential at r < rex
   float rex[3];

   // dimer interaction energy threshold
   float dimer_energy;

   // simulation variables
   //---------------------

   RNGXCI *rgen; // random number generator

   int *channel_i; // channel i (x) indices
   int *channel_j; // channel j (y) indices
   int *channel_s; // channel state: OPEN or CLOSED

   int *ui; // unoccupied sites, i index
   int *uj; // unoccupied sites, j index

   int **lattice; // values: L_UNOCC, L_OPEN, L_CLOSED
   float **dist;  // distance array

   potential *Voo, *Voc, *Vcc; // elastic potentials

   float ***U; // potential look-up table
   // U(VCC, *, *) is the closed - closed potential
   // U(VOC, *, *) is the  open  - closed potential
   // U(VOO, *, *) is the  open  -  open  potential
   //  lipid & area energies, units of kappa
   float Elipid[2]; // lipid configuration energy
   float  Esurf[2]; // surface tension energy
   float    Ela[2]; // sum of lipid and area energies

   float tau_i, tau_f;  // initial and final tension values
   int nopen;        // number of channels in the OPEN state
   int ndimer;       // number of dimers
   long dimer_start; // keep track of MC step where dimer forms
   long *gating_start;  // vector with MC step where channel opened
   float total_dist; // total separation between all channels

   // output request strings
   //-----------------------
   const static string history_request;
   const static string channel_request;
   const static string   dimer_request;
   const static string  gating_request;

   // output request flags
   //---------------------
   bool history_flag;
   bool channel_flag;
   bool   dimer_flag;
   bool  gating_flag;

   // file storing transition/dimer data
   //-----------------------------------
   ofstream *channel;   // data on open probability, dimerization,
                        // average separation
   ofstream *history;   // history of system state changes
   ofstream *dimer;  // dimer lifetimes
   ofstream *gate;      // open channel lifetimes

   // internal methods
   //-----------------

   void show_lattice( int ** );

   /**********/
   /* get_ij */
   /**********/

   inline void get_ij( const int i1, const int j1,
                       const int i2, const int j2,
                       int& di, int& dj ) {
```

```
      int idif = i1 - i2;
      int jdif = j1 - j2;

      di = ABS(idif);
      dj = ABS(jdif);

      // PBC
      // get the distance to the nearest image
      if ( di > N2 ) di = N - di;
      if ( dj > N2 ) dj = N - dj;
} // end get_ij()

/***************/
/* interaction */
/***************/

inline float interaction( int c, int s, int i, int j ) {
   float E = 1.;
   for ( int ci = (c+1) % M; ci != c; ++ci %= M ) {
      int di, dj;
      get_ij( i, j, channel_i[ci], channel_j[ci], di, dj );

      // interaction energy
      E *= U[ s + channel_s[ci] ][ di ][ dj ];
   }
   return E;
} // end interaction()

/**************/
/* metropolis */
/**************/

inline int metropolis( int c, int old_s, int old_i, int old_j,
                       int new_i, int new_j,
                       int *ai, int *aj, int n ) {
   // c is the channel under consideration
   // old_* and new_* are the new states and positions
   // ai,j is an array of i/j that would
   //      need swapping with ci channel's i/j
   // n is the index into the ai,j array

   float Ei, Ef1, Ef2, Vi, Vf1, Vf2; // initial/final energies

   Vi = interaction( c, old_s, old_i, old_j );
   Ei = Vi * Ela[ old_s ];

   Vf1 = interaction( c, old_s, new_i, new_j );
```

```
Ef1 = Vf1 * Ela[ old_s ];

Vf2 = interaction( c, !old_s, new_i, new_j );
Ef2 = Vf2 * Ela[ !old_s ];

double r = rgen->GetRandom();

// two states are considered:
// (1) a move w/o state change: LATTICE_HOP
// (2) a move w/ state change:  HOP_AND_SWITCH
// if both would be accepted, choose randomly between them
// if only one would be accepted, accept it
// otherwise reject both

int move;
if ( (Ef1 < Ei) && (Ef2 < Ei) )
   ( r < 0.5 ) ? move = LATTICE_HOP
            : move = 2;
else if ( Ef1 < Ei )
   move = LATTICE_HOP;
else if ( Ef2 < Ei )
   move = HOP_AND_SWITCH;
else {
   float k1 = Ei/Ef1;
   float k2 = Ei/Ef2;

   if ( r < k1 && r < k2 )
      ( rgen->GetRandom() < 0.5 ) ? move = 1
                           : move = 2;
   else if ( r < k1 )
      move = LATTICE_HOP;
   else if ( r < k2 )
      move = HOP_AND_SWITCH;
   else
      return NO_CHANGE;
}

// MOVE ACCEPTED
int new_s;
float Vf;
if ( move == LATTICE_HOP ) {
   new_s = old_s;
   Vf = Vf1;
} else {
   new_s = !old_s;
   Vf = Vf2;
   ( new_s == OPEN ) ? nopen++ : nopen--;
```

```
      }

      channel_i[ c ] = new_i;
      channel_j[ c ] = new_j;
      channel_s[ c ] = new_s;

      ai[ n ] = old_i;
      aj[ n ] = old_j;

      // ONLY WORKS FOR M=2!!!
      ( Vf <= dimer_energy ) ? ndimer = 1 : ndimer = 0;

      for ( int ci = (c+1) % M; ci != c; ++ci %= M ) {
         int i = channel_i[ci];
         int j = channel_j[ci];


         int old_di, old_dj;
         int new_di, new_dj;

         get_ij( old_i, old_j, i, j, old_di, old_dj );
         get_ij( new_i, new_j, i, j, new_di, new_dj );

         total_dist -= dist[old_di][old_dj];
         total_dist += dist[new_di][new_dj];

      }

      return move;
   } // end metropolis()
}; // end class channel_KMC
```

C.2.3 channels.cc

```
#include "channels.hh"

// const member initializations
//----------------------------

const string channel_KMC::history_request = string("!history");
const string channel_KMC::channel_request = string("!channel");
const string channel_KMC::dimer_request   = string("!dimer");
const string channel_KMC::gating_request  = string("!gating");

/**********************/
/* constructor method */
/**********************/

channel_KMC::channel_KMC( const string phys, const string sim ) {

   // read in physical parameters
   ifstream phys_params( phys.c_str() );

   phys_params >> Ka;
   phys_params >> kappa;
   phys_params >> l;
   phys_params >> da;
   phys_params >> ro;
   phys_params >> rc;
   phys_params >> gating;

   phys_params.close();

   // read in simulation parameters
   ifstream sim_params( sim.c_str() );

   sim_params >> fnames[VCC];
   sim_params >> fnames[VOC];
   sim_params >> fnames[VOO];
   sim_params >> res;
   sim_params >> side_len;
   sim_params >> M;
   sim_params >> NMC;
   sim_params >> seed;
   sim_params >> tau_i;
   sim_params >> tau_f;

   // read in output requests
   history_flag = FALSE;
   channel_flag = FALSE;
```

```
dimer_flag = FALSE;
while (! sim_params.eof() ) {
   string tmp;
   sim_params >> tmp;

   if ( tmp.compare( history_request ) == MATCH )
      history_flag = TRUE;
   if ( tmp.compare( channel_request ) == MATCH )
      channel_flag = TRUE;
   if ( tmp.compare( dimer_request ) == MATCH )
      dimer_flag = TRUE;
   if ( tmp.compare( gating_request ) == MATCH )
      gating_flag = TRUE;
}

sim_params.close();

// initalize derived parameters
lambda = pow( (kappa*l*l)/Ka , 0.25 );
dalpha = da/(lambda * lambda);
rhoo = ro/lambda;
rhoc = rc/lambda;

N = int( (side_len/lambda)*res );
N2 = int( N/2 );
h = side_len/N;
unocc = N*N-M;

// dimer interaction energy
// V(r) <= -1 kT is a dimer
dimer_energy = exp(-1.);

rex[VCC] = 2*rc;
rex[VOC] = rc+ro;
rex[VOO] = 2*ro;

// allocate storage (1d)
channel_i = new int[M];
channel_j = new int[M];
channel_s = new int[M];
gating_start = new long[M];

ui = new int[unocc];
uj = new int[unocc];

// allocate storage (2d)
lattice = new int*[N];
```

```
dist = new float*[N];
for ( int i = 0; i < N; ++i ) {
     lattice[i] = new int[N];
     dist[i] = new float[N];
}

// allocate storage (3d)
U = new float**[3];
for ( int i = 0; i < 3; ++i ) {
   U[i] = new float*[N];
   for ( int j = 0; j < N; ++j )
      U[i][j] = new float[N];
}

// initialize random number generator
rgen = new RNGXCI( seed );

// initialize channels
for (int i = 0; i < M; ) {
   int ii = int( rgen->GetRandom() * N );
   int jj = int( rgen->GetRandom() * N );

   if ( lattice[ii][jj] == L_UNOCC ) {
      channel_i[i] = ii;
      channel_j[i] = jj;

      // initially closed
      channel_s[i] = CLOSED;
      lattice[ii][jj] = L_CLOSED;

      ++i;
   }
}
nopen = 0;

// find unoccupied sites and initialize dist
long n = 0;
for (int i = 0; i < N; ++i)
   for (int j = 0; j < N; ++j ) {
      dist[i][j] = sqrt(  i*i*h*h + j*j*h*h );
      if (lattice[i][j] == L_UNOCC) {
         ui[n] = i;
         uj[n] = j;
         ++n;
      }
   }
   // initialize the potentials
   Voo = new potential( fnames[VOO], N, h, side_len, rex[VOO], dist,
                        U[VOO], lambda, kappa );
   Voc = new potential( fnames[VOC], N, h, side_len, rex[VOC], dist,
                        U[VOC], lambda, kappa );
   Vcc = new potential( fnames[VCC], N, h, side_len, rex[VCC], dist,
                        U[VCC], lambda, kappa );

   // initiallize ndimer, total_dist
   ndimer = 0;
   total_dist = 0.;
   for (int c = 0; c < M; ++c ) {
      int i = channel_i[c];
      int j = channel_j[c];

      for ( int ci = (c+1) % M; ci != c; ++ci %= M ) {
         int di, dj;
         get_ij( i, j, channel_i[ci], channel_j[ci], di, dj );

         total_dist += dist[di][dj];
         if ( U[ VCC ][ di ][ dj ] <= dimer_energy ) ndimer++;
      }
   }

   // ONLY works for M=2!!!
   if ( ndimer ) dimer_start = 0;

   // we counted everything twice in the loop
   total_dist *= 0.5;
   ndimer = round( ndimer / 2 );

   // open output files
   if ( history_flag ) {
      history = new ofstream( "history.txt" );
      // write out number of MC steps for each tau
      (*history) << NMC << endl;
   }
   if ( channel_flag )
      channel = new ofstream( "channel.txt" );
   if ( dimer_flag )
        dimer = new ofstream( "dimer.txt" );
   if ( gating_flag )
        gate = new ofstream( "gating.txt" );

} // end constructor channel_KMC
```

```
/********************/
/* destructor method */
/********************/

channel_KMC::~channel_KMC( void ) {
   // deallocate storage (1d)
   delete [] channel_i;
   delete [] channel_j;
   delete [] channel_s;
   delete [] gating_start;

   // deallocate storage (2d)
   for ( int i = 0; i < N; ++i )
      delete [] lattice[i];
   delete [] lattice;

   // deallocate storage (3d)
   for ( int i = 0; i < 3; ++i ) {
      for ( int j = 0; j < N; ++j )
         delete [] U[i][j];
      delete [] U[i];
   }
   delete [] U;

   delete Voo;
   delete Voc;
   delete Vcc;

   // cleanup output files
   if ( history_flag ) {
      (*history).close();
      delete history;
   }

   if ( channel_flag ) {
      (*channel).close();
      delete channel;
   }

   if ( dimer_flag ) {
      (*dimer).close();
      delete dimer;
   }
} // end destructor channel_KMC
```

```
/***************/
/* PrintParams */
/***************/

void channel_KMC::PrintParams( void ) {

   cout << "      General" << endl;
   cout << "-------------------" << endl;
   cout << "Number of proteins: " << M << endl;
   cout << "MC steps per point: " << NMC << endl;
   cout << "       random seed: " << seed << endl;
   cout << endl;

   cout << "Physical parameters" << endl;
   cout << "-------------------" << endl;
   cout << "    Ka: " << Ka << endl;
   cout << " kappa: " << kappa << endl;
   cout << "     l: " << l << endl;
   cout << "    da: " << da << endl;
   cout << "    ro: " << ro << endl;
   cout << "    rc: " << rc << endl;
   cout << "gating: " << gating << endl;
   cout << endl;

   cout << "Interaction potential" << endl;
   cout << "---------------------" << endl;
   cout << "cc potential: " << fnames[0] << endl;
   cout << "oc potential: " << fnames[1] << endl;
   cout << "oo potential: " << fnames[2] << endl;
   cout << endl;

   cout << "       Lattice" << endl;
   cout << "---------------------" << endl;
   cout << "     box side length: " << side_len << endl;
   cout << "        grid spacing: " << h << endl;
   cout << "  lattice resolution: " << res << endl;
   cout << "grid points per side: " << N << endl;
   cout << endl;

   // print derived parameters

   cout << "  Other" << endl;
   cout << "--------" << endl;
   cout << " lambda: " << lambda << endl;
   cout << " dalpha: " << dalpha << endl;
   cout << "   rhoo: " << rhoo << endl;
   cout << "   rhoc: " << rhoc << endl;
   cout << endl;
} // end PrintParams()

/************/
/* DoSeries */
/************/

void channel_KMC::DoSeries( void ) {
   double sigma, sigma_D, r_avg;

   cout << "\tOne channel series" << setiosflags( ios::fixed) << endl;
   for ( float tau = tau_i; tau <= tau_f; tau += 0.01 ) {
      cout << "\t\t" << setprecision( 2 )
           << tau
           << "\t" << setprecision( 1 )
           << 100*DoTension( tau, sigma, sigma_D, r_avg )
           << "%\n";

      if ( channel_flag )
         (*channel) << tau << "\t"
                    << sigma << "\t"
                    << sigma_D << "\t"
                    << r_avg << endl;
   }
} // end DoSeries()

/*************/
/* DoTension */
/*************/

double channel_KMC::DoTension( const float tau, double &order_param,
                               double &dimer_prob, double &r_avg ) {
   // tau is the leaflet tension (kT/nm^2)
   float delta = tau*lambda*lambda/kappa; // dimensionless tension

   float dfac = 1/( 0.5 * float(M) * float(M-1) );

   Elipid[CLOSED] = 0;
   Elipid[ OPEN ] = gating/kappa;

   Esurf[CLOSED] = 0;
   Esurf[ OPEN ] = -2*delta*dalpha;

   Ela[CLOSED] = exp( kappa*(Elipid[CLOSED] + Esurf[CLOSED]) );
   Ela[ OPEN ] = exp( kappa*(Elipid[ OPEN ] + Esurf[ OPEN ]) );

   long order_sum = 0; // order parameter for the system
```

```
                // probability of a channel being open

long dimer_sum = 0; // number of dimers observed

// total separation of all channels for all MC steps
double dist_sum = 0.;

// initialize counters
dimer_start = 0;
for ( int i = 0; i < M; ++i )
   gating_start[i] = 0;

// fill the potential look-up table U
Voo->table( tau );
Voc->table( tau );
Vcc->table( tau );

if ( history_flag ) {
   (*history) << tau << endl;

   // if this is the first tension, write out the initial state
   if ( tau == tau_i ) {
      for ( int i = 0; i < M; ++i )
         (*history) << channel_s[i] << "\t";
      (*history) << ndimer << "\t0" << endl;
   }
}

if ( dimer_flag )
   (*dimer) << tau << endl;

if ( gating_flag )
   (*gate) << tau << endl;

long naccept = 0;
for ( long nloop = 1; nloop <= NMC; ++nloop ) {
   // nloop keeps track of the total number of MC loops
   // c is the index of the channel we’re working with

   // randomly choose the next channel
   int c = int( rgen->GetRandom() * M );

   int old_i = channel_i[c];  // i index
   int old_j = channel_j[c];  // j index
   int old_s = channel_s[c];  // internal state

   // calculate a move
   int new_i, new_j;

   // lattice hop (non-local)
   // move to an unoccupied site
   long ii = long( unocc * rgen->GetRandom() );
   new_i = ui[ ii ];
   new_j = uj[ ii ];

   int prev_ndimer = ndimer;
   int move = metropolis( c, old_s, old_i, old_j,
                          new_i, new_j, ui, uj, ii );

   if ( move ) naccept++;

   if ( history_flag ) {
      // if a transition occurred (state or dimer)...
      if ( move == HOP_AND_SWITCH || prev_ndimer != ndimer ) {
         // write out state and dimerization info
         for ( int i = 0; i < M; ++i )
            (*history) << channel_s[i] << "\t";
         (*history) << ndimer << "\t";
         (*history) << nloop << endl;
      }
   } // history data output

   if ( gating_flag && move == HOP_AND_SWITCH ) {
      // write out an open channel lifetime
      if ( old_s == CLOSED )
         gating_start[c] = nloop;
      else
         (*gate) << ( nloop - gating_start[c] ) << endl;
   }

   if ( dimer_flag && prev_ndimer != ndimer ) {
      // only works for M=2!!!
      if ( ndimer == 0 )
         (*dimer) << ( nloop - dimer_start ) << endl;
      else
         dimer_start = nloop;
   } // dimer data output

   order_sum += nopen;
   dimer_sum += ndimer;
   dist_sum += total_dist * dfac;

} // end main Monte Carlo loop
```

```
   order_param = double(order_sum)/double(NMC)/double(M);
   dimer_prob = double(dimer_sum)/double(NMC);
   r_avg = dist_sum/double(NMC);
   return double(naccept)/double(NMC);

} // end DoTension()

/****************/
/* show_lattice */
/****************/

void channel_KMC::show_lattice( int **L ) {
   for ( int i = 0; i < (N+2); ++i )
      cout << "#";
   cout << endl;

   for ( int i = 0; i < N; ++i ) {
      cout << "#";
      for ( int j = 0; j < N; ++j )
           ( L[i][j] == L_UNOCC ) ? cout << "." :
         ( ( L[i][j] == L_OPEN )  ? cout << "O" : cout << "C" );
      cout << "#" << endl;
   }

   for ( int i = 0; i < (N+2); ++i )
      cout << "#";
   cout << endl;
}

/**********/
/* PrintV */
/**********/

void channel_KMC::PrintV( int V, float tau, const string fname ) {
   potential *U;

   switch ( V ) {
      case VOO :
         U = Voo;
         break;

      case VOC :
         U = Voc;
         break;

      case VCC :
         U = Vcc;
```

```
            break;
    }

    // calculate table
    U->table( tau );

    // write out file
    U->write_potential( fname );
}
```

C.2.4 potential.hh

```
#include <cmath>
#include <string>
#include <iostream>
#include <fstream>

using std::cout;
using std::cerr;
using std::endl;
using std::string;
using std::ifstream;
using std::ofstream;

/*******************/
/* class potential */
/*******************/

class potential {
public:
   potential( const string, int, float, float, float,
              float **, float **, float, float );
   void table( float );
   ˜potential( void );
   void write_potential( string );

private:
   int nt, nr;
   float lmax; // maximum interaction distance
   float hard_core; // value of hard core in potentials
   float h;    // lattice spacing
   float side_len; // length of box side (nm)
   int N;      // number of lattice sites is N x N
   float **t;  // potential look-up table
   float lambda, kappa; // conversion factors;
                   // KC uses length units of lambda
                   // and energy units of kappa
   float *r;      // radius
   float *tension;  // tension
   float **u;     // potential table read from file
   float **dist;  // table of distance values

   // excluded radius; hard core potential at r < rex
   float rex;

   void interp( float, float * );

}; // end class potential
```

C.2.5 potential.cc

```
#include "potential.hh"

/*********************/
/* constructor method */
/*********************/

potential::potential( const string fname, int N, float h,
                      float side_len, float rex, float **d,
                      float **t, float lambda, float kappa ) {
   this->hard_core = 80.;
   this->rex = rex;
   this->N = N;
   this->h = h;
   this->side_len = side_len;
   this->t = t;
   this->dist = d;
   this->lambda = lambda;
   this->kappa = kappa;

   ifstream ufile( fname.c_str() );

    // read in tension values
   ufile >> nt;
   this->tension = new float[nt];
   for ( int j = 0; j < nt; ++j ) {
      ufile >> tension[j];
   }

    // read in radius values
   ufile >> nr;
   this->r = new float[nr];
   for ( int j = 0; j < nr; ++j ) {
      // read radius, convert to nm
      // KC measures radius from hard core
      float tmp;
      ufile >> tmp;
      r[j] = lambda*tmp + rex;
   }

   // read in potential table
   // columns are by tension
   // rows are by radius
   this->u = new float*[nr];
   for ( int j = 0; j < nr; ++j ) {
      u[j] = new float[nt];
      for ( int k = 0; k < nt; ++k ) {
         // read potential, convert to kT
         ufile >> u[j][k];
         u[j][k] *= kappa;
      }
   }

   ufile.close();
}

/*********************/
/* destructor method */
/*********************/

potential::˜potential( void ) {
   // clean up
   delete [] tension;
   delete [] r;

   for ( int j = 0; j < nr; ++j )
      delete [] u[j];
   delete [] u;
}

/*********/
/* table */
/*********/

// return a table with interpolated potential values at a given tension

void potential::table( float tau ) {
   lmax = 0.; // maximum interaction distance

   // get interpolated potential ui at tension tau
   float *ui = new float[nr];
   interp( tau, ui );

   // create potential look-up table at tension tau
   for ( int j = 0; j < N; ++j )
      for ( int k = 0; k < N; ++k ) {
         float d = dist[j][k];

         if ( d < rex ) {
            t[j][k] = exp( hard_core );
            continue;
         }

         int m;
         for ( m = 0; m < nr && d > r[m]; ++m );
         // d lies between r[m-1] and r[m]

         if ( m == nr ) {
            t[j][k] = exp( ui[m] );
         }
         else {
            // for points outside rex but before 1st potential value
            if (m == 0) m++;

            // linear interpolation
            float a = ui[m] - ui[m-1];
            float x = (d - r[m-1])/(r[m] - r[m-1]);

            t[j][k] = exp( a*x + ui[m-1] );
         }

         // update the maximum interaction distance
         if ( d > lmax && t[j][k] != 1. )
            lmax = d;
      }
   // end double loop over lattice points

   // check that our lattice is large enough for PBC
   if ( side_len < 2*lmax ) {
      cerr << "ERROR: lattice not large enough"
           << " to safely impose periodic BC"
           << endl;
      cerr << "maximum interaction length: " << lmax << endl;
      cerr << " minimum  lattice size: " << lmax*2 << endl;
      cerr << "requested lattice size: " << side_len << endl;
      exit(-1);
   }
}

/**********/
/* interp */
/**********/

void potential::interp( float tau, float *v ) {
   if ( tau < 0 ) {
      cerr << "Can't do tension value: " << tau << endl;
      exit(-1);
   }

   int m;
   if ( tau > tension[nt-1] )
```

```
      m = nt-1;
   else {
      for ( m = 0; m < nt && tau > tension[m]; ++m );
      // tension[m-1] < t <= tension[m]
   }

   if ( m == 0 || tau == tension[m] )
      for ( int j = 0; j < nr; ++j )
         v[j] = u[j][m];
   else
      for ( int j = 0; j < nr; ++j ) {
         // linear interpolation/extrapolation
         float a = u[j][m] - u[j][m-1];
         float x = (tau - tension[m-1])/(tension[m] - tension[m-1]);

         v[j] = a*x + u[j][m-1];
      }
}

/*******************/
/* write_potential */
/*******************/

void potential::write_potential( const string fname ) {
   ofstream output( fname.c_str() );

   for ( int i = 0; i < N; ++i )
      for ( int j = 0; j < N; ++j )
         output << sqrt( i*i*h*h + j*j*h*h ) << "\t" << t[i][j] << endl;

   output.close();
}
```