High throughput prediction of the long term stability of pharmaceutical macromolecules from short term multi-instrument spectroscopic data

BY

Nathaniel Maddux

Submitted to the Department of Physics and Astronomy
and the Graduate Faculty of the University of Kansas
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

John Ralston, Chairperson

Russ Middaugh

Siyuan Han

Chris Fischer

Matthew Antonik

Erik Van Vleck

Date defended: April 16 2013

The Dissertation Committee for Nathaniel Maddux certifies
that this is the approved version of the following dissertation:

High throughput prediction of the long term stability of pharmaceutical macromolecules from short term multi-instrument spectroscopic data

John Ralston, Chairperson

Date approved: April 16, 2013

Abstract

The field of pharmaceutical chemistry is currently struggling with the question of how to relate changes in the physical form of a macromolecular biopharmaceutical, such as a therapeutic protein, to changes in the drug's efficacy, safety, and long term stability (ESS). A great number of experimental methods are typically utilized to investigate the differences between forms of a macromolecule, yet conclusions regarding changes in ESS are frequently tentative.

An opportunity exists, however, to relate changes in form to changes in ESS. At least once during the development of a new drug, a study is undertaken (at great expense) of the ESS of the drug upon perturbation by multiple manufacturing, formulation, storage and transportation variables. The data acquired is then used to build a model that relates changes in ESS to manufacturing, formulation, storage and transportation variables. It is not common in the pharmaceutical industry, however, to relate changes in comprehensive ESS data sets to comprehensive measurements of changes in macromolecular form.

We bridge the gap between physical measurements of a macromolecule's form and measurements of its long term stability, utilizing two data sets collected in a collaboration between our group at the University of Kansas and a group at the Ludwig Maximilians University in Munich, Germany. The long term stability data, collected by the team in Germany, contains measurements of the chemical and conformation stability of Granulocyte Colony Stimulating Factor (GCSF) over a period of two years in 16 different liquid formulations. The short term

physical data, collected in our lab, is comprised of spectroscopic characterization of the response of GCSF to thermal unfolding.

The same 16 liquid formulations of GCSF were used in each study, allowing us to fit models predicting the long term stability of GCSF from short term measurements. We first apply a novel data reduction method to the short term data. This method selects data in the neighborhood of thermal unfolding transitions, and automates traditional comparative analyses. We then model the long term stability measurements using a linear technique, least squares fits, and a non-linear one, radial basis function networks (RBFN). Using a Pearson correlation coefficient permutation test, we find that many of the fitted results have less than a 1% probability of occurring by chance.

Acknowledgements

I would like to thank my advisers, Profs. John Ralston and Russ Middaugh for their guidance, encouragement and support during my time in graduate school. Prof. John Ralston provided extensive sound advice, and I have deeply enjoyed and will miss our long discussions of physics. I am thankful to Russ for giving me the opportunity to write the Empirical Phase Diagram review article, even though it was to be my first published paper, for suggesting several ideas for projects that easily lead to publishable results, and for the financial support provided by the Macromolecule and Vaccine Stabilization Center. I've been fortunate to have advisers who suggested multiple research projects yet gave me the freedom to pick projects, redefine questions, and develop subsidiary topics.

I would also like to thank the people at the Macromolecule and Vaccine Stabilization Center: Prof. David Volkin, Dr. Sangeeta Joshi, Ilan Rosen, Drs. Lei Hu, Christopher Olsen, Kunal Bakshi and Vidyashankara Iyer, and soon to be Drs. Justin Thomas and Jae Kim. I've never experienced a more cheerful and supportive work environment. Your expertise and assistance have been invaluable to me, and I couldn't have completed this project without you.

I extend thanks to our collaborators at the Ludwig Maximilians University in Munich, Germany: Dr. Ahmed Youssef and Prof. Gerhard Winter. This thesis would not have been possible without your long term stability data set, which required considerable expertise and great effort to generate.

I'm grateful to my family for their love and support. As a small child I was fortunate to have been surrounded by many caring adults and teenagers, all of whom enjoyed my curiosity and encouraged me to explore. I don't think I was ever chastised for disassembling anything. Without my family's influence I certainly wouldn't have chosen to be a scientist.

Finally, I am grateful to my girlfriend Jennifer Peterson and my friends Kunal Bakshi, Davi Serrao, Pedro Pontes, Paul Youk, Drew Muller, Nevin Godfrey, Ashley Fox, and Seth Peterson for making boredom improbable and loneliness impossible.

Contents

1 Introduction and Motivation **1**

1.1 Brief review of formulation of macromolecular biopharmaceuticals 1

1.2 Overview of the dissertation 2

2 Review of the Empirical Phase Diagram (EPD) technique **7**

2.1 Introduction 7

2.2 Review of experimental methods 11

2.2.1 X-Ray Crystallography (XRC) and Nuclear Magnetic Resonance (NMR) 11

2.2.2 Near and Far Ultraviolet Absorbance Spectroscopy (UVAS) 12

2.2.3 Near and Far Ultraviolet Circular Dichroism (CD) 12

2.2.4 Intrinsic and Extrinsic Fluorescence 13

2.2.5 Infrared and Raman Spectroscopy 14

2.2.6 Static and Dynamic Light Scattering (SLS, DLS) 14

2.2.7 Differential Scanning Calorimetry (DSC) 15

2.2.8 High Performance Liquid Chromatography (HPLC) 15

2.2.9 Measurements sensitive to intramolecular dynamics 16

2.2.10 Multi-mode Machines ("Protein Machines") 16

2.3 Data Interpretation Challenges 19

2.4 Search space, protein phase space, and measurement phase space 22

2.4.1 Search space 23

2.4.2 Macromolecule phase space 23

2.4.3 Measurement phase space 24
2.5 Data preprocessing and standardization 25
2.5.1 Data preprocessing 25
2.5.2 Data standardization 26
2.6 The Singular Value Decomposition (SVD) 28
2.6.1 Notation 28
2.6.2 Optimal low dimensional representation 29
2.6.3 The relationship between singular values and data reconstruction error 31
2.6.4 Example 32
2.6.5 History 36
2.7 The Empirical Phase Diagram 36
2.7.1 History 38
2.7.2 Interpretation of Empirical Phase Diagrams 38
2.8 Applications and case studies 40
2.8.1 Selection of stress conditions for excipient screening 41
2.8.2 Finding stabilizing conditions 41
2.8.3 Using *EPDs* to investigate the similarity of two proteins 42
2.8.4 Investigating protein dynamics 42
2.8.5 Evaluating a peptide drug (Pramlintide) 43
2.8.6 Investigating the behavior of larger macromolecular complexes 44
2.8.7 Formulation of Clostridium difficile toxins and toxoids 45
2.8.8 Preformulation screening of norwalk virus-like particles 46
2.8.9 Stabilization of measles virus 47
2.8.10 Investigation of polymeric and liposomal gene delivery systems 49
2.9 Extensions of the technique 50
2.9.1 Maximum use of data 51
2.9.2 Representing more than three dimensions 52

2.9.3 Information management 54
2.9.4 New pharmaceutical applications of *EPDs* 54
2.10 Conclusion 57

3 DART: a new programming language for declarative array transformation 67
3.1 Introduction 67
3.2 Design Strategies 70
3.2.1 Declarative array transformation syntax 70
3.2.2 Generality and Extendability 74
3.2.3 Computational reproducibility 74
3.3 Feature Set 75
3.3.1 Importing and collating data 76
3.3.2 Analysis 77
3.3.3 Visualization 79
3.3.4 Comparison with other data processing tools 79
3.4 Summary 82

4 High throughput generation of Empirical Phase Diagrams 85
4.1 Introduction 85
4.2 Methods 86
4.2.1 Materials 86
4.2.2 High Throughput Spectroscopy 87
4.2.3 Circular Dichroism 89
4.2.4 Steady State Intrinsic Trp Fluorescence 89
4.2.5 Absorbance and Optical Density Measurements 90
4.2.6 Construction of EPDs 90
4.2.7 EPD segmentation 91
4.3 Results and Discussion 93

4.4 Conclusion 98

5 Modeling of long term GCSF stability from short term physical data 106

5.1 The Expensive Process of Developing Useful Drugs 106

5.1.1 Combining Two Strengths 108

5.2 Materials and Methods 110

5.2.1 Long Term Stability Studies 110

5.2.2 Accelerated Stability Studies 112

5.3 Analysis 114

5.3.1 Subtraction of buffer spectra 114

5.3.2 Filtering 114

5.3.3 Determination of transition temperatures 116

5.3.4 Construction of data set to predict 117

5.3.5 Construction of data sets from which to predict stability 119

5.3.6 Prediction methods 122

5.3.7 Leave one out cross validation 125

5.3.8 Estimate of fit likelihood 126

5.4 Results and Discussion 127

5.5 Conclusion 132

Appendix A - DART script used in Chapter 5 135

Appendix B - Regression functions used in Appendix A 164

List of Figures

2.1 An empirical phase diagram (*EPD*) assists in the visualization of data set resulting from the methods listed in Table 1. Figures A-F show measurements of an IgG1 monoclonal antibody collected at pH 3 (black), 4 (red), 5 (green), 6 (yellow), 7 (blue), and 8 (magenta). (A) CD molar ellipticity at 218 nm, (B) UV intrinsic fluorescence (UV-IF) peak position and (C) intensity, (D) tryptophan fluorescence lifetime, (E) static light scattering (SLS), and (F) ANS extrinsic fluorescence (ANS-EF) intensity. Error bars in (A-C and E-F) are from three independent experiments.[4] Figure (G) shows an *EPD* based on the above data. Figure (H) shows an *EPD* based on protein dynamics measurements (data not shown, see the applications section for more information). 20

2.2 Empirical phase diagrams have found many uses in the optimization of various types of formulations. Many case studies have been published concerning their application to various systems and their extension by the addition of measurement techniques and search space variables. Refer to Table 2.2 for more information concerning each *EPD*. All *EPDs* have temperature (°C) as the vertical axis. Diagrams 1-35 use pH on the horizontal axis, and diagrams 36-40 have the indicated variables on the horizontal axis. 21

2.3 An illustration of three types of spaces, using simulated data. In Figure (A), four pH values define a one dimensional search grid in *search space,* and ratios of secondary structure type illustrate a *protein phase space.* Figure (B) shows how two measurement types define a *measurement phase space.* The transition pH values disagree when we plot measurements separately, as in Figures (C) and (D). A plot in measurement phase space (B) synthesizes the information, but will not work as a visual aid for high dimensional data. 24

2.4 Principal Components Analysis can be used to project two dimensions into one. The procedure works the same way for high dimensional data (see Figure 2.5 and 2.6). We will use the simulated data shown in Figure 2.3B. (A): First we center the measurements at the origin, since we are interested in transitions, not average values. (B): Next we normalize each measurement so that they have equal influence on the result. (C): Finally, we use the Singular Value Decomposition (SVD) to find the optimal line for projection (shown in blue). (D): If we plot the position along the blue line, we see that the difference is greatest between pH values 5 and 6. 27

2.5 An example of the Singular Value Decomposition (SVD) using simulated data. (A) is a plot of three simulated peak shifts $\Delta\lambda_1$, $\Delta\lambda_2$, and $\Delta\lambda_3$ as a function of temperature. If we could perceive two dimensions but not three, the transition between 50°C and 70°C might be difficult to see. Therefore, we would want to reduce the data to two dimensions in a way that optimally retains the information in the original data set. (B) shows the plane (in pink) which gives the optimum 2D projection. This plane is determined by SVD, and is defined by the vectors X_1 and X_2 (in blue). The projection error is shown as red lines. (C) is a 2 dimensional plot of the same data, using the positions within the pink plane. This is a plot of matrix A (see text). (D) shows the optimal one dimensional projection, demonstrating that the error is larger. This plot uses the first column of matrix A (see text) 33

2.6 Illustration of the steps in the Empirical Phase Diagram method, using simulated data. (A): Choose a search space and a search grid. In this case, the search space is 2 dimensional, varying temperature and pH in this case. In each dimension, two values have been chosen, forming a grid. (B): Collect data at each point of the search grid. The data in this example is 5 dimensional. (C): Standardize the data and project it into 3 dimensions. (D): Rescale to the range $(0, 1)$, and express as a color. (E): Transform the colors into an image. 37

2.7 Empirical phase diagrams (*EPDs*) of the peptide drug pramlintide at low and high concentration,[7] and concentration dependence at pH 4. Low concentrations (0.088 mg/ml) are represented in A-C. The experimental techniques used to construct A-C were as follows: (A) second derivative UV absorbance peak shift and OD_{350}, (B) same as (A), adding fluorescence intensity and peak shift, (C) same as (B), adding the CD change at 204 nm. The peptide at high concentration (8.8 mg/mL) is represented in (D), using the same experimental techniques as (B). An *EPD* at pH 4 as a function of peptide concentration is shown in (E), using the same experimental techniques as (B). 43

2.8 An empirical phase diagram for two toxins and toxoids of *Clostridium difficile*, created using OD_{350}, UV-IF, ANS-EF, and CD data.[16] Data were normalized simultaneously for the corresponding toxin and toxoid. (A) Toxin A; (B) Toxin B; (C) Toxoid A; (D) Toxoid B 45

2.9 Empirical phase diagram for Norwalk virus-like particles (NV-VLPs) based on UV absorbance, intrinsic and extrinsic fluorescence and CD results.[23] Four distinct phases (P) of the NV-VLP were observed: P1, native, intact form; P2, disassembled; P3, soluble VP1 oligomers; P4, aggregated. The nature of the protein in the various phases was confirmed by transmission electron microscopy studies. 47

2.10 Empirical phase diagram of attenuated Measles virus.[24] Data used to generate the *EPD* were measurements of mean effective diameter by DLS, intensity of 562 nm light scattered at 90°, CD at 222 nm, intrinsic fluorescence intensity at 322 nm, ANS peak position, ANS fluorescence intensity at 469 nm and generalized polarization of laurdan fluorescence. 48

2.11 Ionic strength-pH empirical phase diagrams of various nonviral gene delivery complexes formed between plasmid DNA and four cationic carriers.[28] Each *EPD*has pH as the horizontal axis and ionic strength (mM) as the vertical axis. The experimental techniques used were DLS, CD, and YOYO-1 EF. . 49

2.12 When the projection error is large it can be reduced by incorporating more dimensions. In (A)-(C), we show the primary color images (red, green and blue) of an empirical phase diagram. They are ordered by descending significance from left to right. For axis information, see (E) and (F). After solid red, green, and blue, we can use images containing structure that is smaller than the individual phase diagram blocks. This will represent high dimensional information as changes in texture. Such an image is shown in (D). (E) is a 3 dimensional empirical phase diagram of IgG, using FTIR spectra which have been preprocessed with a Fourier filter to emphasize mid-size spectral features. (F) is a 4 dimensional empirical phase diagram of the same data as (E), showing fourth dimensional information as changes in texture. Notice that the reconstruction error has decreased. The diagram has also been automatically segmented into 5 parts (see text). 50

4.1 Overhead line drawing of the Olis Multiscan (OM), a cuvette-based spectrophotometer that measures circular dichroism, absorbance, and fluorescence with high level photometric and wavelength accuracy and repeatability. . . 92

4.2 Representative biophysical measurements of 4 model proteins collected with the OM. The data includes second derivative near UV absorbance, circular dichroism, intrinsic Trp fluorescence and optical density measurements for the model proteins aldolase, BSA, chymotrypsin, and lysozyme. For error bars, see the supplemental figures. The units for the Y-axis of each column are, respectively: absorbance unit/nm2 , °cm2/dmol, nm, number of photons, emission peak shift per excitation wavelength change (nm/nm), and optical density units. The line colors are black-pH 3, red-pH 4, green-pH 5, blue-pH 6, purple-pH 7, and orange-pH 8. Each figure has temperature (°C) as the horizontal axis. 95

4.3 Empirical phase diagrams of 4 model proteins: (a) aldolase, (b) BSA, (c) chymotrypsin, and (d) lysozyme. These EPDs summarize the representative biophysical data from Figure 2 (in conjunction with wavelength measurements using the full spectra; see text) and display protein structural responses to temperature and pH perturbations. The experimental techniques used were second derivative near UV absorbance from 275 to 295 nm (full spectra), fluorescence spectra from 315 to 370 nm (full spectra), far UV CD at 217, 222, and 235 nm, and the mean optical density from 320 to 340 nm. 96

4.4 Error bars associated with biophysical measurements of Aldolase collected with the OM. The units are for each column are, respectively: absorbance unit/nm2 , °cm2/dmol, nm, number of photons, emission peak shift per excitation wavelength change (nm/nm), and optical density units. Each figure has temperature (°C) as the horizontal axis. 99

4.5 Error bars associated with biophysical measurements of BSA collected with the OM. The units are for each column are, respectively: absorbance unit/nm2 , °cm2/dmol, nm, number of photons, emission peak shift per excitation wavelength change (nm/nm), and optical density units. Each figure has temperature (°C) as the horizontal axis. 100

4.6 Error bars associated with biophysical measurements of Chymotrypsin collected with the OM. The units are for each column are, respectively: absorbance unit/nm2 , °cm2/dmol, nm, number of photons, emission peak shift per excitation wavelength change (nm/nm), and optical density units. Each figure has temperature (°C) as the horizontal axis. 101

4.7 Error bars associated with biophysical measurements of Lysozyme collected with the OM. The units are for each column are, respectively: absorbance unit/nm2 , °cm2/dmol, nm, number of photons, emission peak shift per excitation wavelength change (nm/nm), and optical density units. Each figure has temperature (°C) as the horizontal axis. 102

5.1 Correlation plot of the long term stability of 24 GCSF formulations versus thermal transition midpoints measured by dynamic scanning calorimetry (DSC). The long term stability and DSC measurements of the formulations were each ranked 1 to 24 by a procedure described in the text. The horizontal and vertical axii of the plot are, respectively, the long term stability and DSC rankings.(Reprinted from the European Journal of Pharmaceutics, Ahmed M K Youssef and Gerhard Winter, "A critical evaluation of microcalorimetry as a predictive tool for long term stability of liquid protein formulations: Granulocyte Colony Stimulating Factor (GCSF)", In Press, Copyright 2013, with permission from Elsevier) 113

5.2 Plots of short term spectroscopic measurements for formulation 11. The buffers presented a relatively large spectral signal. A small part of the signal in the buffers is presumably due to the buffer itself. The signal is much larger than what would usually be attributable to buffer alone, however, and may be partly due to protein contamination. The band near 60 °C in the ANS fluorescence plot is discussed in the text. 115

5.3 Derivatives of intrinsic fluorescence melts for formulations 13 and 14. The melts with no derivative are the fluorescence intensity averaged over wavelength. The third derivative of a melt results in peaks corresponding to the transition onset, midpoint, and endset. The positions of these peaks were determined with a peak finding algorithm, and the resulting transition temperatures are shown in Table 5.3. 118

5.4 Plots of spectral melts that have been restricted to a range of temperatures around the thermal transition temperatures. Each temperature axis zero value corresponds to the thermal transition temperature for the associated formulation. 123

5.5 A measure of goodness of fit, the Pearson correlation coefficient, compared to a null distribution constructed for the measure. The histogram shows a null distribution for the Pearson correlation coefficient between predictions and observations of chemical stability of GCSF. Chemical stability was predicted from thermal melts using a least squares fit. The null distribution was generated using 5000 permutations of observations of the chemical stability. The red line is the Pearson correlation coefficient between predictions and observations, showing a fit likelihood of 2.7 sigma. See the text for a more complete description. 127

5.6 Correlation plots of fits resulting from leave-one-out cross validation. The plot labels indicate the measurement predicted, the data used to predict from, and the prediction method. Each point corresponds to a formulation. The horizontal and vertical axii correspond to observed and predicted measurements, respectively, and extend over the same range. The likelihood of each fit was estimated using the Pearson coefficient permutation test, and the 16 best fits are shown here in descending order. 129

List of Tables

2.1 Lower resolution biophysical techniques commonly used to characterize and monitor higher order structure as well as aggregates of biomolecules and macromolecular complexes. 18

2.2 Biomolecules and larger macromolecular complexes, analytical techniques and environmental stress conditions evaluated by empirical phase diagrams. See Table 2.1 for definitions of the technique abbreviations. 40

3.1 A comparison of the capabilities of various array processing solutions. 81

5.1 Formulation parameters for 16 formulations of GCSF in a factorial design of experiment. 111

5.2 Long term stability measurements for 16 formulations of GCSF. Measurements are shown with 2 digits of precision. See text for details. 112

5.3 Transition temperatures that were determined automatically using the derivative method. See Figure 5.3 for a summary of the method. 117

5.4 Fit likelihood values of fits resulting from leave-one-out cross validation for the least squares prediction method. Refer to Figure 5.5 and the text for information on how the fit likelihoods were determined. Significance values greater than 2 are highlighted in yellow, and those greater than 3 are highlighted in green. 128

5.5 Fit likelihood values of fits resulting from leave-one-out cross validation for the radial basis function network prediction method. Refer to Figure 5.5 and the text for information on how the fit likelihoods were determined. Significance values greater than 2 are highlighted in yellow, and those greater than 3 are highlighted in green. 130

Chapter 1

Introduction and Motivation

1.1 Brief review of formulation of macromolecular biopharmaceuticals

Proteins, plasmid DNA and macromolecular complexes such as viruses, virus-like particles and adjuvanted antigens are used to treat or prevent conditions as diverse as growth deficiencies (Humatrope), cancer (Avastin), hemophilia (NovoSeven), viral illnesses (Recombivax HB), stroke (Activase) and cystic fibrosis (Pulmozyme).[1] Since approval of recombinant human insulin in 1982, over 100 protein drugs have been introduced into clinical practice.[1] Annual sales of biopharmaceutical proteins in 2009 totaled $99 billion,[2] and future growth has been estimated at 7 to 15% annually for the next several years.[3]

Determining and preserving the structural integrity and conformational stability of macromolecular biopharmaceuticals is frequently a significant barrier to the successful stabilization and formulation of a biopharmaceutical drug or vaccine. The core question being asked in such studies is easily stated: will a particular form of a macromolecular biopharmaceutical provide required levels of efficacy, safety and stability?

During the course of manufacturing and formulating a biopharmaceutical drug, the need to compare forms of the drug occurs hundreds of times. Consequently it is too costly and slow to perform full efficacy, safety and stability tests each time, including storage for 2-3

years in clinically relevant storage conditions, followed by animal and possibly human testing.

The comparison of biopharmaceuticals thus generally proceeds by use of fast, inexpensive substitute measurements: forced degradation in place of long term storage at clinically relevant conditions, animal testing in place of human testing, or blood derived assays in place of human testing. The experimental techniques may include spectroscopic measurements of the system's resistance to thermal degradation, chromatographic investigation of aggregated and cleaved forms of the macromolecule after forced degradation by exposure to light, shaking, or elevated temperature, or the molecule's biological activity as measured by blood derived assays. Specific experimental techniques include spectroscopic techniques such as circular dichroism, absorbance, raman scattering and fluorescence spectroscopy. Also included are chromatographic techniques such as size exclusion high performance liquid chromatography (SE-HPLC), reverse phase high performance liquid chromatography (RP-HPLC), mass spectrometry, or gel electrophoresis. These measurements yield information concerning a great number of physical and biological phenomena. In addition, a biopharmaceutical's response to a large variety of perturbations must be tested. These include manufacturing process variables, storage and transportation conditions and the effects of various formulation additives (called excipients) and their combinations. The resulting parameter spaces are vast and thus usually only explored sparsely.

1.2 Overview of the dissertation

Due to the size and complexity of data sets acquired during the testing of biopharmaceuticals, it is often difficult to extract, interpret and summarize the information obtained. In Chapter Two we review a method developed in our lab that represents the state of the art in visualization and comparative analysis of measurements of biopharmaceutical drugs. The Empirical Phase Diagram (EPD) technique is a vector-based multidimensional analysis method for summarizing large data sets resulting from a variety of biophysical techniques. It

can be used to provide comprehensive preformulation characterization of a macromolecule's higher-order structural integrity and conformational stability. In its most common mode, it represents a type of stimulus-response diagram using environmental variables such as temperature, pH, and ionic strength as the stimulus, with alterations in macromolecular structure being the response.

In Chapter Three we describe a new programming language for processing and visualizing arrays that is suitable for use by non-experts yet covers the range of capabilities necessary for complex analysis of array data. Data processing techniques of considerable flexibility and complexity are required to perform comparative analyses of macromolecules using multiple experimental techniques to investigate the influence of multiple perturbing factors. Analyzing these data sets requires operations such as mapping filenames to dimension positions, discarding noisy or corrupt data, handling data that is non-existent due to instrument glitches and operator error, subtracting reference spectra, performing statistical and signal processing operations, generating complex plots, and performing original data analysis research. These tasks are typically performed using a combination of many programs such as Excel, Origin, instrument software, Matlab, Mathematica and custom scripts. Even for modest size data sets, the data analysis is tedious and requires much labor. Data analysis is often the bottleneck in high throughput pharmaceutical experiments. Furthermore, the unstructured nature of the process make it difficult to document.

Thus the increasing size and complexity of pharmaceutical formulation data sets has created the need for a tool to simplify, automate and document the processing of data. This tool should be capable of advanced multidimensional data processing, yet remain simple enough to be used by non-programmers. It should also provide built in support for almost any math operation that might be desired, provide a record of changes to data, and ease the task of regularizing data sets filled with inconsistent array shapes and missing data.

We call our solution the Declarative Array Transformation language, or DART. Various novel techniques allow it to approach a natural language, and scripts written with it tend

to be lists of one liners without control structures. Literate programming facilities help ensure that persons other than the authors of an analysis can reproduce computations in that analysis. The resulting language has broad applicability and a high level of versatility.

In Chapter Four we apply DART to the generation of EPDs using data from a new robotic instrument. Until now EPD analysis has not been available in a high throughput mode because of the large number of experimental techniques and environmental stressor/stabilizer variables typically employed. A new instrument has been developed that combines circular dichroism, UV-absorbance, fluorescence spectroscopy and light scattering in a single unit with a 6-position temperature controlled cuvette turret. Using this multifunctional instrument and DART we have generated EPDs for four model proteins. Results confirm the reproducibility of the apparent phase boundaries and protein behavior within the boundaries. This new approach permits two EPDs to be generated per day using only 0.5 mg of protein per EPD. Thus, the new methodology generates reproducible EPDs in high-throughput mode, and represents the next step in making such determinations more routine.

In Chapter Five we face the original question head on using the tools developed in this project. We ask: will a particular form of a macromolecular biopharmaceutical provide required levels of efficacy, safety and stability? Although much effort has been expended by the pharmaceutical chemistry community in pursuit of an answer to this question, it has been difficult to answer definitively using inexpensive substitute measurements alone, as these are only loosely related to the information derived from full testing.

We proceed with an interpretation of the question in which general rules regarding relationships between direct and indirect measurements of efficacy, safety and stability are replaced by mathematical models developed and validated from measurements. Thus the form of a macromolecular biopharmaceutical is defined by physical measurements of the drug, and the drug's levels of efficacy, safety and stability are defined by measurements of those properties. The question then becomes: how does one relate physical measurements of a macromolecule's form to measurements of the drug's efficacy, safety and stability?

We apply a new data reduction technique that is based on the idea that the behavior of proteins near thermal transitions, in particular the relative behavior of different types of protein structure, provides information about unfolding mechanisms. The technique is a way of automating traditional comparative short term structural analyses.

We then apply a linear technique and a nonlinear technique to predict the long term stability measurements of 16 formulations of a protein drug. A data set of this size or larger is commonly generated once during the formulation of a protein drug. Traditionally, however, the long term stability of a drug is modeled from formulation parameters. We instead model long term behavior from short term form and behavior, allowing us to develop predictive models that answer the core question of the thesis.

Bibliography

[1] David M Dudzinski and Aaron S Kesselheim. "Scientific and legal viability of follow-on protein drugs". In: *New England Journal of Medicine* 358.8 (2008), pp. 843–849.

[2] Gary Walsh. "Biopharmaceutical benchmarks 2010". In: *Nature biotechnology* 28.9 (2010), p. 917.

[3] A Hiller. "Fast growth foreseen for protein therapeutics". In: *Gen* 29.1 (2009), pp. 153–155.

Chapter 2

Review of the Empirical Phase Diagram (EPD) technique

2.1 Introduction

We now review a method that represents the state of the art in visualization and comparative analysis of measurements of biopharmaceutical drugs. It was developed from scratch by the collaborators on this project and is used in the interpretation of multidimensional data arising in the pharmaceutical formulation of proteins and other macromolecules. This chapter contains a review of the experimental methods commonly used in the optimization of formulations, a discussion of the challenges of interpreting multidimensional data, an overview of the EPD technique, and discussions of common applications of the technique and extensions to the technique.

The pharmaceutical uses of proteins, nucleic acids and higher order macromolecular complexes such as viruses, virus-like particles, plasmid DNA and polymer associations, and adjuvanted antigens represent the major advance in the biotechnology and vaccine industries in the last 30 years. Due to their more natural biological character, macromolecules offer a degree of safety and efficacy that has resulted in their continuously increased use for a wide variety of therapeutic and prophylactic applications.

Traditional analytical methods of ensuring the structural integrity and conformational

stability of these macromolecules have not, however, kept up with this progress. For example, due to the inability of individual experimental methods to monitor all aspects of the structural integrity of macromolecules, biological potency assays are required to ensure overall structural properties have been maintained. Moreover, in the case of protein-based drugs including monoclonal antibodies, loss of conformational integrity leading to aggregation during manufacturing and storage has raised potential safety concerns due to immunogenicity.[1,2] This problem has become especially acute not only in terms of defining shelf life and ensuring proper administration, but it arises frequently as a comparability issue during the biopharmaceutical drug development process. For example, some of the challenges of establishing analytical comparability for different monoclonal antibodies during early and late stage development have recently been highlighted.[3] With the advent of biosimilars, the ability to better define the higher order structure of proteins, nucleic acids, and macromolecular complexes in pharmaceutical dosage forms over time will most likely emerge as a critical analytical challenge.

Because the more complex three dimensional structures of macromolecules (typically involving tens of thousands of atoms or more) often play the key role in defining their biological activity and efficacy, characterization of higher order secondary, tertiary and quaternary structures remains a significant barrier to their pharmaceutical development. The problem is simple enough to state, although it is remains difficult to address experimentally: How does one demonstrate that pharmaceutical macromolecular systems are sufficiently structurally similar (at the beginning and end of shelf life or in comparison to an analogous macromolecular system) that they can for all intents and purposes be considered sufficiently identical for therapeutic use in terms of their safety, efficacy, and stability?

A number of standard methods currently exist with the ability to obtain high resolution structural information for proteins, nucleic acids and their complexes, resulting in commonly used representations such as stick and ball models, ribbon diagrams and van der Waals and electrostatic surface maps. Such three dimensional images of structure are the most

common way to think of macromolecular systems. Among the experimental methods used to generate these images are X-Ray crystallography, nuclear magnetic resonance (NMR), cryo-electron microscopy and molecular mechanics calculations based on detailed force potentials. At present, however, these approaches are seldom directly applicable to biopharmaceutical dosage forms due to practical limitations. For example, X-Ray crystallography requires crystallization, while NMR spectroscopy requires isotopic labeling and high concentrations. Moreover, complete structural characterization is most appropriate when it serves the overall goal of developing formulations. For these reasons, lower resolution biophysical methods are commonly employed to monitor structural integrity and hydrodynamic properties. These techniques include circular dichroism (CD), fluorescence, differential scanning calorimetry (DSC), chromatography, and light scattering, among others (see Table 2.1).

Unfortunately, no one method provides sufficient information to establish the identity and integrity of complex macromolecular systems. Therefore, the use of more than one of these methods is generally preferred to better characterize these entities. The multidimensional nature of such data sets makes adequate characterization of higher order structural integrity problematic. To develop stable dosage forms, formulation scientists typically analyze stress-induced transitions in macromolecular structure under varying solution conditions in the presence of different excipients by using techniques that look at the data locally, using, for example, visual inspection and/or mathematical fitting of thermal unfolding curves to sigmoidal functions. Unfortunately, the global features of high-dimensional data spaces are not always revealed by such local data inspection. A more comprehensive analysis of the complex behavior typically observed is clearly desirable.

We review here the use of a newly formulated global mathematical analysis technique developed for evaluation of large data sets generated from the biophysical analysis of biopharmaceuticals and vaccines. The mathematical methodology finds and quantifies multidimensional regularities in the data sets that often are difficult to detect with local inspection. The mathematical information is converted into a visual map that serves to better define

and investigate structural integrity and conformational stability of biomolecules and macromolecular complexes.

From the dozens of test cases to date, we find that these maps tend to be segmented into regions of distinct structural behavior. We call areas of a single contiguous color on these maps "apparent" phases, and the related diagram an *empirical phase diagram* (*EPD*). The word "empirical" serves to distinguish the diagrams from thermodynamic phase diagrams, in which the phase transformations are necessarily reversible. In spite of a common lack of reversibility in many protein transformations, the word "phase" to describe a physically distinctive form of a substance reasonably applies to a pharmaceutical usage, as described in more detail below.

An example is shown in Figure 2.1 of a representative data set generated for a monoclonal antibody (IgG1), along with the resulting empirical phase diagram. Various analytical methods were used to monitor both the structural integrity as well as the dynamic properties of the immunoglobulin as a function of temperature and solution pH.[4] These data sets are then summarized for analysis in the form of an empirical phase diagram. This approach has been applied widely by our laboratory to different proteins, plasmid DNA-lipid complexes, virus like particles and viruses. As shown in Figure 2.2, dozens of empirical phase diagrams have been generated and published over the past 7-8 years. Refer to Table 2.2 for references and more detailed information concerning each empirical phase diagram.

Our group's *EPD* method has found many uses in the development and optimization of various types of biopharmaceutical and vaccine formulations. Empirical phase diagrams serve as guides to the interpretation of multidimensional data, determining regularities that may be difficult to visualize otherwise. These data sets are presented in an easy to inspect format, assisting in the determination of protein state and transition points as a function of environmental conditions such as temperature and solution pH. Many case studies have been published concerning not only the application of *EPDs* to various macromolecular systems, but also their extension by the addition of new biophysical measurement techniques and

search space variables. Common pharmaceutical applications have been to aid in selecting stress conditions for excipient screening, finding optimal ranges of stabilizing solution conditions, and investigating the overall physical behavior of large macromolecular complexes. *EPDs* have been applied to the characterization, stabilization and formulation of proteins,[4–21] virus like particles,[22,23] viruses,[24–27] and nucleic acids and their complexes with lipid delivery vehicles,[28] as well as whole bacterial cells.[29] In principle, one can incorporate almost any kind of information into *EPDs*, including measurements of structural dynamics, chemical integrity or biological function. Empirical phase diagrams have also been shown to contain information concerning the functional and evolutionary relationships of proteins.[10–12,16,17] These applications will be discussed in more detail below.

2.2 Review of experimental methods

2.2.1 X-Ray Crystallography (XRC) and Nuclear Magnetic Resonance (NMR)

Since XRC and NMR have the potential to determine the full three dimensional structure of macromolecules, they would be ideal were it not for confounding factors. Both methods require costly instrumentation and highly trained support staff. XRC requires the preparation of crystals, which cannot always be grown, and do not necessarily represent structure in the solution state. The experimental procedure typically takes at least days to weeks to optimize and perform. Full structure determination by NMR currently takes a similar length of time, but only works for small to medium size proteins (thus not including monoclonal antibodies). Furthermore, isotopic labeling is necessary for full structural determination. These limiting aspects of NMR may, however, be reduced in the future.[61,62] Both methodologies are also difficult to apply to pharmaceutical dosage forms due to interfering effects of excipients. The goal in the work described here is primarily to find transitions in higher order structure as a function of environmental conditions (e.g. temperature and pH in the

presence of different excipients), which requires far less information than that required for full structure determination.

A wide variety of lower resolution biophysical techniques are available for characterization of biomolecules and their macromolecular complexes. In general, these methods can be employed over a wide range of concentrations (from a few micrograms to hundreds of milligrams per milliliter), although interference by factors such as light scattering, absorbance flattening and solute interference can sometimes be a problem.

Very brief descriptions of many of these techniques now follow. References and a summary of the capabilities of each method are shown in Table 2.1.

2.2.2 Near and Far Ultraviolet Absorbance Spectroscopy (UVAS)

Both proteins and nucleic acids contain a number of environmentally sensitive chromophores which absorb in the UV region. While the peptide bonds of proteins display intense absorbance in the far UV (180-220nm) region, thus yielding secondary structure information, analysis in this region is normally done by circular dichroism or FTIR due to their better resolution (see below). In contrast, derivative analysis of protein spectra in the near UV typically provides 5 to 6 well resolved peaks from the three aromatic residues (Trp,Tyr,Phe), which are quite sensitive to structural changes. Nucleic acids also produce distinct spectra from the bases in the same spectral region, which can be used to follow structural alterations. Conveniently, when a macromolecular system aggregates, optical density (OD) in non-absorbing regions (>340nm) can be used to monitor this phenomenon simultaneously with near UV spectral analysis.

2.2.3 Near and Far Ultraviolet Circular Dichroism (CD)

Due to the high optical activity of helical structures, CD can be used to detect changes in both nucleic acid and protein secondary structure in the far-UV region for proteins and mid-UV region for nucleic acids. The optical asymmetry of the environment of the aromatic side

chains in proteins also produces distinct signals typically of some complexity in the near-UV region. Thus by monitoring both regions, structural changes in secondary and tertiary structure can be detected. Deconvolution analysis of CD spectral shape in the far UV region also allows fairly accurate estimates (within 2-3%) of actual secondary structure content. The induced CD of certain dyes can also be used to determine structural information, especially with nucleic acids and polysaccharides.

2.2.4 Intrinsic and Extrinsic Fluorescence

The intrinsic UV fluorescence (UV-IF) of proteins is dominated by emission from indole side chains when Trp residues are present and not endogenously quenched. Such fluorescence is very environmentally sensitive, making the peak position and intensity of Trp fluorescence a particularly useful probe of protein structural change. The use of extrinsic fluorescence (EF) probes is applicable to virtually all forms of macromolecules and their complexes, including proteins, nucleic acids, and membranes. For example, dyes are available which are particularly attracted to apolar regions in proteins as well as the characteristic intermolecular β-structures which often form when proteins associate. A wide variety of fluorophores bind both within nucleic acid grooves as well as between bases (intercalation). In addition, there exist a large number of dyes that interact with lipid bilayers such as those present in some viruses and virus-like particles as well as bacterial cells. Some of the most commonly used dyes are 8-anilino-1-naphthalenesulfonate (ANS), used in protein studies; laurdan, used for lipid bilayers; and YOYO-1, used for DNA. In all of the above cases, large changes in fluorescence intensity, peak position, and polarization often occur as these dyes bind to their various targets. Thus, they can be used to probe a plethora of aspects of macromolecular structure and associated changes.

2.2.5 Infrared and Raman Spectroscopy

Both infrared and Raman spectroscopy can be used to obtain structural information about the complex series of vibrational transitions present in macromolecules. Infrared spectroscopy is performed almost exclusively in a Fourier transform mode (FTIR). While FTIR is an absorptive technique and Raman is a scattering measurement, both have significant although sometimes different utility. Each can be used to examine the secondary structure of both proteins and nucleic acids (as well as complexes such as viruses) through deconvolution of constituent amide bands (signals from peptide bonds and various nucleic acid base signals). FTIR is the more widely used technique due to instrument availability and sensitivity. In contrast, signals from side-chains tend to be much better detected in Raman spectra.

2.2.6 Static and Dynamic Light Scattering (SLS, DLS)

The size and shape of macromolecules both in their monomeric and associated forms can be characterized by static and dynamic light scattering. In the former, the intensity of the scattered light is measured (often as a function of angle), while in the latter, fluctuations in intensity of scattered light due to Brownian motion are analyzed. Size and shape information obtained are model dependent and complicated by the presence of non-homogeneous scatterers, although various data analysis methods exist to produce useful numerical values from both methods. Imposition of an external electromagnetic field can be used to obtain zeta-potential values. A method we do not discuss here is analytical ultra-centrifugation (AUC). Although AUC is very information rich in terms of evaluating hydrodynamic properties of biomolecules and macromolecular complexes, this methodology is not available in a high throughput mode, unlike the scattering based methods.

2.2.7 Differential Scanning Calorimetry (DSC)

Differential scanning calorimetry is a technique based on measuring differential heat capacities in macromolecules, from which transitions in state can be detected. Virtually every biomolecule from proteins and nucleic acids to membranes and viral particles undergo thermally induced transitions that can be detected by this method and used as indicators of thermal stability. Like the methods described above, DSC is now available in a high throughput mode making it useful for the formulation and stability purposes discussed below.

2.2.8 High Performance Liquid Chromatography (HPLC)

High Performance Liquid Chromatography is a technique which passes a solution at high pressure through a filter column. Depending on physical traits of the filter media, different molecules in the solution pass through the media at different speeds. Thus a chromatogram can be constructed by measuring the time-varying absorbance of the solution exiting the filter column. Although not generally adaptable to a high throughput mode in the sense of the above methods (i.e. one cannot easily and rapidly perform measurements over a wide range of pH and temperatures), the use of auto-samplers does permit a variety of chromatographic methods to be used after exposure to a wide range of conditions. Probably the three most useful to the formulation scientist are size-exclusion (SEC), ion-exchange (IE), and reversed phase (RP) chromatography. All three methods will be well known to most readers so we just mention their applicability to size, charge, and polarity changes, respectively. To characterize chemical degradation (oxidation, deamidation, hydrolysis, etc.), RP-HPLC is commonly used in combination with fragmentation and mass spectrometry to characterize sites of covalent alteration. Methods such as capillary isoelectric focusing are also commonly used for this purpose.

2.2.9 Measurements sensitive to intramolecular dynamics

It has become increasingly apparent that macromolecular stability is dependent on the various types of internal molecular motions present in macromolecular systems, such as side-chain movements, breathing modes, domain motions, etc. Thus, measurements of such motions should ultimately be included in a thorough analysis of stability. A number of high-throughput methods are available, including ultrasonic spectroscopy (to measure compressibility), pressure perturbation DSC (to measure coefficients of thermal expansion), as well spectral approaches such as temperature induced pre-transition peak shifts in second derivative UV absorbance spectra, fluorescence anisotropy (rotational correlation times), red-edge fluorescence excitation, and fluorescence and UV absorbance solute-induced spectral shifts. Methods specifically designed for this purpose such as isotope exchange and various forms of NMR are not generally applicable to high-throughput applications, although this may change in the future.

2.2.10 Multi-mode Machines ("Protein Machines")

Instruments are currently being developed by several vendors that simultaneously collect data using several of the above methods. For example, the Chirascan from Applied Photophysics collects near and far UV CD and near and far UV absorbance. Fluorescence emission spectra can also be collected, although not simultaneously with the other techniques. The Protein Machine from Olis Instruments collects far UV CD, near UV absorbance, fluorescence emission and excitation spectra, and red-edge excitation spectra. Both instruments can also acquire light scattering signals during several of these measurements.

Difficulty of simultaneously measuring near and far UV signals

Simultaneous near and far UV measurements require intermediate path lengths and concentrations. Longer path lengths or higher concentrations yield excess absorbance, causing absorbance flattening in far UV measurements. Shorter path lengths or lower concentrations

yield too little signal, resulting in a significant amount of noise in near UV measurements. It is not possible in principle to find an optimum trade off between path length and concentration, because both have the same effect on absorbance. Changing slit widths can overcome these problems to only a very limited extent. Thus, the existence of conflicting requirements makes it technically difficult, but not impossible, to simultaneously collect data in the near and far UV regions. In the far UV region, peptide bonds yield very strong absorbance. To avoid absorbance flattening in this region one must use short path lengths or low concentrations. The near UV absorbance spectra of aromatic residues are comparatively weak, so short path lengths or low concentrations result in noisy measurements. Nevertheless, these instruments do permit simultaneous collection of data from multiple techniques with good to excellent resolution. In combination with multiple sample holders, *EPDs* can be obtained directly from such instruments over periods of 3-12 hours.

Currently, the only way to simultaneously collect data in the near and far UV is to use very long integration times in the near UV, to reduce excessive noise. These long integration times offset the time saved by simultaneous collection. Short of waiting for instruments with lower noise to be developed, there is at least one possible option to be considered: variable path length cells would permit automatic adjusting of absorbance for each wavelength range. This feature is available in a few UV-Vis absorbance instruments built for measuring concentrations, and could potentially be applied to multi-modal spectrometers.

Table 2.1: Lower resolution biophysical techniques commonly used to characterize and monitor higher order structure as well as aggregates of biomolecules and macromolecular complexes.

Method	2° Structure Ratios	2° Structure Transitions	3° Structure Transitions	4° Structure Transitions	Aggregate Presence	Aggregate Size	Aggregate Population[d]	Dynamics	References
Near[a] UV Absorbance (UVAS)		○	●		●			●	4,6,18,30–33
Far[b] UV Absorbance	●	●	●						30
Near[a] UV Circular Dichroism (CD)			●						34,35
Far[b] UV Circular Dichroism	●	●	○						33–36
Intrinsic Fluorescence (IF)		○	●					●	37
Extrinsic[c] Fluorescence (EF)			●		●				38–43
Red Edge Excitation (REES)								●	44,45
Time Resolved Fluorescence (TRFS)								●	46,47
Fourier Transform Infrared (FTIR)	●	●			●				48–50
Raman spectroscopy (RS)	●	●	●						51
Differential Scanning Calorimetry (DSC)		●	●	●	●				52,53
Pressure Perturbation Calorimetry (PPC)								●	53–55
High Res. Ultrasonic Velocimetry (HRUS)								●	53,56
Dynamic Light Scattering (DLS)				●	●	●	●		57
Static Light Scattering (SLS)				●	●	●			58
Optical Density (OD)				●	●				58
High Perf. Liquid Chromatography (HPLC)		●	●	●	●	●	●		59,60

[a]240-320nm. [b]190-260nm. [c]Die conjugated. [d]Size distribution profile. ○ Limited data.

2.3 Data Interpretation Challenges

A wealth of data tends to be generated when several of the above methods are employed under varying environmental conditions. Figure 2.1A-F shows one of these data sets for an IgG molecule.[4] Data were collected as a function of temperature and pH, from pH 3 to 8 at one pH unit increments (6 different conditions), and temperatures from 20 to 90°C at 2.5°C intervals (29 different conditions), resulting in a 6×29 assay grid. At each point on this grid, measurements were taken of CD molar ellipticity at 218 nm (Panel A), intrinsic fluorescence peak position and intensity (Panels B and C), tryptophan fluorescence lifetime (Panel D), static light scattering (Panel E), and ANS fluorescence intensity (Panel F).

The data set shown in Figure 2.1A-F presents challenges as well as opportunities. Traditionally, we look for evidence of conformational changes, unfolding, and aggregation, then estimate transition temperatures. This approach suffers three major drawbacks. First, experimental methods sometimes disagree on transition temperatures and protein state. Second, plots like Figure 2.1A do not convey much information to the non-expert. Third, important variations and/or regularities in the data may not carry through to the final analysis when they are unexplained, too complex to easily observe, or partially hidden by noise.

Each experimental technique provides a picture of one or more different aspects of a protein or other macromolecular system. The formulation scientist must assemble this information into an overall picture of the behavior of the protein. The situation is similar to the tale of the "blind men and the elephant", where the macromolecular drug is the elephant, and the experimental methods are the blind men touching different parts of the elephant (tusk, trunk, ear, tail, etc). The formulation scientist is the one who must assemble the information from the others and decide what the elephant looks like. When experimental methods disagree, the formulation scientist must make an educated guess. Sometimes even a single method will report conflicting information, as when transition temperatures between folded and unfolded conformations of a biomolecule differ for measurements from two different wavelengths during the same circular dichroism temperature melt experiment.

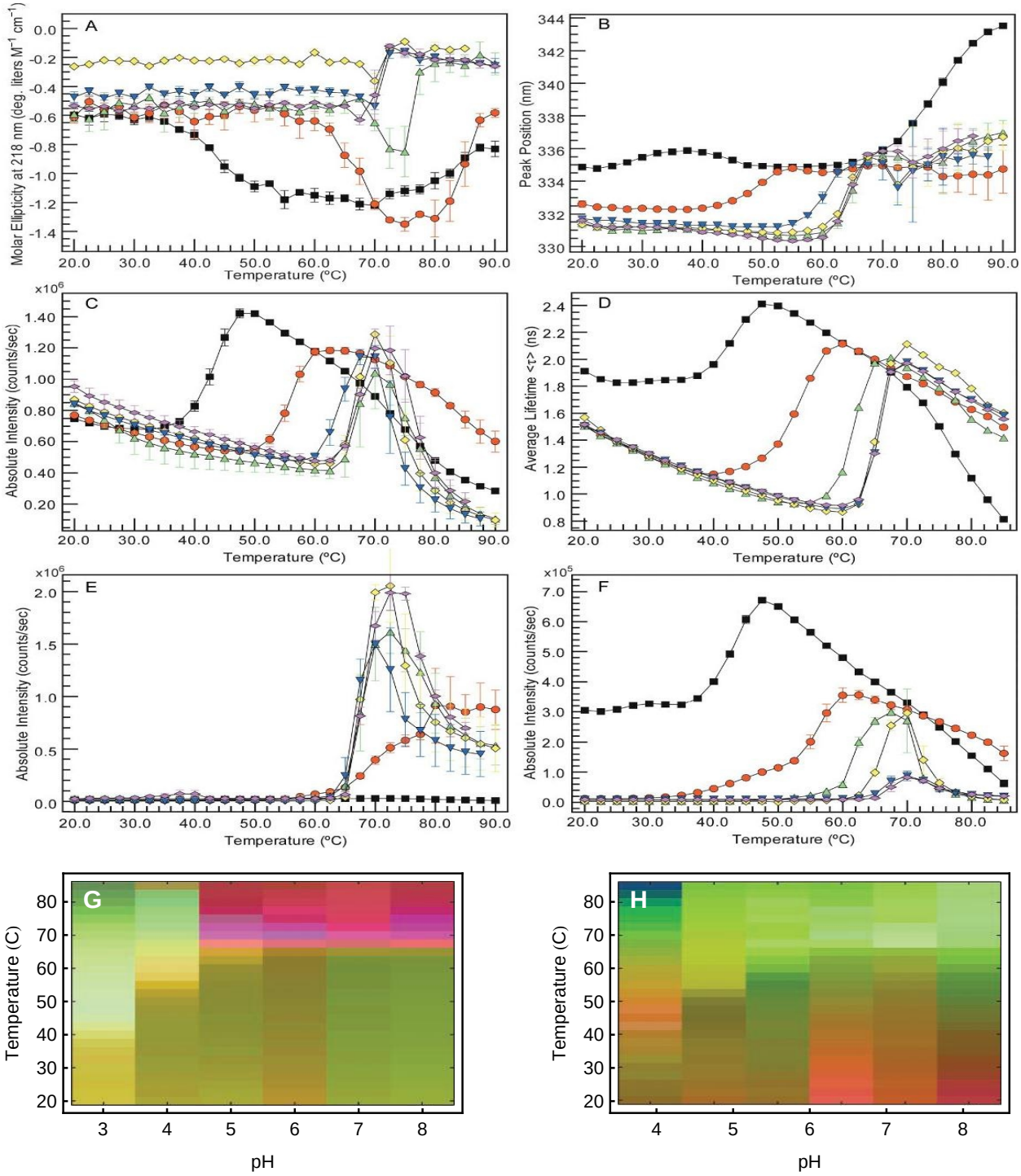


Figure 2.1: An empirical phase diagram (*EPD*) assists in the visualization of data set resulting from the methods listed in Table 1. Figures A-F show measurements of an IgG1 monoclonal antibody collected at pH 3 (black), 4 (red), 5 (green), 6 (yellow), 7 (blue), and 8 (magenta). (A) CD molar ellipticity at 218 nm, (B) UV intrinsic fluorescence (UV-IF) peak position and (C) intensity, (D) tryptophan fluorescence lifetime, (E) static light scattering (SLS), and (F) ANS extrinsic fluorescence (ANS-EF) intensity. Error bars in (A-C and E-F) are from three independent experiments.[4] Figure (G) shows an *EPD* based on the above data. Figure (H) shows an *EPD* based on protein dynamics measurements (data not shown, see the applications section for more information).

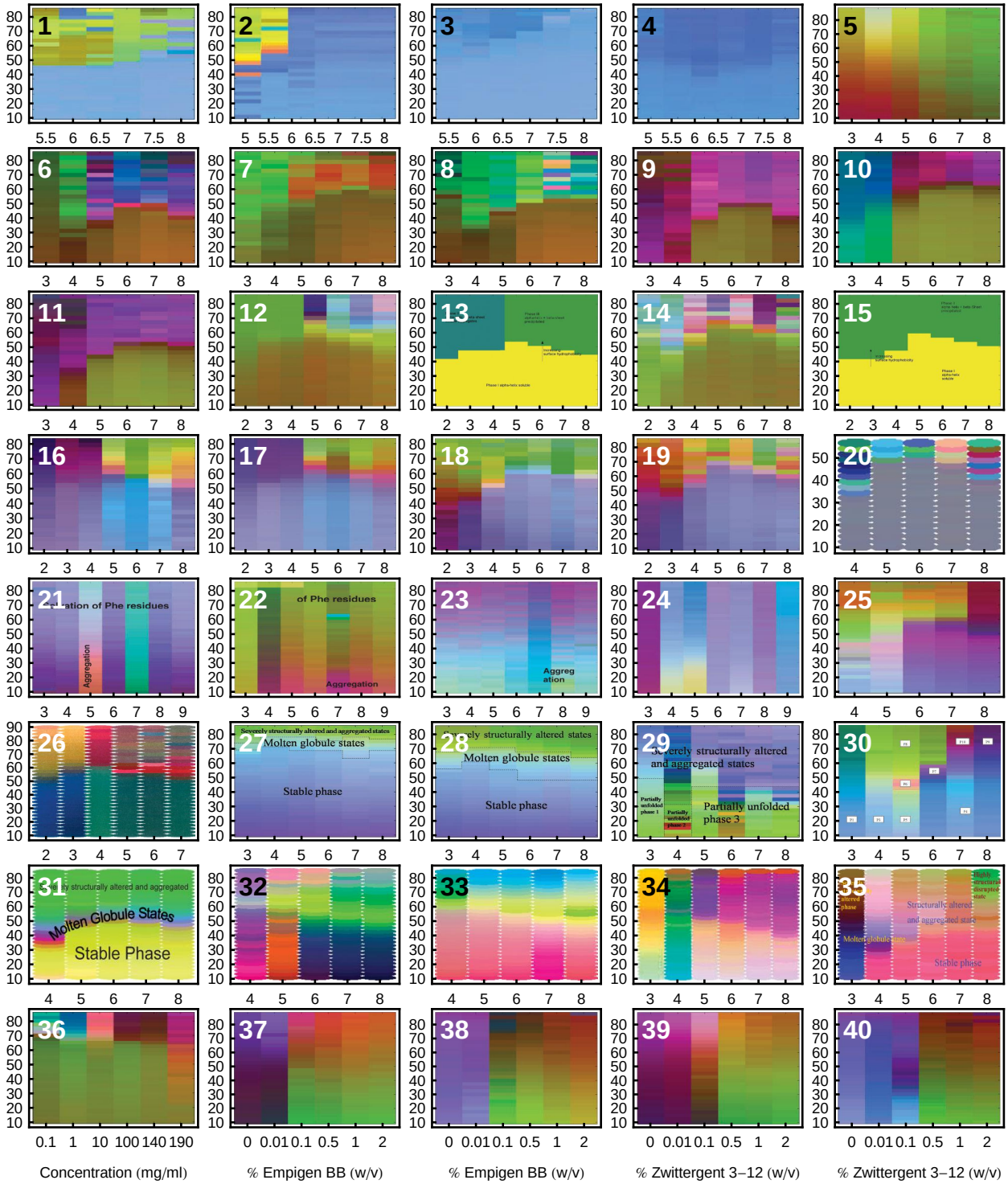


Figure 2.2: Empirical phase diagrams have found many uses in the optimization of various types of formulations. Many case studies have been published concerning their application to various systems and their extension by the addition of measurement techniques and search space variables. Refer to Table 2.2 for more information concerning each *EPD*. All *EPDs* have temperature (°C) as the vertical axis. Diagrams 1-35 use pH on the horizontal axis, and diagrams 36-40 have the indicated variables on the horizontal axis.

Although each experimental method is sensitive to different aspects of protein behavior, different methods often provide overlapping information as well. This manifests itself as regularities in the combined data sets. One would not expect these regularities to always be easily visible in data such as that shown in Figure 2.1A-F. In these plots we show the results from six biophysical methods to monitor the higher order structure of an IgG molecule as a function of pH and temperature. Similar experiments can generate even larger data sets with many more instruments and/or environmental conditions. To find the regularities, we would need to find patterns in a high dimensional space. This is not possible in the simple plots of Figure 2.1A-F. An empirical phase diagram of the data in Figure 2.1A-F is shown in Figure 2.1G (Figure 2.1H will be discussed in the applications section). The red region of Figure 2.1G tells us that high temperature behavior is clearly different between low and high pH (pH values above 4). Inspecting the data, the distinction appears to be subtle and complex, but the *EPD* shows us that in the multidimensional space, the difference is actually pronounced. Furthermore, focusing on measurements at pH 3 (shown in black), we see that the positions of transitions near 40°C are not well defined. On the phase diagram, the transition is sharper and positioned near 40°C.

Formulation scientists must often resort to educated guesses when further information is hidden in the complex data sets generated from a series of measurements. *EPDs* use the results of a global analysis, increasing the use of information and reducing the role of guesswork. Such plots present the results in a simple format, so the eye of a non-expert can pick out regularities and transitions with little difficulty.

2.4 Search space, protein phase space, and measurement phase space

To better understand the mathematical aspects of generating empirical phase diagrams, we first review terms and concepts that arise naturally from the quantitative characterization of

large data sets. Each mathematical term can be made as formal as desired, which we avoid here. Instead, our emphasis is on conveying relevant concepts by using precise mathematical terms in a manner as informal and pictorial as possible.

2.4.1 Search space

The search space is defined by the experimental control variables. One may use virtually anything as a control variable, such as concentrations of excipients, temperature, pH, or variables describing protein history. We cannot test every point in the space, so one usually forms a grid of points to test. We will call this the "search grid". The terms "search space" and "search grid" are borrowed from the field of protein crystallization. In Figure 2.3A, a one dimensional grid has been chosen consisting of 4 pH values. If we had varied both solution pH and temperature, we would have needed two variables to define the solvent state, and we would have tested points in a two dimensional grid (as will be discussed later in Figure 2.6A).

2.4.2 Macromolecule phase space

The state of a target biomolecule or macromolecular complex can be described by a list of numbers. For example, we can use a long list of the positions of all the atoms in a protein.[63] If we consider each list as a point in a high dimensional phase space, then changes in protein shape equate to movement of the corresponding point in phase space. In Figure 2.3A we have illustrated a protein phase space with ratios of secondary structure. An exhaustively complete protein phase space would require thousands of variables to completely describe a protein state.

Preferred molecule states correspond to equilibrium points caused by energy minima in phase space. Due to thermal vibrations, the molecular states fluctuate around these energy minima, and can be visualized as a cloud of points around each minimum, usually described by a Boltzmann distribution.[54,63,64]

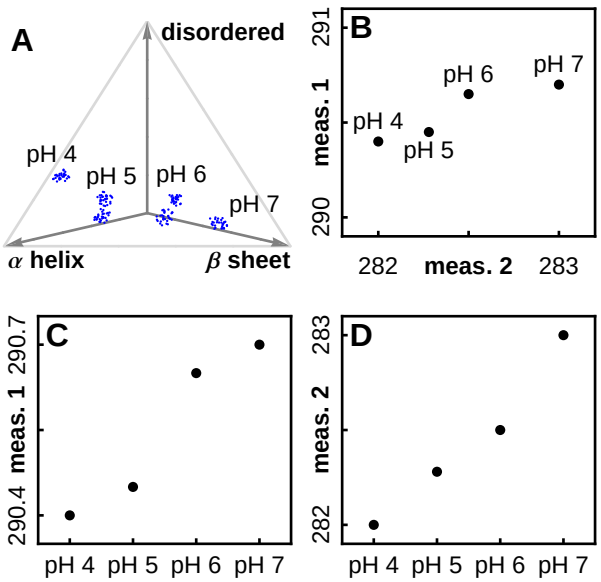


Figure 2.3: An illustration of three types of spaces, using simulated data. In Figure (A), four pH values define a one dimensional search grid in *search space,* and ratios of secondary structure type illustrate a *protein phase space.* Figure (B) shows how two measurement types define a *measurement phase space.* The transition pH values disagree when we plot measurements separately, as in Figures (C) and (D). A plot in measurement phase space (B) synthesizes the information, but will not work as a visual aid for high dimensional data.

For a given solvent condition, there can also be more than one accessible stable protein state, due to the existence of multiple minima in the protein energy landscape.[63] Instead of a single cloud of points for the given solvent condition, there may be several (see Figure 2.3A, pH 5 and 6). When we collect spectroscopic data, we see the average of the contributions from all the protein states.

2.4.3 Measurement phase space

This space is defined by all of the measurements used to probe a macromolecular state. For example, in Figure 2.3B we show how 2 measurements define a 2 dimensional measurement space. If we collect CD data at 3 wavelengths and UVAS data at 2 wavelengths, we can join these into a single 5 dimensional vector (as will be discussed in more detail later; see Figure 2.6B).

The measurements in a data set contain information generated by multiple physical

processes. The types of information derived from these physical processes possess varying levels of prominence in the data. Some stand out on their own, while others require extensive processing to isolate.

We also attribute varying levels of significance to the different types of information. For example, for formulation purposes, information concerning aggregation is highly significant, while information concerning protonation may be less so.

Since the data are generated by physical processes, one cannot expect prominence to be related to significance. Thus, types of data usually require a certain amount of preprocessing.

2.5 Data preprocessing and standardization

2.5.1 Data preprocessing

Data preprocessing steps are designed to extract significant information from data in which it may be hidden in complex ways amid less important information. Preprocessing usually consists of finding the position, width, or intensity of spectral or calorimetric peaks, using methods such as second derivative processing, Fourier self deconvolution, or determination of the spectral center of mass. Another preprocessing practice is the hand-selection of data that is deemed to be pertinent, information rich, and sufficiently free of noise.

For example, the simulated data in Figure 2.3B is similar to preprocessed data from near UVAS second derivative peak position analysis. A spectrum would have been collected and preprocessed for each pH value, yielding the positions of several peaks. Selection of two of the peaks would have resulted in a data set like the one plotted in Figure 2.3B.

Typically, on the order of ten measurements remain after preprocessing. It is best to not overdo preprocessing, which may erase information about transitions. Preprocessing constitutes a bias concerning the significance of types of information, so it must be applied judiciously. An example of extreme preprocessing would be to take an FTIR absorbance spectrum measured at 3000 frequencies and reduce it to a single frequency. The global

analysis we will describe is capable of finding optimal low-dimensional representations of high dimensional data, and tends to perform better when a large number of measurements are used.

2.5.2 Data standardization

Preprocessing results in a collection of numbers that cannot be expected to have appropriate units, scales, or dimensions. The units of most data are standardized by scientific and engineering conventions that have no relation to their significance for formulation development. For example, fluorescence emission photon peak counts of proteins tend to range from 10^4 to 10^6, but absorbance values tend to be kept below 1 AU. The scale of data must be adjusted so that artificial unit conventions do not cause one type of data to overwhelm another. Furthermore, mathematics alone does not contain knowledge of formulation, so it cannot in principle determine the scale choices, preprocessing, and standardization that will lead to useful summaries. Perhaps surprisingly, once these choices are made by the user, mathematics can determine optimal low dimensional representations of data. Fortunately, the adjustment of scale variables is straightforward as described below and rather robust outcomes are not difficult to obtain.

We now discuss an example of the influence of scales on estimates of transition values. Figures 2.3C and 2.3D illustrate how measurements can disagree on the position of transitions. Plotting the measurements separately, measurement 1 (Figure 2.3C) shows a transition between pH 5 and 6, but measurement 2 (Figure 2.3D) shows a transition between pH 6 and 7. The two dimensional plot (2.3B) shows the largest transition between pH 6 and 7. Measurement 2 dominates the two dimensional plot since that peak's variation stretches over a larger range.

Our current approach to resolving the conflict is to resize the variation in each measurement so that they have equal magnitude. There are many ways to do this, and we show only one of them. Since we are only interested in transitions, we begin by centering the mea-

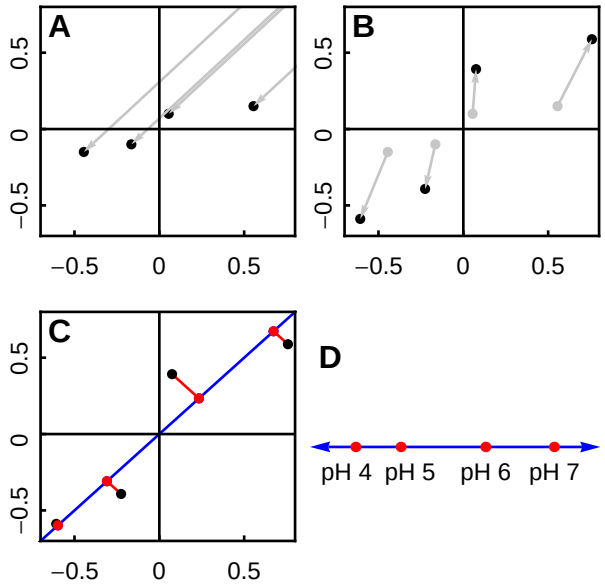


Figure 2.4: Principal Components Analysis can be used to project two dimensions into one. The procedure works the same way for high dimensional data (see Figure 2.5 and 2.6). We will use the simulated data shown in Figure 2.3B. (A): First we center the measurements at the origin, since we are interested in transitions, not average values. (B): Next we normalize each measurement so that they have equal influence on the result. (C): Finally, we use the Singular Value Decomposition (SVD) to find the optimal line for projection (shown in blue). (D): If we plot the position along the blue line, we see that the difference is greatest between pH values 5 and 6.

surements at the origin by subtracting the mean measurement from all the measurements (Figure 2.4A). Then we can normalize each measurement to equalize their variation (Figure 2.4B). We see in this figure that the largest transition occurs between pH 5 and 6.

Hypothetically speaking, if humans could only perceive one dimension, we would want to represent the data in one dimension while preserving the information content as much as possible. Figure 2.4C conceptually illustrates the process. We begin with points in a 2 dimensional space (the black dots), and seek to project the data onto an optimal 1 dimensional space. The term "optimal" is defined by minimizing the projection error, indicated by the red lines. Once the optimal 1 dimensional space has been found, data can be plotted within that space, giving a 1 dimensional plot. This is shown in Figure 2.4D.

2.6 The Singular Value Decomposition (SVD)

Many natural phenomena are poorly understood and currently impossible to model completely. This is the case, for instance, for the behavior of proteins in solution upon perturbation by changes in solution variables such as pH and temperature. When confronted with poorly understood phenomena, it's useful to have a method to help get analysis started. Perhaps the most natural thing to do in such a situation is to consider the shape of the data. In fact, modeling can be described as the attempt to find the least complex summary of the shape of a data set.

The shape of a complex high dimensional data set is usually not perceivable by human visualization capabilities. Attempting to fit different models to such a data set is therefore the only way we have to investigate its shape. In the beginning stages of analysis, it can be useful to use models that corresponding to intuitive notions of shape. In this section we discuss one such model.

2.6.1 Notation

A typical phase diagram utilizes a search grid in temperature and pH, covering pH values from 3 to 8 and temperatures from 10° to 85°C. Collecting a measurement at each search grid point results in data $D_{i_1 i_2}$, where

$$i_1 = 1 \rightarrow \text{pH } 3, \;\; i_1 = 2 \rightarrow \text{pH } 4, \;\; ..., \;\; i_1 = 6 \rightarrow \text{pH } 8$$
$$i_2 = 1 \rightarrow 10°\text{C}, \;\; i_2 = 2 \rightarrow 12.5°\text{C}, \;\; ..., \;\; i_2 = 31 \rightarrow 85°\text{C}$$

(The search grid points need not be evenly spaced.) Collecting m different types of measurements at each search grid point gives data $D^j_{i_1 i_2}$, where $j = 1...m$. Generalizing from a search grid with 2 variables to one with n variables, we have data

$$D^j_{i_1 i_2 ... i_n}, \tag{2.1}$$

where $i_k = 1...s_k$.

In typical applications of the EPD technique, m has typically been around 10, including measurements such as intrinsic and extrinsic fluorescence peak intensity and position, percentages of secondary structure types as determined by circular dichroism, and positions of ultraviolet absorbance peaks as determined by second derivative analysis. When entire spectra are used, m can be several thousand. Since the data set is high dimensional, the human eye cannot easily find its patterns. Humans prefer two-dimensional diagrams. Thus, a method is required to find optimal low dimensional representations of data for visual depiction.

2.6.2 Optimal low dimensional representation

We begin by making the tensor D into a list of vectors:

$$D^j_{i_1 i_2 ... i_N} \rightarrow D_{ij}, \tag{2.2}$$

where the indexes $i_1...i_N$ have been combined, or flattened, into one index i of dimension $l = \sum_{k=1}^{n} s_k$. As before, $j = 1...m$ indexes measurement types.

The matrix D_{ij} is a list of data points j taken at control variables i. We wish to find a projector

$$P = A^T A \tag{2.3}$$

which projects the row vectors D_i into a subspace of a desired number of dimensions , while minimizing some yet to be defined matrix norm:

$$\left\| DP - D \right\| \approx 0 \tag{2.4}$$

The singular value decomposition is a way of easily finding such a projector. The singular

value decomposition of D is

$$D_{ij} = \sum_{\alpha=1}^{d} U_{i\alpha} W_\alpha V_{\alpha j}, \tag{2.5}$$

where $d = Min(l, m)$. The decomposition exists for any matrix, whether real or complex, square or rectangular. The matrices $U_{i\alpha}$ and $V_{\alpha j}$ are calculated by solving for the eigenvectors of the covariance matrices DD^T and $D^T D$:

$$DD^T \cdot U_\alpha = W_\alpha^2 U_\alpha, \tag{2.6}$$

$$V_\alpha \cdot D^T D = W_\alpha^2 V_\alpha, \tag{2.7}$$

where U_α is a column of the matrix $U_{i\alpha}$ and V_α is a row of the matrix $V_{\alpha j}$. For complex matrices replace the transpose D^T by the adjoint $D^\dagger$.

Both DD^T and $D^T D$ are real symmetric, or complex self-adjoint, and positive definite. The numbers W_α, called singular values, are by convention real and positive by a choice of sign (or complex phase) of the eigenvectors. Also by convention, the singular values are sorted in order of decreasing size, and the eigenvectors are sorted accordingly.

The rows of $V_{\alpha j}$ are normalized. Since they are eigenvectors, they are orthogonal to one another:

$$VV^T = \mathbf{I}_{d\times d}. \tag{2.8}$$

Likewise for the columns of $U_{i\alpha}$:

$$U^T U = \mathbf{I}_{d\times d}. \tag{2.9}$$

The rows of V and the columns of U are called singular vectors. When D is real, U and V are also real.

The decomposition is unique up to a complex factor chosen for each pair of eigenvectors:

$$D_{ij} = \sum_{\alpha=1}^{d} (U_{i\alpha} e^{-i\theta_\alpha}) W_\alpha (e^{i\theta_\alpha} V_{\alpha j}), \tag{2.10}$$

Even when D is real, the signs of the singular vectors are not uniquely determined by the decomposition.

2.6.3 The relationship between singular values and data reconstruction error

The norm of each summand in Equation 2.5 is W_α:

$$\|U_{i\alpha} W_\alpha V_{\alpha j}\| = \sqrt{\sum_{i=1}^{l} \sum_{j=1}^{m} |U_{i\alpha} W_\alpha V_{\alpha j}|^2} \tag{2.11}$$

$$= \sqrt{|W_\alpha|^2 \sum_{i=1}^{l} \sum_{j=1}^{m} |U_{i\alpha}|^2 |V_{\alpha j}|^2} \tag{2.12}$$

$$= W_\alpha. \tag{2.13}$$

Since the numbers W_α are sorted in decreasing order, Equation 2.5 is a series of corrections decreasing in size. For many data sets, the most common result is that the data can be approximated well by a sum over just the top few summands α, since the largest singular values tend to be much larger than the rest. If we use the top t singular values, where $1 \leq t < d$, Equation 2.5 becomes

$$\widetilde{D}_{ij} = \sum_{\alpha=1}^{t} U_{i\alpha} W_\alpha V_{\alpha j}, \tag{2.14}$$

where $\widetilde{D}$ is the approximated data.

In the vector space of $l \times m$ matrices, the summands are orthogonal:

$$\sum_{i=1}^{l}\sum_{j=1}^{m}(U_{i\alpha}V_{\alpha j})(U_{i\beta}V_{\beta j}) = \sum_{i=1}^{l}\sum_{j=1}^{m}(U_{i\alpha}U_{i\beta})(V_{\alpha j}V_{\beta j}) \tag{2.15}$$

$$= \delta_{\alpha\beta}\delta_{\alpha\beta} \tag{2.16}$$

$$= \delta_{\alpha\beta}. \tag{2.17}$$

Since the summands in Equation 2.5 are orthogonal and their individual norms are W_α, the norm of the partial sum $\widetilde{D}$ is the same as the ordinary vector norm of the W_α included in the sum:

$$\left\|\widetilde{D}\right\| = \sqrt{\sum_{\alpha=1}^{t} W_\alpha^2}, \tag{2.18}$$

The RMS reconstruction error, directly expressed as

$$\left\|D - \widetilde{D}\right\| = \left\|D - \sum_{\alpha=1}^{t} U_{i\alpha}W_\alpha V_{\alpha j}\right\|, \tag{2.19}$$

can also be expressed as

$$\left\|D - \widetilde{D}\right\| = \sqrt{\sum_{\alpha=t+1}^{d} W_\alpha^2}. \tag{2.20}$$

2.6.4 Example

We now illustrate SVD with a simple example (Figure 2.5). Some familiarity with linear algebra will assist the reader, but the following discussion should also be accessible to a general audience. Suppose we are given the following measurements of a protein at different

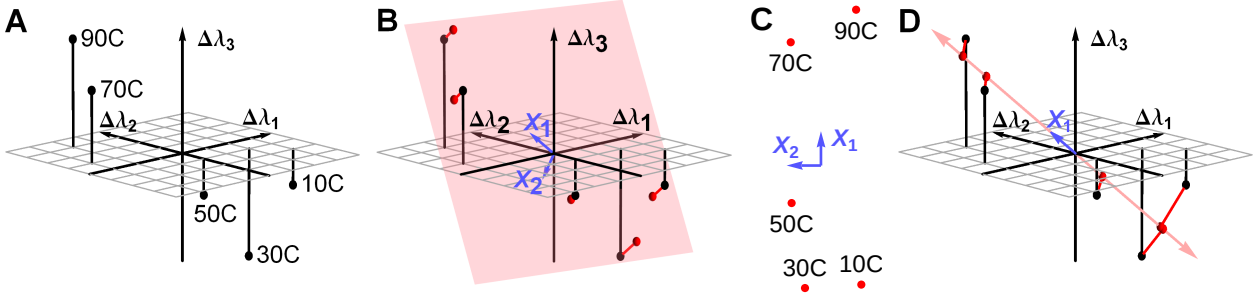


Figure 2.5: An example of the Singular Value Decomposition (SVD) using simulated data. (A) is a plot of three simulated peak shifts $\Delta\lambda_1$, $\Delta\lambda_2$, and $\Delta\lambda_3$ as a function of temperature. If we could perceive two dimensions but not three, the transition between 50°C and 70°C might be difficult to see. Therefore, we would want to reduce the data to two dimensions in a way that optimally retains the information in the original data set. (B) shows the plane (in pink) which gives the optimum 2D projection. This plane is determined by SVD, and is defined by the vectors X_1 and X_2 (in blue). The projection error is shown as red lines. (C) is a 2 dimensional plot of the same data, using the positions within the pink plane. This is a plot of matrix A (see text). (D) shows the optimal one dimensional projection, demonstrating that the error is larger. This plot uses the first column of matrix A (see text) .

temperatures.

	$\Delta\lambda_1$	$\Delta\lambda_2$	$\Delta\lambda_3$
10°C	3	-2	-1
30°C	2	-1	-3
50°C	0	-1	-1
70°C	-3	1	2
90°C	-2	3	3

(2.21)

For instance, the data might represent second derivative ultraviolet absorbance peak shifts in hundredths of a nanometer.

A plot of this data is shown in Figure 2.5A. Each row is plotted as a point in three dimensions, and each point corresponds to a different temperature. The data is three dimensional in the sense that at each temperature we have three numbers, $\Delta\lambda_1, \Delta\lambda_2$, and $\Delta\lambda_3$, which represent the state of the target molecule.

If we could perceive two dimensions but not three, the transition between 50°C and 70°C might be difficult to see. So we would want to reduce the data to two dimensions in a way

that optimally retains the information in the original data set. To see how to do this, refer to Figure 2.5B. The black points are the data points, and the pink area represents a plane. The red points are the positions within the plane that are nearest to the data points. They are two dimensional approximations to the data points. The red lines represent the error in the approximation. We seek the plane which minimizes the total error, defined as the sum of the squares of the lengths of all of the red lines.

To show how this works, we first express the data as a matrix:

$$D = \begin{pmatrix} 3 & -2 & -1 \\ 2 & -1 & -3 \\ 0 & -1 & -1 \\ -3 & 1 & 2 \\ -2 & 3 & 3 \end{pmatrix}. \tag{2.22}$$

SVD finds an optimal, unique two dimensional approximation, which we will call $\widetilde{D}$.

$$\widetilde{D} = AX \tag{2.23}$$

$$= \begin{pmatrix} -3.5 & -1.2 \\ -3.6 & 0.46 \\ -1.1 & 0.85 \\ 3.6 & 0.88 \\ 4.5 & -1.0 \end{pmatrix} \begin{pmatrix} -0.63 & 0.48 & 0.61 \\ -0.77 & -0.28 & -0.57 \end{pmatrix} \tag{2.24}$$

The matrix A consists of the left singular vector matrix multiplied by the singular value matrix, and retains only the top 2 singular vectors. We have done this to simplify the presentation.

The two rows of the matrix X are perpendicular to each other. In Figure 2.5B, the two rows of X are represented as blue vectors. They are axes in a two dimensional plane (shown

in pink), and serve to define that plane. This plane is the unique plane that minimizes the total error. When we perform the matrix multiplication AX, each row of A specifies a linear combination of the rows of X. The matrix multiplication places the approximated data points within the plane defined by the row vectors of X.

In this example, SVD actually returned three optimal axii and we have chosen and shown only the two most important ones (the two rows of X, shown as blue vectors in Figure 2.5B). When we choose optimal axii to define the lower dimensional space, we generally discard the other axii returned by linear algebra. If we had used only the first row of X, we would have approximated the data within a one dimensional space (Figure 2.5D). In that case, the error would have been larger. (On the other hand, the optimal one dimensional axis may well encompass most of the data, depending on the relative magnitude of the singular values.)

When we use all of the axii given by SVD, there is no error, and the approximation $\widetilde{D}$ is equal to the original matrix D. Error results from excluding axii, as we have done in Figures 2.5D and 2.5B. If we exclude axii that only result in a small increase in error, the approximation $\widetilde{D}$ can be very close to the original matrix D.

For many data sets, the most common result is that only a few of the axii are important, resulting in a large increase in error when they are dropped. The rest of the axii can usually be eliminated with very little effect on the approximation. We can choose in advance the number of axii to use. In this example, we have three dimensional data that we want to reduce to two dimensions. We can minimize the error for a two dimensional projection by using the two most important axii returned by SVD.

Since we want a true two dimensional representation of D, it is self-consistent to use the positions within the optimal plane instead of the three dimensional positions. The two dimensional positions within the plane are given by the matrix A, and are plotted in Figure 2.5C. Each point in Figure 2.5C represents a row of A. The error in the approximation from D to $\widetilde{D}$ is the sum of the squares of the lengths of all the error vectors (the red lines in Figure 2.5B). This is the error that SVD minimizes.

The entire procedure we use to project data is known as Principal Components Analysis (PCA). PCA consists of subtracting the mean from a data set and applying SVD. These steps are shown in Figure 2.4A and 2.4C. The extra step of normalizing the measurements, shown in 2.4B, is a known extension to PCA.

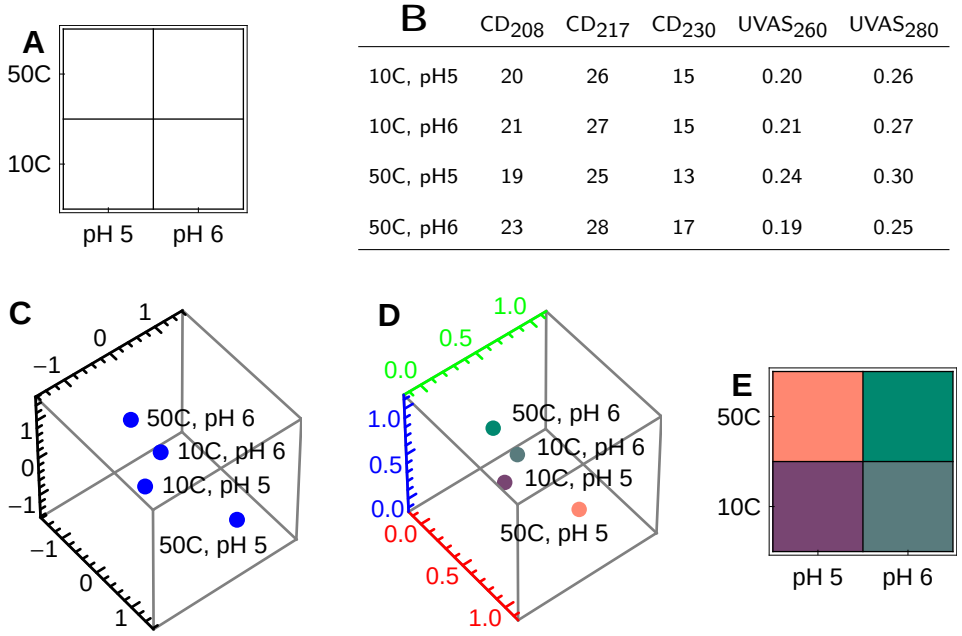
It is important to note that the procedure gains its power from the fact that it works the same way in higher dimensions. Instead of three peak shifts, as in the previous example, we might be given 5 measurements at each temperature. These are vectors in a five dimensional space. After standardization, we apply SVD to the matrix of data, returning up to 5 axii. The most significant axii are then used to define a lower dimensional space. The projection onto that space is the best possible approximation to the data that can be made based on the number of dimensions retained. Just as in the example above, the approximated matrix still appears high dimensional. Yet we can get a true low dimensional view of the data by using the data point positions within the space defined by the retained axii (as illustrated for two dimensional projections in Figure 2.5C).

2.6.5 History

The singular value decomposition is attributed to the mathematicians Beltrami and Jordan, who discovered a version in the 1870's. The physicist Carl Eckhart is credited with extending the procedure to non-square matrices. It seems to have been re-discovered many times, and is sometimes associated with Householder and Karhunen-Loeve.[65]

2.7 The Empirical Phase Diagram

We begin by choosing a search grid (Figure 2.6A). The most common search grid previously used for protein phase diagrams covers pH values from 3 to 8 in one pH unit increments and temperatures from 10°C to 85°C in 2.5°C increments. Measurements typically include a series of biophysical techniques such as CD, fluorescence, and UV absorbance spectroscopy



B	CD_{208}	CD_{217}	CD_{230}	$UVAS_{260}$	$UVAS_{280}$
10C, pH5	20	26	15	0.20	0.26
10C, pH6	21	27	15	0.21	0.27
50C, pH5	19	25	13	0.24	0.30
50C, pH6	23	28	17	0.19	0.25

Figure 2.6: Illustration of the steps in the Empirical Phase Diagram method, using simulated data. (A): Choose a search space and a search grid. In this case, the search space is 2 dimensional, varying temperature and pH in this case. In each dimension, two values have been chosen, forming a grid. (B): Collect data at each point of the search grid. The data in this example is 5 dimensional. (C): Standardize the data and project it into 3 dimensions. (D): Rescale to the range $(0, 1)$, and express as a color. (E): Transform the colors into an image.

as well as light scattering. In this simulated example, we choose a simpler case of two pH values (5 and 6) and two temperature values (10°C and 50°C) as measured by CD (at 3 wavelengths) and UVAS (at two wavelengths).

After collecting and preprocessing the data, a matrix is created in which the rows correspond to all search grid positions and the columns correspond to all measurement types (Figure 2.6B). The matrix is standardized as described in the previous section, then projected down to 3 dimensions using the singular value decomposition. The result of standardization and projection is shown in Figure 2.6C. The number of rows remains the same but the number of columns has been reduced to 3.

To provide a convenient visual image, the resulting 3 dimensional positions are converted into ratios of red, green and blue color. First the data is shifted and resized so that all the numbers fit in the range (0,1) (Figure 2.6D). Then the 3 dimensional positions are expressed

as colors. To create the phase diagram, the colors are reorganized into a grid and plotted (Figure 2.6E).

2.7.1 History

A technique similar to EPDs was used in 1989 to merge satellite imagery.[66] It is called "PCA based image fusion", is widely employed in geo-sensing and in-vivo imaging,[67,68] and is spreading to other areas such as art conservation and astronomy.[69,70] Unaware of this history, we first applied PCA in 2003 to characterize transitions in higher order protein structure under different environmental stresses.[6]

2.7.2 Interpretation of Empirical Phase Diagrams

Once the empirical phase diagram has been generated, the mathematical work is done. What remains is interpretation. The first step is to inspect the phase diagram to determine regions of conserved structure. Areas of search space that produce similar measurements in the abstract 3-dimensional space manifest themselves on the phase diagram as areas of a single contiguous color. Transitions are then manifested as changes in color, with noise showing as irregular and often quite subtle color variation.

The color of an area is itself a "code", not universally meaningful information. To get an idea of why this is, refer to Figures 2.5B and 2.5C. PCA gave the two vectors X_1 and X_2, defining a plane for the optimal two dimensional projection. An entirely different data set projected into 2 dimensions will also give an optimal plane whose absolute orientation relative to the first cannot be known without comparing the sets with each other. Thus, two different meanings can (and generally will) be applied to a given color code. This is not a matter of much concern, because the color code is not actually used in a quantitative analysis. The colors serve no purpose other than to identify areas of different behavior. One might just as well have labeled contiguous regions with names or numbers, as in traditional thermodynamic phase diagrams.

While the results of PCA are unique for any given data set, small changes in a data set can sometimes result in rotation of the principal axii. That will occur when two large, important singular values are nearly equal. Then distinguishing them by size-ordering can hinge on small variations. The result of swapping the order of axes is a swap of two colors. The shapes of the regions and transitions will, however, remain the same, because the projection plane in question is an absolute concept that does not depend on the labeling. Notice in Figure 2.5C that a deliberate rotation of X_1 and X_2 does not alter any information about the transition.

In a study of *Clostridium difficile* toxins and toxoids, discussed below, phase diagrams were generated jointly to achieve uniform meaning of their colors.[16] In such an analysis, the target macromolecule becomes one of the control variables. For example, if the control variables had been pH and temperature, they will now also include the target macromolecule. The matrix shown in Figure 2.6B will contain additional rows to incorporate the increased number of combinations of control variable positions. The matrix is then standardized, projected into 3 dimensions, rescaled, and interpreted as colors, as shown in Figures 2.6C and 2.6D. Finally, the colors are made into multiple phase diagrams, one for each target macromolecule.

After the *EPDs* are inspected to determine regions of conserved structure, one tries to determine as much as possible about the actual physical state of the protein or macromolecular complex within those regions. To do this, one must refer to the original measurements and consider the physical processes that generated them. To reiterate, the best one can hope from quantitative analysis is optimal projection, which still needs expert scientific evaluation of the original biophysical data, and perhaps further targeted experimentation. By referring back to the source data, empirical phase diagrams can usually be segmented into the following types of structure: *low temperature inactive, active form, molten globule states, high temperature or acidic pH unfolded forms,* and forms which are *aggregated or dissociated* to various extents. Sometimes, however, a region of an *EPD* may have no ready interpretation,

indicating that the data and mathematics have found something the expert does not readily recognize.

2.8 Applications and case studies

Here we summarize some applications and case studies using empirical phase diagrams to formulate and stabilize various biomolecules and larger macromolecular complexes. As highlighted earlier, common pharmaceutical applications have been to select stress conditions

Table 2.2: Biomolecules and larger macromolecular complexes, analytical techniques and environmental stress conditions evaluated by empirical phase diagrams. See Table 2.1 for definitions of the technique abbreviations.

Target	Techniques	Search Space	Figure	Ref.
Measles virus	CD, DLS, SLS, EF	pH, T[a]	2.10	24
Human respiratory syncytial virus	CD, UVAS, OD_{350}, UV-IF	pH, T		25
Live att. Ty21a bacterial typhoid vaccine	CD, EF	pH, T		29
Adenovirus type 5 (Ad5)	UVAS, DLS, UV-IF, EF	pH, T	2.2.32, 2.2.33	26
Recombinant ricin toxin A-Chain vaccine	CD, UF-IF, EF	pH, T	2.2.20, 2.2.31	71
Adenovirus type 2 (Ad2)	CD, UVAS, OD_{350}, DLS...	pH, T	2.2.34	27
Hep. C virus envelope glycoprotein E1	CD, DLS, UV-IF, EF	pH, T, S[a]	2.2.37 - 2.2.40	21
Clostridium difficile toxins and toxoids	CD, OD_{350}, UV-IF, EF	pH, T	2.8	16
Type III secretion system tip proteins	CD, UVAS, UV-IF, EF	pH, T		14
Type III secretion system needle proteins	CD, UVAS, EF	pH, T		17
Malaria antigen EBA-175 RII-NG	CD, UV-IF, EF	pH, T		15
H1N1 influenza virus-like particles	CD, DLS, EF	pH, T	2.2.25	22
Norwalk virus-like particles	CD, UVAS, UV-IF, EF	pH, T	2.9	23
Nonviral gene delivery complexes	CD, DLS, EF	pH, I[a]	2.11	28
Human Inteferon-β-1a	CD, UVAS, UV-IF, EF	pH, T	2.2.12, 2.2.19	5
Bovine granulocyte colony stim. factor	UVAS	pH, T	2.2.26	6
Immunoglobulin-G (IgG)	CD, EF, PPC, HRUS, TRFS	pH, T	2.1	4
Pramlintide (antihyperglycemic peptide)	CD, UVAS, OD_{350}, UV-IF	pH, T, C[a]	2.7	7
Monoclonal antibodies	CD, UVAS, OD_{350}, UV-IF	T, C	2.2.36	8
Clostridium botulinum A neurotoxin	CD, UV-IF, EF	pH, T		9
Molecular chaperones Hsc70 and gp96	CD, UVAS	pH, T		10
Human fibroblast growth factor 1	CD, UVAS, UV-IF, EF	pH, T, S	2.2.6 - 2.2.11	11
Fibroblast growth factor 20 (FGF-20)	CD, UVAS, UV-IF	pH, T		12
rPA of *B. anthracis*	CD, UV-IF, EF	pH, T	2.2.31, 2.2.35	13
Recombinant vault particles	CD, UV-IF, EF	pH, T	2.2.30	18
Recombinant human gelatins	CD, UV-IF, UVAS	pH, T	2.2.21 - 2.2.24	19
EC5 domain of E-Cadherin	CD, UV-IF, UVAS	pH, T, N/R [a]	2.2.27 - 2.2.29	20

[a] T = Temperature, I = Ionic Strength, C = Concentration, S = Stabilizer, N/R = Native/Reduced

for high throughput excipient screening, to find ranges of solution conditions resulting in optimized stability, and to investigate the overall structural integrity and conformational stability behavior of large macromolecular complexes.

2.8.1 Selection of stress conditions for excipient screening

Screening compounds and polymers for stabilization of a liquid formulation of a biomolecule or macromolecular complex is a time consuming process due to both the large number of excipients that should be tested, and the time it takes to complete each test. The latter can be reduced by selecting conditions which accelerate degradation processes. (Although the danger always exists that the degradation reactions induced may not be directly relevant to actual storage conditions.) The *EPD* approach can be used to select these accelerated conditions. Since each region of color in an *EPD* represents a different state of the system, it is presumably related to a local minimum in the energy landscape. Thus, at transitions between these regions, the system may have a somewhat higher energy and be farther from equilibrium. This makes it more likely (but not guaranteed) that the system can access other minima in the energy landscape under these conditions. By selecting transition conditions within pharmaceutically accessible regions, it seems probable that relevant degradation mechanisms during real time storage will be enhanced under these accelerated conditions. This basic concept has been applied to many formulation projects with significant success as described below, and is a commonly used general assumption in pharmaceutical preformulation and formulation efforts.

2.8.2 Finding stabilizing conditions

By the same argument, we can also find stabilizing solution conditions (e.g. pH and ionic strength) for a liquid formulation by selecting conditions distant from *EPD* boundaries. More routinely, *EPDs* have assisted in the more standard stabilization and formulation process, in which one finds solution conditions that increase stability as measured by the elevation of

thermal unfolding/melting temperatures or reduction of aggregation.[24,71]

2.8.3 Using *EPDs* to investigate the similarity of two proteins

In the construction of *EPDs*, we perturb the system by varying solution conditions such as temperature, pH, and ionic strength, while measuring the system's response. Rather than focusing on transitions, we can also use an *EPD* in its entirety to gain additional information about the identity of the system's native form. We have found that *EPDs* of proteins of similar function do indeed appear similar.[10–12,16,17] For example, the two heat shock proteins Hsc70 and gp96 have very little sequence homology, but demonstrate apparent phase changes in their *EPDs* which are nearly identical.[10]

2.8.4 Investigating protein dynamics

The intramolecular mobility of large molecular systems is a critical factor in their behavior, and a role in molecular recognition and enzymatic catalysis is now generally recognized.[72] The relationship of the dynamic behavior of such systems to their stability remains, however, poorly understood.[53] In this regard, *EPDs* have been employed to characterize the intramolecular dynamics of an IgG1 monoclonal antibody on a temperature-pH perturbation grid.[4] This study employed measurements sensitive to protein dynamic motions such as molecular tumbling, domain movement, and the degree of solvation. A combination of the following measurements was used: adiabatic compressibility determined from PPC, coefficient of thermal expansion determined from HRUS, REES, and rotational correlation times determined by TRFS anisotropy. (See Table 2.1 for instrument abbreviations.) An *EPD* was also generated based on the following time averaged methods: steady-state UV-IF, far-UV CD, light scattering, and ANS-EF. The latter methods are sensitive to alterations in protein secondary and tertiary structure. The *EPDs* from the dynamic and static measurements are shown in Figures 2.1G and 2.1H, respectively. In both *EPDs*, a very different conformational state was observed at pH values 3 and 4. The *EPD* based on the dynamics measurements

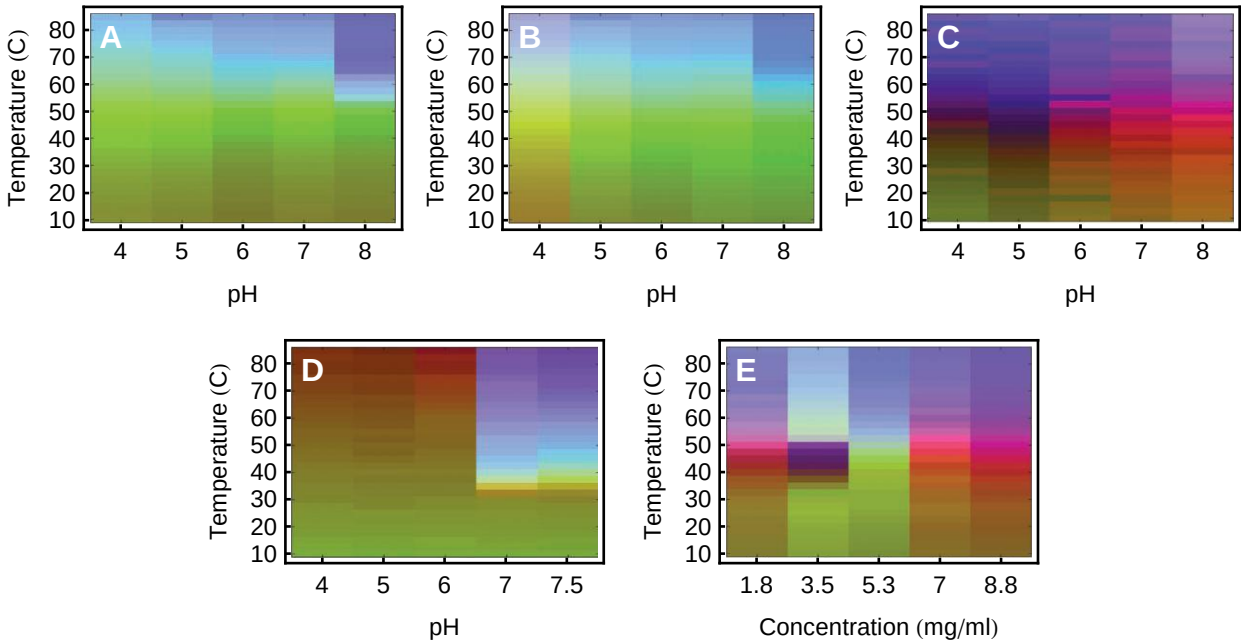


Figure 2.7: Empirical phase diagrams (*EPDs*) of the peptide drug pramlintide at low and high concentration,[7] and concentration dependence at pH 4. Low concentrations (0.088 mg/ml) are represented in A-C. The experimental techniques used to construct A-C were as follows: (A) second derivative UV absorbance peak shift and OD_{350}, (B) same as (A), adding fluorescence intensity and peak shift, (C) same as (B), adding the CD change at 204 nm. The peptide at high concentration (8.8 mg/mL) is represented in (D), using the same experimental techniques as (B). An *EPD* at pH 4 as a function of peptide concentration is shown in (E), using the same experimental techniques as (B).

is more complex overall, with low temperature events seen that are not present in the static *EPD*. This study indicates that measurements of protein dynamics potentially provide a more sensitive probe of protein stability and the effect of potential stabilizers. Related approaches are under further development in our laboratories.

2.8.5 Evaluating a peptide drug (Pramlintide)

The *EPD* method has not yet been used with small molecule pharmaceutical drugs, but it has been employed to characterize peptides. An analogue of amylin, the 37-residue peptide Pramlintide is currently used as an antihyperglycemic agent to treat diabetes. This peptide was characterized using a combination of CD, intrinsic Tyr fluorescence, second derivative UV absorbance, and optical density as a function of pH, temperature, and peptide concentration.[7] Despite the fact that the data shows that the peptide is primarily unstructured at low

concentration (confirmed by isotope exchange NMR), the *EPDs* are still surprisingly complex with distinct pH and temperature dependence reflecting very gradual structural alterations and some limited aggregation (Figure 2.7A-C). When the characterization was conducted over a wide range of Pramlintide concentrations, much more distinctive changes in color were observed with transitions shifted to much lower temperatures and a narrower range of pH (Figure 2.7D-E).

2.8.6 Investigating the behavior of larger macromolecular complexes

The *EPD* approach enables visualization of high dimensional data, assisting in the determination of regularities and transition points. For the *EPD* approach to work, only two conditions are necessary, including that the system under study possess a well-defined structural identity, and that transitions in this identity are manifested in the data. A complete physical understanding of the processes governing the transitions is not necessary.

For example, viruses, virus like particles (VLPs), carbohydrate-conjugates, gene delivery vehicles, and other related macromolecular complexes have defined shapes, sizes, structural features and stability profiles. With selection of appropriate techniques, transitions in structure will be manifested in the data as multidimensional transitions in the measured values. These transitions can reflect significant structural changes that may be associated with changes in biological activity.

Signals obtained from such large systems, however, are the sum of signals from many subsystems and these subsystems are themselves large. Thus, unlike smaller biomolecules such as purified proteins, it is unlikely that one will be able to directly relate the changes seen to actual molecular events in these larger macromolecular complexes. It may well be, however, that the experimental signals observed are due to subsystems that are present in multiple copies, and therefore reflect stress induced changes in key components of the complexes (for example, many copies of a viral coat protein within an intact virus.) Thus, such *EPD* data may still be quite useful in characterization studies. The *EPD* approach has, in fact, been

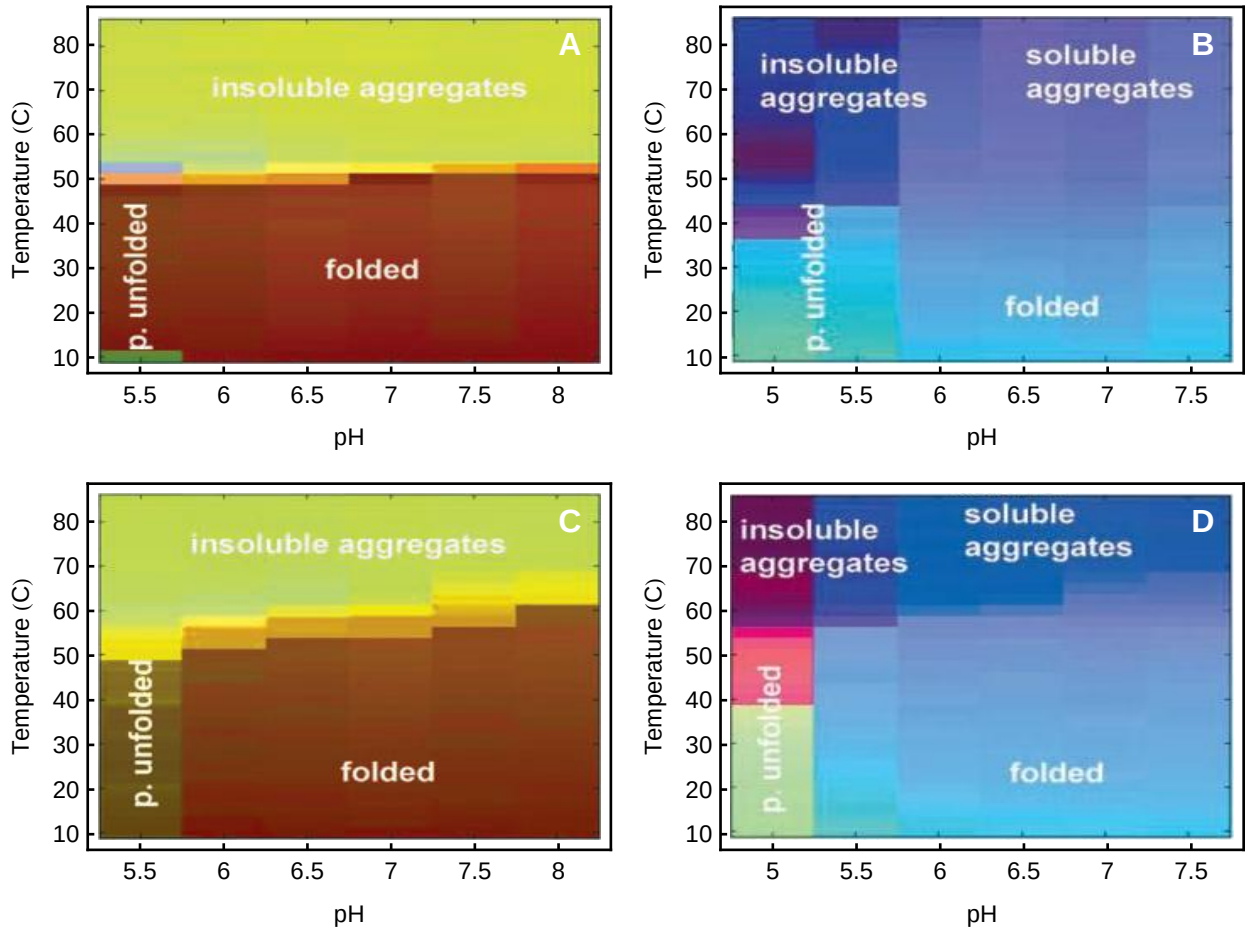


Figure 2.8: An empirical phase diagram for two toxins and toxoids of *Clostridium difficile*, created using OD_{350}, UV-IF, ANS-EF, and CD data.[16] Data were normalized simultaneously for the corresponding toxin and toxoid. (A) Toxin A; (B) Toxin B; (C) Toxoid A; (D) Toxoid B

applied successfully to the development and stabilization of numerous vaccines, including live attenuated bacterial vaccines,[24] inactivated and live viruses and VLPs,[21–23,25–27,29] as well as gene deliver complexes.[28,33]

2.8.7 Formulation of Clostridium difficile toxins and toxoids

To further describe the *EPD* approach, we present a few representative examples of applications to biopharmaceutical drugs and vaccines based on proteins and larger macromolecular complexes. For example, studies using the *EPD* method were conducted of the A and B toxoids of *Clostridium difficile*, which are in clinical trials as a diarrheal vaccine.[16] The pro-

teins were characterized with CD, intrinsic and extrinsic (ANS) fluorescence, optical density, UV absorbance, and DLS. Clearly defined regions corresponding to folded protein, partially unfolded states as well as both soluble and unsoluble aggregates are observed (Figure 2.8A-B).[16] Differences in *EPDs* are seen when the two toxins are cross-linked with formaldehyde to produce toxoids for use as vaccines (Figure 2.8C-D) including enhanced thermal stability. Further utility of *EPDs* is illustrated by their use in pre-formulation characterization studies of the toxoid. Based on the apparent phase boundaries observed in the initial studies, a high throughput screening study was developed based on thermally induced aggregation of the proteins at low pH. A collection of 30 GRAS compounds was then screened and a number were identified which inhibited aggregation. To differentiate effects on conformational stability and aggregation, the proteins were also studied with spectroscopic methods in the presence of presumptive stabilizers. Finally, stabilization studies of the toxoids on the surface of an aluminum salt adjuvant were conducted using DSC. Thus, a series of stabilizers were identified which were successfully employed in final formulations of a candidate *C. difficile* vaccine.

2.8.8 Preformulation screening of norwalk virus-like particles

Multimeric biocomplexes can also be analyzed by use of *EPDs*. The most successful recombinant protein vaccines are, in fact, of the virus-like particle (VLP) type (i.e. Hepatitis B vaccine, HBV, and the human papillomavirus vaccine, HPV). One recent example of a candidate vaccine based on VLP technology is that of the Norwalk virus. This VLP consists of an icosahedral assembly of 180 copies of the VP1 capsid protein of the native virus with only a few copies of the VP2 protein also present. The resultant 38 nm particle was characterized by a combination of CD, DSC, intrinsic and extrinsic fluorescence, near UV absorbance and DLS, as a function of pH and temperature.[23] A series of apparent phases could be identified in the *EPD* corresponding to a variety of conformational and aggregative states (Figure 2.9), including various states of dissociation of the particles. The precise nature of the latter was

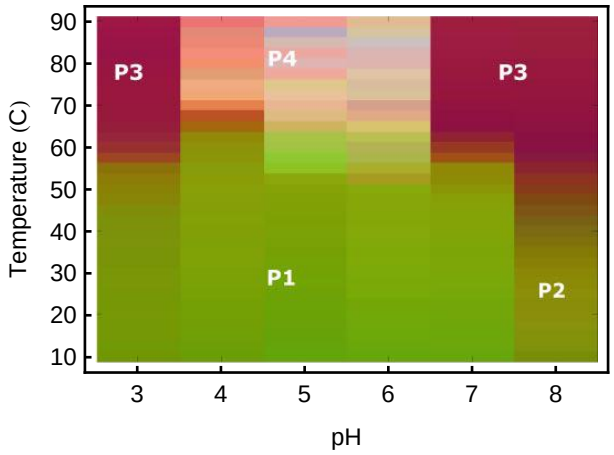


Figure 2.9: Empirical phase diagram for Norwalk virus-like particles (NV-VLPs) based on UV absorbance, intrinsic and extrinsic fluorescence and CD results.[23] Four distinct phases (P) of the NV-VLP were observed: P1, native, intact form; P2, disassembled; P3, soluble VP1 oligomers; P4, aggregated. The nature of the protein in the various phases was confirmed by transmission electron microscopy studies.

established by complementary transmission electron microscopy (EM) experiments. Again, the *EPD* was used as a basis to select conditions to analyze the aggregation state of, in this case, the virus like particles. Compounds which were found to inhibit aggregation were also examined for their effects on ANS-EF, DSC and CD, with sucrose, trehalose, glutamate, and chitosan all found to both inhibit aggregation and conformationally stabilize the Norwalk VLP's.[24] This study led to formulation of a candidate vaccine which has been successful in Phase II trials.[75,76]

2.8.9 Stabilization of measles virus

Larger macromolecular complexes such as killed and live viruses have also been characterized by the *EPD* approach. For example, the relatively unstable attenuated measles virus which is the basis for the important live virus measles vaccine has been examined using *EPDs*.[24] This enveloped attenuated virus contains multiple copies of six different proteins as well as a ssRNA genome. Analysis is further complicated by the fact that the vast majority of viral particles have been inactivated during large scale preparation of the virus. Thus, the potential utility of biophysical studies is based on the assumptions that any change that affects the biological activity (immunogenicity in this case) of immediate interest is still detectable

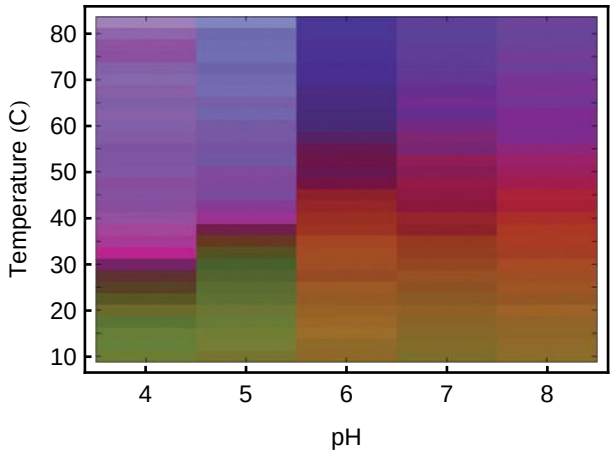


Figure 2.10: Empirical phase diagram of attenuated Measles virus.[24] Data used to generate the *EPD* were measurements of mean effective diameter by DLS, intensity of 562 nm light scattered at 90°, CD at 222 nm, intrinsic fluorescence intensity at 322 nm, ANS peak position, ANS fluorescence intensity at 469 nm and generalized polarization of laurdan fluorescence.

in a significant number of the remaining complexes and that individual measurements detect significant amounts of altered components (presumably due to their presence in multiple copies). While this is no doubt not always true, we have found such assumptions in most cases to be reliable. The measles virus was first purified from its crude vaccine preparation and then examined by the usual combination of spectroscopic and light scattering techniques.[24] One additional *EPD* method not previously described involved the use of the fluorescent dye laurdan, a probe of membrane fluidity. The resulting *EPD* displays at least 6 regions of differing structure (Figure 2.10). An excipient screening method based on aggregation of the virus was used to identify potential stabilizers as determined by melting temperatures with the generalized polarization of laurdan fluorescence used as a confirmatory method. The compounds identified were then examined in cellular infectivity assays and served as a basis for a significant improvement in the thermal stability of the vaccine.

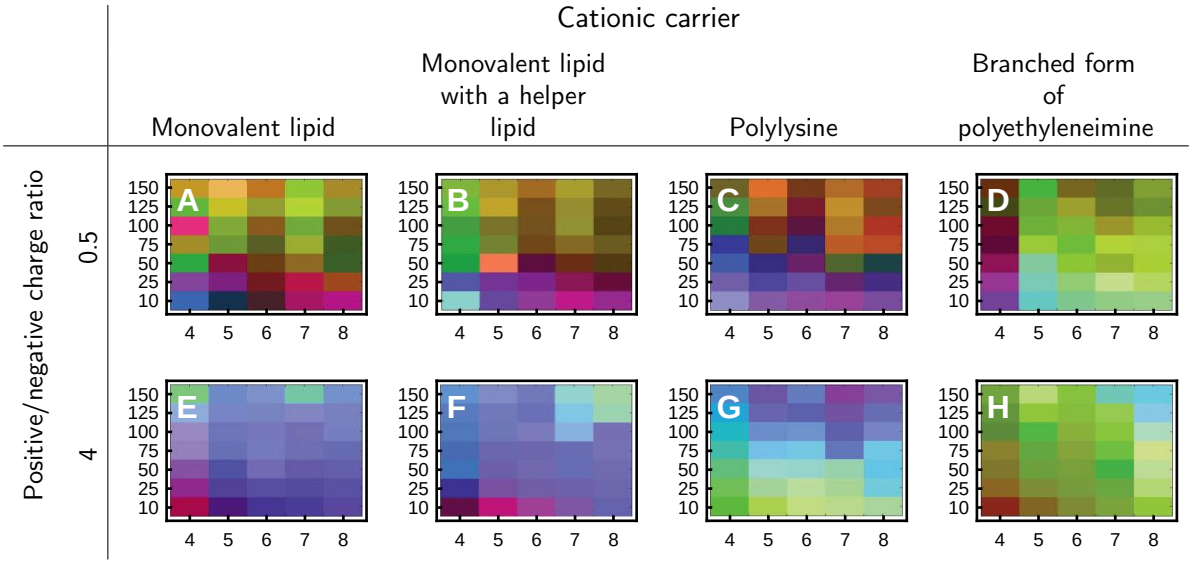


Figure 2.11: Ionic strength-pH empirical phase diagrams of various nonviral gene delivery complexes formed between plasmid DNA and four cationic carriers.[28] Each *EPD*has pH as the horizontal axis and ionic strength (mM) as the vertical axis. The experimental techniques used were DLS, CD, and YOYO-1 EF.

2.8.10 Investigation of polymeric and liposomal gene delivery systems

As a final example, polyplexes and lipoplexes containing plasmid DNA molecules complexed to various polymers and cationic lipids, respectively, were examined by the *EPD* method. Because of the high thermal stability of the DNA component, pH and ionic strength (rather than temperature) were used as the stress variables. Due to the electrostatic nature of the complexes, they were characterized over a wide range of positive and negative nitrogen to phosphate ratios using circular dichroism, extrinsic fluorescence with a DNA intercalating dye (YOYO-1) and dynamic light scattering.[28] The *EPDs* derived for the polyplexes and lipoplexes lacked the sharp definition of those obtained in the proteinaceous systems described above, but still manifested distinct structural phases which were more complex than plasmid DNA alone (Figure 2.11). Application of *EPD* analyses to plasmid DNA and their delivery vehicle systems is still in its infancy, but appears to be a promising approach.

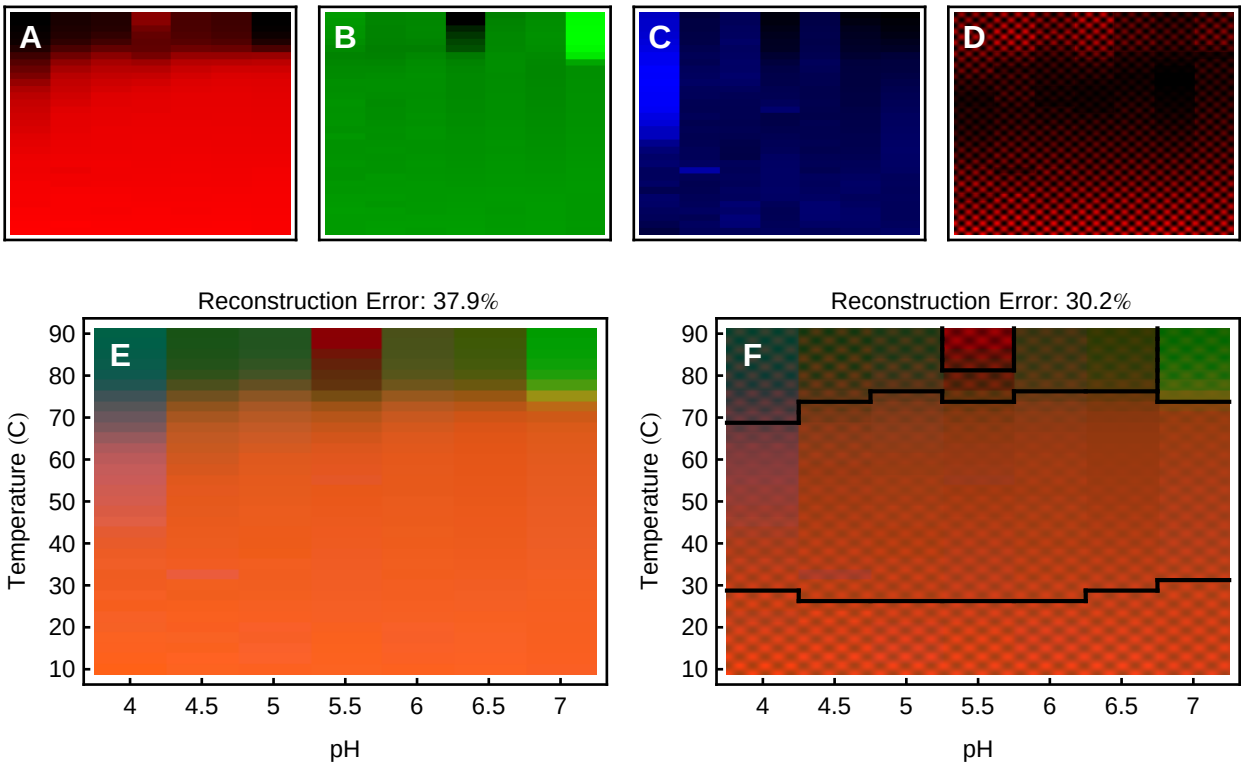


Figure 2.12: When the projection error is large it can be reduced by incorporating more dimensions. In (A)-(C), we show the primary color images (red, green and blue) of an empirical phase diagram. They are ordered by descending significance from left to right. For axis information, see (E) and (F). After solid red, green, and blue, we can use images containing structure that is smaller than the individual phase diagram blocks. This will represent high dimensional information as changes in texture. Such an image is shown in (D). (E) is a 3 dimensional empirical phase diagram of IgG, using FTIR spectra which have been preprocessed with a Fourier filter to emphasize mid-size spectral features. (F) is a 4 dimensional empirical phase diagram of the same data as (E), showing fourth dimensional information as changes in texture. Notice that the reconstruction error has decreased. The diagram has also been automatically segmented into 5 parts (see text).

2.9 Extensions of the technique

The development and use of *EPDs* has provided a high throughput method to quickly determine relative higher order conformational states of biomolecules and larger macromolecular complexes over a large "search space" using multiple biophysical techniques. The optimal determination of regions of conserved structure in the *EPD* can, however, be hindered by the presence of important structural information from multiple measurements that cannot be readily reduced to 3 dimensions for display in the *EPD*.

The *EPD* method's speed of data collection can also be hindered by the size of the

search space. Its practicality can be further limited by the complexity of data processing. In addition, in the absence of reliable automated pattern recognition, the need for an expert scientist to interpret the biophysical data to assign structural meaning to the various phases observed in the *EPD*, often on the basis of limited information, can also inhibit the method's speed, accuracy, and utility. Here we report on tactics under investigation in our laboratory to tackle and diminish such current limitations of the EPD approach.

A number of new pharmaceutical applications of *EPDs* are also being explored. These areas include extensions of the current approach to different stresses and a variety of pharmaceutical and vaccine dosage forms. In addition, possible applications of *EPDs* to describe the chemical stability of macromolecules will also be discussed. Finally, the ability of the *EPD* methodology to generate and analyze a large amount of biophysical data assessing the overall higher-order (secondary, tertiary and quaternary) structure of biomolecules as a function of solution conditions could potentially be applied to analytical comparability.

2.9.1 Maximum use of data

One typically provides the *EPD* method with a limited selection of peak positions, widths, or intensities, obtained from various experimental techniques. This results in a drastic reduction of the potential data set that precedes any global analysis. Data reduction in advance of processing is undesirable, since the excluded data may contain significant information concerning individual structural states and the transitions between them. More information would also allow the mathematical steps (PCA) to better distinguish signal from noise. To address these issues, one seeks a way to pass all of the data through a global analysis first, using minimal preprocessing. The first approach one might consider is to pass unprocessed spectra directly to the empirical phase diagram method. We have attempted this with FTIR and UV absorbance spectra, but the results do not resemble *EPDs* obtained by using the usual peak parameters (data not shown). Instead, the pH columns in the diagram show very large color differences from each other, dominating much smaller transitions in temperature

or pH. The very large pH dependent signal is presumably due to changes in the charge state of amino acids.

We might also apply preprocessing methods that are known to highlight useful information without explicitly dropping data. The second derivative of a spectrum contains information on peak position and width. We can therefore use it it to highlight information involving peak parameters. Another method is the use of a mid-pass Fourier filter to emphasize mid-size spectral features, while suppressing offsets and noise. To apply these methods, one simply filters spectra and passes them to the *EPD* method. A preliminary result is shown in Figure 2.12E, in which the Fourier mid-pass method has been applied to the FTIR spectra of an IgG molecule. The spectra covered the 900 to $4000cm^{-1}$ range, and were measured over the temperature-pH search grid shown in the *EPD*.

2.9.2 Representing more than three dimensions

The error resulting from truncating the singular value decomposition to the top 3 singular values can sometimes be large. Some criterion for what is too large, such as an error of 20% or greater, must be assigned and validated by the user. Large errors signal the presence of information that cannot be reduced to 3 dimensions. The color-coded *EPD* are limited by the colors the eye can perceive, given as ratios of red, green, and blue intensities. This is not due to a limitation in PCA or SVD: data can just as easily by projected into more than 3 dimensions. The challenge is to represent the extra dimensions. To represent more than 3 dimensions in each pixel, we can use the eye's ability to recognize shapes, textures, or other signals.

This is not to say that the number of phases that can be shown on an *EPD* is limited by the number of primary colors used to generate the *EPD*. Different ratios of red, green, and blue can generate a multitude of colors. Therefore a color coded *EPD* can display a multitude of phases. The goal of displaying more than 3 dimensions in each pixel of an *EPD* is to reduce the role of projection and the accompanying error.

We show an example of 4-dimensional visualization in Figure 2.12F. In Figures 2.12A-C, we show the primary color images (red, green and blue) of an empirical phase diagram. They are ordered by descending significance from left to right. For axis information, see Figures 2.12E and F. After solid red, green, and blue, we can use images containing structure that is smaller than the EPD pixels. This will represent information as changes in texture. Perhaps the easiest way to generate these images with small scale structure is through the use of different 2 dimensional harmonic modes, which is what we have done here. The projection error of the example in Figure 2.12 is significantly smaller for the four dimensional *EPD* than for the 3-dimensional one. In this case, the fourth principal component shows an additional transition between low and mid range temperatures.

The main reason for expressing information from the *EPD* method with three colors or selected textures is to exploit the visual processing power of the human eye and brain to segment the *EPD* into different phases. The traditional "black and white" representation of different regions in a phase diagram is also perfectly acceptable. It amounts to assigning a name, or number, for each distinctly observed and coherent region of the system's physical properties. The 15 known phases of ice are conventionally represented by 15 numbers labeling regions of the pressure-temperature diagram.

Machine learning techniques exist and are being developed to perform the task of segmenting an image. Among the techniques are clustering, support vector machines (SVM's), and Kohonen networks. The main advantage of these techniques is that they can operate on high dimensional data, reducing the role of projection and its accompanying error. In Figure 2.12F we show the results of a simple image segmentation. We have selected 5 characteristic points on the phase diagram to represent the 5 visible phases. Then, the remaining temperature-pH points have been categorized by their euclidean distance from the characteristic points, where the distance is calculated in measurement space. This is one example; mathematics and computer science possess an abundance of methods designed to recognize and organize information.

2.9.3 Information management

After an *EPD* is generated, the colors must be assigned meaning based on the information in the experimental data. The standard approach is for a scientist to assign meaning to the colors based on inspection of the original experimental data and the principal components given by PCA. As discussed earlier, however, local inspection of multidimensional data is difficult and does not maximize its utility. It should be better to pursue the assignment of meaning within a mathematical framework, allowing data to be automatically correlated with observables of interest, such as aggregation pathways and known protein conformational states.

The main difficulty in getting started with an automated approach is that raw multi-instrument digital data sets from different instruments and different experiments tend to be organizationally very complex. They involve multiple data formats and missing data points due to instrument malfunctions and differing experimental protocols. The interpretation of an archived data set often requires additional information that must be located. Without an organizing software framework it is difficult to enforce uniform, comprehensive documentation and organization of different biophysical data from diverse sources.

We have developed such an automated approach to the assignment of meaning to spectral differences. Our approach is described in the remaining chapters of this thesis.

2.9.4 New pharmaceutical applications of *EPDs*

Several new pharmaceutical applications for *EPDs* are currently being explored in our laboratories. One straightforward new application is the extension of the current *EPD* approach to different stresses and different dosage forms. As shown in Table 2.2, most *EPDs* generated to date have evaluated liquid formulations using temperature, solution pH, ionic strength and macromolecular concentration as the primary stresses to perturb the structure of a biomolecule or macromolecular complex. Additional environmental stresses that could easily be adapted to *EPD* analysis include freeze-thaw, lyophilization and shaking/agitation.

For example, in terms of development of a frozen liquid or lyophilized dosage form, the effect of multiple freeze/thaw steps as well as the effect of freeze-drying cycles and reconstitution could be evaluated using the same biophysical techniques described above. Measurements of protein conformational integrity and stability in the solid state itself could also be explored by *EPDs* using FTIR and Raman spectroscopies, as well as DSC. Identification of phase transition regions could then be used to setup an excipient screening approach for these stresses.

The *EPD* approach could also be applied to develop a better understanding of different degradation pathways as a function of environmental conditions. In the case of shaking or agitation stress, different shaking speeds, or rotations per minute, could replace temperature as a stress factor. Moreover, new biophysical analytical approaches could be added including detection of protein particles by multiflow digital imaging (MFI) or Nanosight technology. If combined with SE HPLC and OD_{350} measurements, an *EPD* could be generated to better characterize protein aggregation and subvisible particle formation. In addition, the *EPD* approach could also be used to examine chemical stability of macromolecules. For example, the rate and extent of specific Asn deamidations or Met oxidations in a protein could be mapped as a function of temperature and solution pH. These "chemical" *EPDs* could also be overlaid with conformational *EPDs* to better understand the inter-relationships(s) between chemical and physical stability.

Finally, the unique ability of the *EPD* method to use a wide variety of biophysical techniques to generate and analyze a large amount of data assessing overall structural integrity and conformational stability of biomolecules could potentially be applied to analytical comparability during development of different biopharmaceutical drugs and vaccines. For example, since the *EPD* method does not require much protein (1-10 mg), and since availability of protein is often a limiting factor in early formulation development, the generation of *EPDs* for different candidate molecules could be used as a tool to select the best candidate in terms of "developability" properties such as stability and solubility profiles. Moreover,

during later development, process and product changes are usually required to scale up the process for commercial use. These changes often lead to subtle or more dramatic changes in the biomolecules post-translational modifications or degradation profiles (e.g., glycosylation pattern or extent of oxidation of a specific Met residue). The ability to monitor the effect of these changes on the overall structural integrity and conformational stability of biomolecules remains an area of ongoing interest, especially as a possible surrogate for more complex assessments of conformation such as biological assays. The ability of the *EPD* approach to compare the same biomolecule with differing glycosylation patterns and/or chemically altered amino acid residues is currently being evaluated.

An application of the *EPD* method that could potentially have great impact may be to drastically reduce the size of high throughput screening searches to identify stabilizing excipients. The accelerated time-lines of modern drug formulation efforts, and the complexity and size of the search spaces involved, typically result in suboptimal screening.[5,77] The limited procedures available to screen a wide formulation design space can often result in suboptimal formulations or potentially even product failure during long term storage. A brute-force approach would test conformational and chemical stability at every relevant solvent condition. This approach is, however, cost prohibitive because of the exponentially large number of variables. For example, if one tested 5 different excipients at 4 different concentrations each, the number of combinations to test would be 4^5, or 1024 experiments. The use of empirical phase diagrams permits the size of these high throughput screening search spaces to be reduced in a very natural and pragmatic way. Using *EPDs*, macromolecule identity has been found to be conserved over contiguous regions of search space. The identification of unique and/or consistent conformational states reduces the search space from an exponentially large and unexplorable one to one that is much smaller yet adapted to the system of interest. More time consuming and extensive excipient screening and analytical characterization tests can then subsequently be performed on the smaller set of conditions to better design and develop optimized formulation conditions.

2.10 Conclusion

Modern biopharmaceutical drug development time-lines, combined with limited availability of sufficient material, can result in a variety of challenges for the formulation scientist attempting to rapidly design and develop stable dosage forms for clinical use. Our goal in the development of the EPD technique has been to enable faster and more thorough screening searches of stabilizing agents and solution conditions by more fully utilizing the information contained in data sets from experimental methods which examine the structural integrity and conformational stability of macromolecules and their complexes. We strive to explore as much of the available search space as possible, using mathematical techniques to obtain the maximum amount of information from the data.

The optimal way to reduce the dimensionality of data is by use of the singular value decomposition (SVD). SVD returns a number of spatial axii, defining spaces on which the data can be approximated. The approximation error can be minimized by using a space defined by the most important spatial axii. The projection onto that space is the best possible approximation to the data that can be made on the number of dimensions retained. To provide a convenient visual image, the resulting low-dimensional positions can be converted into colors or textures and presented as an image. Such an image is called an empirical phase diagram (*EPD*). The empirical phase diagram method guides the formulation scientist by assisting in the visualization of high dimensional information, the determination of macromolecule identity and transition points, and a reduction of the size of search spaces. This approach is quite different from that of the commonly used "Design of Experiment" approach which lacks high density data and produces holes in the picture produced.

The *EPD* method has found many uses in the optimization of various types of formulations, and many case studies have been published concerning their application to various macromolecular complexes such as viruses and lipoplexes. The *EPD* approach has been extended over time to include the addition of multiple biophysical measurement techniques and different search space variables. The use of empirical phase diagrams is not limited to proteins

and plasmid DNA molecules, but includes larger macromolecular complexes such as viruses and whole cells. One can potentially incorporate almost any kind of information, including measurements of structural dynamics, aggregation kinetics, chemical stability or biological function as well as other common pharmaceutical variables of stress such as agitation and freeze/thaw cycles. Empirical phase diagrams have also been demonstrated to contain information concerning the functional and evolutionary relationships of proteins.[10–12,16,17] Using *EPDs*, macromolecule identity has been found to be conserved over contiguous regions of search space.

The use of *EPDs* has brought us to a vantage point where we see clear evidence for a previously unrealized treasure trove of hidden information concerning the higher order structural integrity and conformational stability of biomolecules and larger macromolecular complexes such as viruses and lipoplexes. Much work remains, however. Data consists of combinations of different types of information. Each type is mixed with other types in complex ways, and has its own meaning, prominence in the data, and significance for the task at hand. The inter-relationships between these factors is complex, requiring systematic study within a mathematical framework.

Bibliography

[1] D S Pisal, M P Kosloski, and S V Balu-Iyer. "Delivery of Therapeutic Proteins". In: *J. Pharm. Sci.* 99 (2010), pp. 2557–75.

[2] S K Singh. "Impact of Product-Related Factors on Immunogenicity of Biotherapeutics". In: *J. Pharm. Sci.* 100 (2011), pp. 354–87.

[3] A Lubiniecki et al. "Comparability assessments of process and product changes made during development of two different monoclonal antibodies". In: *Biologicals* 39 (2011), pp. 9–22.

[4] J D Ramsey et al. "Using empirical phase diagrams to understand the role of intramolecular dynamics in immunoglobulin G stability". In: *J. Pharm. Sci.* 98 (2009), pp. 2432–47.

[5] H Fan et al. "Solution behavior of IFN-β-1a: An empirical phase diagram based approach". In: *J. Pharm. Sci.* 94 (2005), pp. 1893–911.

[6] L A Kueltzo et al. "Derivative absorbance spectroscopy and protein phase diagrams as tools for comprehensive protein characterization: A bGCSF case study". In: *J. Pharm. Sci.* 92 (2003), pp. 1805–20.

[7] A Nonoyama et al. "A biophysical characterization of the peptide drug pramlintide (AC137) using empirical phase diagrams". In: *J. Pharm. Sci.* 97 (2008), pp. 2552–67.

[8] N Harn et al. "Highly concentrated monoclonal antibodies: Direct analysis of structure and stability". In: *J. Pharm. Sci.* 96 (2007), pp. 532–46.

[9] D.T. Brandau et al. "Stability of the Clostridium botulinum type A neurotoxin complex: An empirical phase diagram based approach". In: *Molecular Pharmaceutics* 4.4 (2007), pp. 571–582.

[10] H. Fan, R.S. Kashi, and C.R. Middaugh. "Conformational lability of two molecular chaperones Hsc70 and gp96: Effects of pH and temperature". In: *Archives of biochemistry and biophysics* 447.1 (2006), pp. 34–45.

[11] H Fan et al. "Effects of solutes on empirical phase diagrams of human fibroblast growth factor 1". In: *J. Pharm. Sci.* 96 (2007), pp. 1490–503.

[12] H. Fan et al. "Effects of pH and polyanions on the thermal stability of fibroblast growth factor 20". In: *Molecular Pharmaceutics* 4.2 (2007), pp. 232–240.

[13] GE Jiang et al. "Anthrax vaccine powder formulations for nasal mucosal delivery". In: *J. Pharm. Sci.* 95.1 (2006), pp. 80–96.

[14] A.P. Markham et al. "pH sensitivity of type III secretion system tip proteins". In: *Proteins: Structure, Function, and Bioinformatics* 71.4 (2008), pp. 1830–1842.

[15] L.J. Peek et al. "A systematic approach to stabilizing EBA-175 RII-NG for use as a malaria vaccine". In: *Vaccine* 24.31-32 (2006), pp. 5839–5851.

[16] M S Salnikova et al. "Physical characterization of *Clostridium difficile* toxins and toxoids: Effect of the formaldehyde crosslinking on thermal stability". In: *J. Pharm. Sci.* 97 (2008), pp. 3735–52.

[17] B S Barrett et al. "The response of type three secretion system needle proteins $MxiH^{\Delta 5}$, $BsaL^{\Delta 5}$, and $PrgI^{\Delta 5}$ to temperature and pH". In: *Proteins* 73 (2008), pp. 632–43.

[18] R Esfandiary et al. "Structural stability of vault particles". In: *J. Pharm. Sci.* 98 (2009), pp. 1376–86.

[19] N Thyagrajapuram, D Olsen, and C R Middaugh. "The structure stability and complex behavior of recombinant human gelatins". In: *J. Pharm. Sci.* 96 (2007), pp. 3363–78.

[20] K. Zheng, C.R. Middaugh, and T.J. Siahaan. "Evaluation of the physical stability of the EC5 domain of E-cadherin: Effects of pH, temperature, ionic strength, and disulfide bonds". In: *J. Pharm. Sci.* 98.1 (2009), pp. 63–73.

[21] F He et al. "Structural stability of hepatitis C virus envelope glycoprotein E1: Effect of pH and dissociative detergents". In: *J. Pharm. Sci.* 98 (2008), pp. 3340–57.

[22] J Kissmann et al. "H1N1 influenza virus-like particles: Physical degradation pathways and identification of stabilizers". In: *J. Pharm. Sci.* 100 (2010), pp. 634–45.

[23] S F Ausar et al. "Conformational stability and disassembly of Norwalk virus-like particles: Effect of pH and temperature". In: *J. Biol. Chem.* 281 (2006), pp. 19478–88.

[24] J Kissmann et al. "Physical stabilization of norwalk virus-like particles". In: *J. Pharm. Sci.* 97 (2008), pp. 4208–18.

[25] S.F. Ausar et al. "Analysis of the thermal and pH stability of human respiratory syncytial virus". In: *Molecular Pharmaceutics* 2.6 (2005), pp. 491–499.

[26] J Rexroad, R K Evans, and C R Middaugh. "Effect of pH and ionic strength on the physical stability of adenovirus type 5". In: *J. Pharm. Sci.* 95 (2006), pp. 237–47.

[27] J Rexroad et al. "Thermal stability of adenovirus type 2 as a function of pH". In: *J. Pharm. Sci.* 95 (2006), pp. 1469–79.

[28] M Ruponen, C S Braun, and C R Middaugh. "Biophysical characterization of polymeric and liposomal gene delivery systems using empirical phase diagrams". In: *J. Pharm. Sci.* 95 (2006), pp. 2101–14.

[29] Y. Zeng et al. "Towards development of stable formulations of a live attenuated bacterial vaccine". In: *Human Vaccines* 5.5 (2009), pp. 322–331.

[30] H. Mach et al. "Ultraviolet absorption spectroscopy". In: *Methods Mol Biol* 40 (1995), pp. 91–114.

[31] L H Lucas et al. "Probing protein structure and dynamics by second derivative ultraviolet absorption analysis of cation-pi interactions". In: *Protein Sci.* 15 (2006), pp. 2228–43.

[32] H Mach, C R Middaugh, and R V Lewis. "Detection of proteins and phenol in DNA samples with second-derivative absorption spectroscopy". In: *Anal. Biochem.* 200 (1992), pp. 20–26.

[33] S F Ausar, S B Joshi, and C R Middaugh. "Spectroscopic methods for the physical characterization and formulation of nonviral gene delivery systems". In: *Methods Mol. Biol.* 434 (2008), pp. 55–80.

[34] K Nakanishi, N Berova, and R W Woody. *Circular dichroism - principles and applications*. New York: VCH Publishers Inc., 1994.

[35] N Sreerama et al. "Tyrosine, phenylalanine, and disulfide contributions to the circular dichroism of proteins: Circular dichroism spectra of wild-type and mutant bovine pancreatic trypsin inhibitor". In: *Biochemistry* 38 (1999), pp. 10814–22.

[36] S Y Venyaminov and J T Yang. "Determination of protein secondary structure". In: *Circular dichroism and the conformational analysis of biomolecules*. Ed. by G D Fasman. New York: Plenum Press, 1996, pp. 69–107.

[37] W Jiskoot et al. "Fluorescence Spectroscopy". In: *Methods for structural analysis of protein pharmaceuticals*. Ed. by W Jiskoot and D. Crommelin. Arlington, VA: AAPS Press, 2005, pp. 27–82.

[38] M. Cardamone and NK Puri. "Spectrofluorimetric assessment of the surface hydrophobicity of proteins". In: *Biochemical Journal* 282.2 (1992), p. 589.

[39] L Stryer. "The interaction of a naphthalene dye with apomyoglobin and apohemoglobin: A fluorescent probe of non-polar binding sites." In: *J. Mol. Biol.* 13 (1965), pp. 482–95.

[40] R Guntern et al. "An improved thioflavin S method for staining neurofibrillary tangles and senile plaques in Alzheimer's disease". In: *Experientia* 48 (1992), pp. 8–10.

[41] W E Klunk, J W Pettegrew, and D J Abraham. "Quantitative evaluation of congo red binding to amyloid-like proteins with a beta-pleated sheet conformation". In: *J. Histochem. Cytochem.* 37 (1989), pp. 1273–81.

[42] H. Levine. "Thioflavine T interaction with synthetic Alzheimer's disease beta-amyloid peptides: Detection of amyloid aggregation in solution". In: *Protein Science* 2.3 (1993), pp. 404–410.

[43] H Levine. "Quantification of beta-sheet amyloid fibril structures with thioflavin T". In: *Methods Enzymol.* 309 (1999), pp. 274–84.

[44] AP Demchenko. "Red-edge-excitation fluorescence spectroscopy of single-tryptophan proteins". In: *European Biophysics Journal* 16.2 (1988), pp. 121–129.

[45] A P Demchenko. "The red-edge effects: 30 years of exploration". In: *Luminescence* 17 (2002), pp. 19–42.

[46] J M Beechem and L Brand. "Time resolved fluorescence decay in proteins". In: *Ann. Rev. Biochem.* 54 (1985), pp. 43–71.

[47] J.R. Alcala, E. Gratton, and FG Prendergast. "Fluorescence lifetime distributions in proteins". In: *Biophysical journal* 51.4 (1987), pp. 597–604.

[48] D.M. Byler and H. Susi. "Examination of the secondary structure of proteins by deconvolved FTIR spectra". In: *Biopolymers* 25.3 (1986), pp. 469–487.

[49] W K Surewicz and H H Mantsch. "New insight into protein secondary structure from resolution enhanced infrared spectra". In: *Biochim. Biophys. Acta* 952 (1988), pp. 115–30.

[50] D Aichun, H Ping, and S C Winslow. "Protein secondary structures in water from second-derivative amide I infrared". In: *Biochemistry* 29 (1990), pp. 3303–08.

[51] G.J. Thomas Jr. "Raman spectroscopy of protein and nucleic acid assemblies". In: *Annual review of biophysics and biomolecular structure* 28.1 (1999), pp. 1–27.

[52] E. Freire. "Differential scanning calorimetry". In: *Methods Mol. Biol* 40 (1995), pp. 191–218.

[53] T J Kamerzell and C R Middaugh. "The complex inter-relationships between protein flexibility and stability". In: *J. Pharm. Sci.* 97 (2008), pp. 3494–517.

[54] A Cooper et al. "Heat does not come in different colours: Entropy-enthalpy compensation, free energy windows, quantum confinement, pressure perturbation calorimetry, solvation and the multiple causes of heat capacity effects in biomolecular interactions". In: *Biophys. Chem.* 93 (2001), pp. 215–230.

[55] P D H Heerklotz. "Pressure perturbation calorimetry". In: *Methods Mol. Biol.* 400 (2007), pp. 197–206.

[56] A P Sarvazyan. "Ultrasonic velocimetry of biological compounds". In: *Ann. Rev. Biophys. Biophys. Chern.* 20 (1991), pp. 321–42.

[57] J Berne and R Pecora. *Dynamic light scattering with applications to chemistry, biology, and physics*. New York: Dover, 2000.

[58] C M Wiethoff and C R Middaugh. "Light-scattering techniques for characterization of synthetic gene therapy vectors". In: *Nonviral vectors for gene therapy: Methods and protocols*. Ed. by M A Findeis. Totowa, NJ: Humana Press Ind., 2001, pp. 349–76.

[59] GB Irvine. "Size-exclusion high-performance liquid chromatography of peptides: a review". In: *Analytica chimica acta* 352 (1997), pp. 387–397.

[60] M D Bond et al. "Evaluation of a dual-wavelength size exclusion HPLC method with improved sensitivity to detect protein aggregates and its use to better characterize degradation pathways of an IgG1 monoclonal antibody". In: *J. Pharm. Sci.* 99 (2010), pp. 2582–97.

[61] K Ding, J M Louis, and A M Gronenborn. "Insights into conformation and dynamics of protein GB1 during folding and unfolding by NMR". In: *J. Mol. Biol.* 335 (2004), pp. 1299–1307.

[62] Y. Aubin, G. Gingras, and S. Sauve. "Assessment of the three-dimensional structure of recombinant protein therapeutics by NMR fingerprinting: Demonstration on recombinant human granulocyte macrophage-colony stimulation factor". In: *Analytical Chemistry* 80.7 (2008), pp. 2623–2627.

[63] H. Frauenfelder and P.G. Wolynes. "Biomolecules: where the physics of complexity and simplicity meet". In: *Physics Today* 47.2 (1994), pp. 58–66.

[64] A Cooper. "Thermodynamic fluctuations in protein molecules". In: *Proc. Natl. Acad. Sci. USA* 73 (1976), pp. 2740–41.

[65] G.W. Stewart. "On the early history of the singular value decomposition". In: *SIAM review* (1993), pp. 551–566.

[66] J R Chavez and A Y Kwarteng. "Extracting spectral contrast in Landsat Thematic Mapper image data using selective principal component analysis". In: *Photogrammetric Engineering and Remote Sensing* 53 (1989), pp. 339–48.

[67] D J Werring et al. "The structural and functional mechanisms of motor recovery: Complementary use of diffusion tensor and functional magnetic resonance imaging in a traumatic injury of the internal capsule". In: *J Neurol Neurosurg Psychiatry* 65 (1998), pp. 863–69.

[68] T. Twellmann et al. "Image fusion for dynamic contrast enhanced magnetic resonance imaging". In: *Biomedical engineering online* 3.1 (2004), p. 35.

[69] J.R. Mansfield et al. "Near infrared spectroscopic reflectance imaging: a new tool in art conservation". In: *Vibrational spectroscopy* 28.1 (2002), pp. 59–66.

[70] J E Steiner et al. "PCA tomography: How to extract information from data cubes". In: *Mon. Not. R. Astron. Soc.* 295 (2009), pp. 64–75.

[71] L.J. Peek, R.N. Brey, and C.R. Middaugh. "A rapid, three step process for the preformulation of a recombinant ricin toxin A chain vaccine". In: *J. Pharm. Sci.* 96.1 (2007), pp. 44–60.

[72] G.G. Hammes. "Multiple conformational changes in enzyme catalysis". In: *Biochemistry* 41.26 (2002), pp. 8221–8228.

[73] A. Kohen et al. "Enzyme dynamics and hydrogen tunnelling in a thermophilic alcohol dehydrogenase". In: *Nature* 399.6735 (1999), pp. 496–499.

[74] J Kraut. "How do enzymes work?" In: *Science* 242 (1988), pp. 533–40.

[75] A Roldao et al. "Virus-like particles in vaccine development". In: *Expert Rev. Vaccines* 9 (2010), pp. 1149–76.

[76] S S El-Kamary et al. "Adjuvanted intranasal Norwalk virus-like particle vaccine elicits antibodies and antibody-secreting cells that express homing receptors for mucosal and peripheral lymphoid tissues". In: *J. Infect. Dis.* 202 (2010), pp. 1623–5.

[77] T J Gibson et al. "Application of a high-throughput screening procedure with PEG-induced precipitation to compare relative protein solubility during formulation development with IgG1 monoclonal antibodies". In: *J. Pharm. Sci. in press* 100 (2011), pp. 1009–1021.

Chapter 3

DART: a new programming language for declarative array transformation

3.1 Introduction

The project discussed in this chapter was motivated by the need to regularize, analyze, and plot the high dimensional, multiple instrument data sets that are encountered in the formulation of biopharmaceuticals.

The field of pharmaceutical formulation commonly uses spectroscopic instruments due to their sensitivity, appropriate information content, availability, reliability, and high throughput. A large formulation data set may cover 32 x 6 x 30 x 2 combinations of formulation and perturbation conditions.[1] At each combination a measurement is collected, which might employ 3 instruments that return spectra of dimensions 40, 1024, and 100 x 100. Processing these data sets involves operations such as mapping filenames to dimension positions, discarding noisy or corrupt data, handling data that is non-existent due to instrument malfunctions and operator error, subtracting reference spectra, performing statistical and signal processing operations, generating complex plots, and performing original data analysis research. These tasks are typically performed by hand and by human judgment using a combination of many programs such as Excel, Origin, instrument software, Matlab, Mathematica, and custom scripts. Even for data sets of modest size, a generous amount of labor is needed to

perform the processing steps. In addition, the unstructured nature of the process makes it difficult to document.

Thus the increasing size and complexity of pharmaceutical formulation data sets has created the need for a tool to simplify, automate, and document the processing of data. This tool should be capable of advanced multidimensional data processing, yet remain simple enough to be used by non-programmers. It should also provide built in support for almost any math operation that might be desired, provide a record of changes to data, and ease the task of regularizing data sets filled with inconsistent array shapes and missing data.

The challenges above are shared by other fields that generate large multidimensional, multi-instrument data sets, and solutions are being studied in astronomy, microarray data analysis, and high-throughput biology and chemistry.[2–4] As science explores the universe utilizing scientific instruments that incorporate a growing number and variety of sensors, experimental data sets become larger and more complex. This results in greater difficulty of developing data analysis solutions that are verifiably correct, well documented and efficiently maintainable. This difficulty affects all users of data, from the least to the most technical. Users without programming experience often face data analysis tasks that are too complex to perform efficiently using conventional tools. The productivity of expert analysts is reduced by the need to write code connecting subsystems that ideally would inter-operate automatically. Users of all types must communicate increasingly complex methods and results to one another. Since there is no end in sight for the trend toward more complex instruments and corresponding data, the existing ecosystem of information management and data analysis tools is destined to become inadequate.

One approach to the problem is the development of declarative languages for array manipulation.[2,4] These languages define more complete array data types than traditional languages, including dimension names, units, and dimension scale positions, along with processing methods that hide procedural details. This allows the development of intuitive languages with easier learning curves, while still enabling complex data processing.

Another approach to conquering large and complex data analysis projects is the development of methods and frameworks for enhancing the communication and reproducibility of data analysis methods. Computational data analyses in scholarly work are often not reproducible due to missing parameter values, codes, proofs, high-level descriptions or implementation details.(Vandewalle et al., 2009) The study of solutions to this problem is known as the field of computational reproducibility, and has the goal of ensuring that persons other than the authors of a scholarly work can reproduce computations used in that work.[5–11] It was largely initiated by Jon Claerbout and Donald Knuth, who separately developed systems capable of automatically regenerating text, computations, and figures for published works, a strategy known as literate programming.[8,12] Other strategies include publishing code on websites and using open source software, high level languages, and frameworks that allow tracking the history of data manipulation steps. The benefits of reproducibility are better work habits, improved teamwork, greater impact on the scientific community, and increased continuity of research.(Donoho, 2010) Low reproducibility of computations can obscure errors. In the medical field these errors can potentially lead to unsafe activities.[7]

The general strategy chosen here was to combine the two approaches just described. We have created a declarative array transformation language (DART) overlying mathematics software that already possesses built-in support for literate programming and high level mathematics. DART is a language in the sense that one can use it to express data analysis projects in their entirety. Furthermore, its data types and functions are partitioned into orthogonal areas of functionality and are designed to be used in unforeseen creative combinations. Nearly all functions in the DART language accept and return arrays while handling dimensions and units automatically, making it easy to synthesize new operations by combining existing ones. To facilitate the processing of heterogeneous data sets, functions of arrays also accept lists of arrays of differing shapes, as long as the function's operation is well defined for each array shape. A comprehensive set of commands for merging, splitting, and renaming dimensions and dimension positions assist with regularization of data. Array

functions are also provided for importing, exporting, and plotting data, constructing tables, and interactively browsing arrays.

This chapter presents the architecture of DART, including its data types, command set, and strategies for achieving general data processing goals. DART is not only a specification, but a working prototype consisting of around 100 documented functions, a help index, tutorials, and several complex examples.

3.2 Design Strategies

3.2.1 Declarative array transformation syntax

A declarative language is one that expresses programs without using control flow constructs such as looping and if/then commands. In order to create a declarative array transformation syntax, it was necessary to create data types corresponding to the properties of arrays so that these properties could be referred to by name in array manipulation commands.

Data types

The array data type found in conventional languages is a minimalist implementation that leaves out much of the labeling information that must typically be associated to arrays. As implemented in conventional languages, an array is a multidimensional grid containing an object in each cell of the grid. In practice, however, arrays must be related to many types of subsidiary data, including labels for the dimensions, labels for the positions along each dimension, and units for the dimension positions and the data. In standard practice, the use of these properties is not automated. Methods must typically be coded to handle dimensions and units when importing, exporting, processing, and plotting.

DART defines three data types to support automatic handling of array metadata and provide an efficient declarative syntax. The first data type is the *dimension*, which is a data structure consisting of a name for the dimension, stored as a string, the unit for its scale, also

stored as a string, and a scale. The scale of a dimension is an ordered list of values indexing that dimension of the array. The values in the scale must all be of the same data type, and the data type must be a string, integer, real number, or other data type that has a defined ordering relation. The second data type is the *array*, which is a data structure consisting of an ordered list of dimensions and a conventional array of data. The order of dimensions in the list of dimensions must match the order of dimensions in the array. The length and ordering of the scale in each dimension must match the array's respective dimension. The values in the array can be of any type, including strings, plots, or even further DART arrays. Zero dimensional arrays are allowed, in which case the list of dimensions is empty and the data array is a single value. The third and final data structure is simply a list of DART arrays.

Mechanisms

An efficient declarative syntax has been achieved in primarily three ways. First, dimensions are inherent to the array data type. This enables an increase in syntactical parsimony over comparable languages such as SciDB and OLAP. (In this section we will sometimes compare DART to certain other languages known as SciDB and OLAP. It is impossible to describe them in full detail here, but we hope that the comparisons will be clear on general grounds.) In a call to a DART array operation, dimensions can simply be referred to by name. In SciDB and OLAP, a join must be specified between an array and a dimension whenever dimension information is needed. This more general behavior is useful when multiple dimensions may be related to one dimension of an array, or when multiple arrays share a dimension and care must be taken that there is only one version of the dimension. DART architecture, however, would allow such ad-hoc joining of dimensions to arrays, since in DART the dimensions that are inherent to arrays are required to be unique indexes.

The second manner in which efficient declarative syntax has been achieved is by ensuring that full array structure, including dimensions, passes through most DART functions. In

other words, functions that operate on DART arrays usually return DART arrays. This allows functions of arrays to be strung together without glue functions.

The combination of the two strategies above permits code that is similar to natural language. For example, let a be an array with dimensions named "Temperature", "pH", and "Wavelength". In the following code snippet

```
a = operate[gaussianFilter[#, {5, 3}]&, a, {``Wavelength'', ``Temperature''}];
a = operate[listPlot, a, {``Wavelength''}];
a = operate[table, a, ``pH''];
browse[a]
```

three subsystems inter-operate seamlessly: mathematical operations, plotting, and interactive browsing. (A note on Mathematica syntax: functions use the syntax function[arguments] , lists use the syntax element 1, element 2, ... and are nestable. The syntax (expression using a # symbol)& creates an anonymous function.)

- In the first step of the above code, a gaussian filter is applied to every (wavelength x temperature) subarray and threaded over the remaining dimensions. The function gaussianFilter takes and returns DART arrays. The DART arrays supplied to gaussianFilter contain dimension unit information, which is used to interpret the filter widths as 5nm and 3C.

- The second line of code takes each wavelength subarray and makes a line plot for that subarray. Wavelength dimension axis information is used to automatically label the x-axis, and the y-axis is the numerical value in the array. The listPlot function returns a zero dimensional array containing a single plot, with the result that the second line of code drops the wavelength dimension, giving a DART array with dimensions "Temperature" and "pH".

- The third line takes the subarray of plots for each pH and makes a table of plots. The table function takes an array of up to 2 dimensions, draws a table, and returns a

zero dimensional array. Rows and columns are automatically labeled with dimension information. When one dimensional arrays are passed to the table function (as above), the default behavior is to draw a table with a single row. The table elements can be of any type that Mathematica can display, including plots, numbers or strings. In the third line of code above the pH dimension is used in the tables and dropped, so the result is an array with a "Temperature" dimension.

- The browse function in the fourth line takes a DART array and constructs an interactive GUI element with sliders for each dimension in the array and a display area for the array element selected by the sliders. In this case only one slider is constructed for the temperature dimension.

Small modifications to the code above can produce highly varying outputs. For example, if tables of numerical values are desired, the second line could be deleted and the wavelength dimension added to the third line. The table function could be used twice in a row to construct nested tables. It is also significant that the above code snippet would not be different for an array with many more dimensions. Instead of one slider for temperature, additional sliders would be shown, one for each additional dimension.

The third manner in which DART achieves efficient declarative syntax is by making functions of DART arrays capable of operating on a list of arrays of differing shapes. For example, let a be defined as above, and let b have dimensions named "Temperature", "pH", "Excitation Wavelength", and "Emission Wavelength". Furthermore, let x be a list comprised of a and b (in Mathematica notation x=a,b). In DART the following operation is well defined.

```
x = operate[gaussianFilter[#, 3]&, x, ‘‘Temperature’’];
```

Even though a and b have different shapes, they both have the temperature dimension, so the filtering operation is defined for both a and b.

3.2.2 Generality and Extendability

The goal of making DART capable of general processing yet extendable by expert users has been supported by designing the system in tiered fashion, with each tier open for use by programmers. At the lowest tier are constructors, getters and setters for arrays and dimensions. In the next tier are functions for dimension management. The next tier contains array transformation operations such as flattening arrays along specified dimensions, merging arrays, selecting subarrays, threading operations over subarrays, etc. In the highest tier are data analysis functions for standardization, filtering, clustering, and matrix decompositions, along with plotting, tabling, and GUI data browsing operations. Each tier uses the functionality of the tiers below. The functions in each tier tend to be short and readable, and functionality unfolds incrementally as one progresses through the tiers.

3.2.3 Computational reproducibility

DART supports fully documented, computationally reproducible data analyses due to it being implemented within a computer algebra system (CAS) possessing built-in support for literate programming and high level mathematics. DART uses Mathematica, but a similar system such as Maple or Sage would also have been suitable. A program or script written in one of these systems can be laid out typographically with full font freedom and display of images, drawings, tables, plots, and typeset equations. In addition to supporting symbolic mathematics and literate programming, these systems interface with standard numeric libraries for linear algebra, signal processing, and more, as well as providing extensive high level scripting primitives for importing and exporting data.

A scripting environment promotes computational reproducibility by providing a record of changes to data. This record of changes can be brought about by modifying data exclusively within scripts and saving scripts along with input and output data. In a medical or pharmaceutical research setting, administration of file use privileges would serve to maintain data authenticity and auditability (i.e. maintain a trail of information that allows the final

data to be audited).

A different strategy for maintaining the auditability of data is to never overwrite data, but rather to maintain an automatic record of changes. In this manner data modifications are tracked involuntarily. SciDB, for instance, stores a fine-grained record of changes to individual elements of arrays. This is a very storage efficient strategy in applications where small parts of an array are changed while most of the array remains unchanged. Data analyses, however, commonly modify entire arrays during standardization, filtering, and analysis operations. In this situation it is more storage efficient to track changes by storing scripts. Intermediate results do not need to be stored, because in order to understand changes to data one has only to read the script that produced the changes. Storing scripts also promotes greater reproducibility: knowing the operation that changed an array allows one to deduce the resulting numerical change in the array. Knowing the numerical change, however, may not tell one what the operation was.

Scripting also reduces labor relative to traditional point-and-click data analyses. Entire analyses can be customized and run in minutes, as opposed to days or weeks. This is useful in high throughput experiments when one may need to know the results of an experiment prior to initiating the next experiment. It is also useful in the development of new methods of data analysis, since the result of modifying an analysis method can be evaluated quickly.

3.3 Feature Set

We now discuss the functions available in DART, which cover the full range of capabilities necessary for complex multidimensional analysis. In addition to basic language features enabling declarative array transformation, DART also has functions for reshaping arrays, performing statistical and signal processing operations, and tabling, plotting, and interactively browsing data. These features will be discussed in the order in which they appear in a typical data processing script. For an example of a script written using DART, see the

appendices to this dissertation.

3.3.1 Importing and collating data

Before DART can operate on an array it must be made into a DART array. Functions are provided for importing from a few formats such as Excel files and various instrument formats. Import functions for other formats can easily be coded. Internally, these functions read data using Mathematica's import functions, then use the DART functions newDimension[] and newArray[] to construct a DART array. Importing single arrays is thus a simple matter. Much more, however, is required to collate complex data sets.

The essential problem in collating data sets is that of assigning arrays in a set of files to the correct portions of a larger array or set of arrays. A file's position in the larger set of arrays may be found in filenames or folder names, inside files, or come from another source such as a laboratory notebook. The subarrays can sometimes be out of order and grouped in the wrong folders. For instance, a spectroscopic instrument with 6 optical cell holders may be used to collect data of length 9 in the cell dimension. Or, at the end of such an experiment some cells that gave corrupt data may be rerun. Files can also be missing due to instrument glitches. The best collating solution will vary from instrument to instrument and application to application. DART therefore provides a few elementary functions that can be used to build custom collating functionality.

In bookbinding, collating involves 4 steps: labeling items, separating them into groups by their labels, combining them within the groups, then combining the groups. The same four steps are found in data analysis. A useful technique for the first step, that of labeling data sets, goes as follows. Suppose we are importing a data set consisting of a set of files, each file containing an array of the same dimensions as all the others. Further assume that each file corresponds to an experiment at a different temperature. To import this data set, we import each file into a different DART array, and as we create each array we add to it a temperature dimension containing only 1 position: the temperature for that experiment.

In DART this is performed using the addSingletons[] function, which accepts a DART array and a list of singleton dimensions, and returns a DART array with the singleton dimensions added. These singleton dimensions then function as labels in the rest of the collating process.

The second step of collating, that of separating items into groups by their labels, is accomplished in DART with the select[] function. The syntax for the function is

```
select[dartArray, dimName1, anonFunc1, dimName2, anonFunc2, ...],
```

where dimName1 is the name of a dimension in dartArray, and anonFunc1 is a boolean valued anonymous function applied to the positions in dimension dimName1 to determine whether those positions will be included in the output array. When no positions in a dimension are selected, the select[] function returns a single Null for the entire array. The select[] function also takes a list of DART arrays, in which case it is applied separately to each array in the list.

The third step of collating, that of combining items within groups, is done by use of the merge[] function, which accepts a list of DART arrays and returns a single array containing the data from all of the arrays passed to it. It works in the following manner. First it verifies that all the arrays in the list share the same dimension names. Then, for each dimension name, it finds the set of all positions used in that dimension in all the arrays passed to it. It then initializes a single large array containing those dimension names and positions, and copies the data from the sub arrays into the single large array.

The final step of collating, that of combining groups into a single package, is done simply by storing multiple DART arrays in a single variable as a Mathematica list. This variable can then be passed to DART functions that accept lists of DART arrays.

3.3.2 Analysis

Once data has been imported and collated into a list of regular arrays, analysis can begin. The most frequently used DART analysis function is the operate[] function, with the syntax

operate[dartArray, operation[], dimNames]. The function operation[] is threaded over the array dartArray, taking subarrays of dimensions dimNames and threading over the remaining dimensions in dartArray. If operation[] is a function that accepts a DART array, operation[] is allowed to return an array of different dimensions than the array passed to it, but must return an array of the same dimensions on every call. If operation[] is a function that does not accept DART arrays but traditional arrays instead, operation[] is not allowed to change the dimensions of the arrays passed to it. This is not allowed because if operation[] changed the dimensions of the arrays without returning dimensional information, operate[] would not have the information required to construct the output array. The function operate[] can also optionally be given a list of dimensions that are both threaded over and operated on. These dimensions are passed to operation[] as singletons, and can be used by operation[] as an index of the current position in the larger array.

DART overloads the arithmetic operations +, -, ×, ÷ and ^ so that they can be used on DART arrays and combinations of DART arrays and numbers. Dimension names that are in one array but not the other are broadcast before the operation is applied. When both operands are DART arrays, any dimension names in common between the arrays are required to have the same dimension positions.

A variety of statistical and signal processing functions are available within DART, such as noise estimation, Fourier filtering, multidimensional kernel filtering, Finite Impulse Response (FIR) filtering (including Gaussian and Savitzky-Golay filters), singular value decomposition (SVD) based filtering, interpolation, and more. More functionality is easy to develop, as Mathematica functions can be repackaged to take advantage of dimension information.

Several functions are available for reshaping arrays. The fuse/unfuse combination is useful for performing principal components analysis (PCA) or partial least squares regression (PLS) on multi-way data. The fuse[] function flattens an array along specified dimensions, combining those dimensions into one. The unfuse[] function does the opposite, restoring the original dimensions from the flattened dimension. When the shapes of arrays differ,

they can be joined using the concatenate/unconcatenate pair of functions. These functions are useful for performing PCA or PLS on multi-instrument data. The functions fuse[] and concatenate[] can be combined to analyze multi-instrument, multi-way data. For details on how these functions work, see the examples in the help index in the supplementary material.

3.3.3 Visualization

DART also provides several data visualization functions that automatically use array dimension information in advanced ways. For example, when the listPlot[] function is given an array possessing two dimensions, multiple lines are drawn on the same plot in different colors, and a legend is drawn next to the plot using the scale of the second dimension. This legend can optionally be suppressed, for example when multiple plots are arranged in a grid. If the array passed to listPlot[] includes a dimension named "moments", listPlot[] automatically uses the second moment (the standard deviation) to construct error bars. All the standard options for the Mathematica ListPlot[] functions, such as whether to join points with a line or not, are also available.

The plot[] function is a one-liner for quick plotting, tabling, and browsing of data. It performs listPlot[], table[], and browse[] on an array, in that order. Dimension names specified by the user determine what dimensions are plotted, tabled, and browsed. The plot[] function outputs an interactive browsing pane showing a table of plots for the array position given by the sliders.

3.3.4 Comparison with other data processing tools

Table 3.1 shows a comparison of DART with other tools often used in the processing of scientific data. Table 3.1 compares array processing solutions based on whether they provide the features that have been necessary or could foreseeably be necessary in the first applications of DART. Many of these array processing solutions possess features not listed in this table. SciDB, for example, was designed to process extremely large data sets generated

by astronomical sky surveys, and possesses features not listed here that assist in processing such data sets. DART wasn't designed to handle arrays of this size, but its capabilities are sufficient for many scientific data sets.

DART does not currently include two potentially useful features that are provided by SciDB. It does not have the ability to store fine grained tracking of data revision history, and it does not provide in-situ access to pieces of very large arrays. These disadvantages could be overcome by reworking DART to use a data storage technology that possesses these features. As noted above, however, for most applications computational reproducibility is more effectively supported by modifying data within scripts.

DART also does not have a point and click GUI interface, but a GUI wizard for editing script commands could be built. Parallel processing of arrays may be supported in future versions by using the parallelization features of Mathematica.

DART currently requires the Mathematica shell to run, but it could easily be rewritten to be independent of Mathematica and usable from a great variety of platforms.

Table 3.1: A comparison of the capabilities of various array processing solutions.

	Ordinary arrays	RDBMS and SQL	HDF5 and libraries	Spreadsheets	AML	Plotting software[a]	Math software	SciDB	DART
Advanced array processing									
Storage of arbitrarily high dimensional arrays	●	●	●		●		●	●	●
Built-in interop. with most numerical libraries	●				●		●	●	●
Merging arrays					●			●	●
Nesting arrays	●		●					●	●
Comprehensive plotting				●		●	●		●
Comprehensive exporting and importing				●		●	●		●
Storage of dimension units and position labels		○	●	○		●		●	●
Dimensions referred to by name		●	●			●		●	●
Declarative array manipulation				●				●	●
Automatic use of dimensional information						●			●
Automatic propagation of statistical error						●		●	●
Simplified multidimensional threading					●		○	●	●
Point and click interface		●		●		●			
Support for reproducible research									
High level scripting		●		●		●	●	●	●
Storage of data revision history								●	
Literate programming interface				○			●		●
Open source code	●	●	●	○	●	○		●	○
Performance									
Compact storage of regular arrays	●		●	●	●	●	●	●	●
Fast access to subarrays	●		●		●	●	●	●	●
Parallelism	○	●	●				●		○
In-situ access to parts of very large arrays			●					●	

[a]Eg: Origin, SigmaPlot, QtiPlot, LabPlot

3.4 Summary

This chapter has presented DART, a language for declarative processing of arrays. Various novel mechanisms in DART grant it a high level of syntactical parsimony. Scripts tend to be lists of one-liners, and are about as close to natural language as possible given the subject matter. It includes the full range of capabilities required for complex array analysis, including functions for importing, collating, and regularizing data, reshaping arrays, threading operations over subarrays, signal processing, performing arithmetic with arrays, plotting data, constructing tables, and interactively browsing arrays. Implementation within Mathematica allows immediate access to high level mathematics, and the literate programming features of Mathematica support computational reproducibility. DART will be useful for analysis of data in any field that requires complex processing of multidimensional arrays, such as finance, astronomy, physics, geography, geology, pharmaceutical formulation, behavioral sciences or government.

Bibliography

[1] N.R. Maddux et al. "Multidimensional methods for the formulation of biopharmaceuticals and vaccines". In: *J. Pharm. Sci.* 100.10 (2011), pp. 4171–4197.

[2] A.P. Marathe and K. Salem. "A language for manipulating arrays". In: *Proc. of VLDB 1997 Conference* (1997).

[3] K.A. Baggerly, K.R. Coombes, and J.S. Morris. "An introduction to high-throughput bioinformatics data". In: (2006).

[4] P.G. Brown. "Overview of SciDB: large scale array storage, processing and analysis". In: *Proceedings of the 2010 international conference on Management of data*. ACM. 2010, pp. 963–968.

[5] R. Gentleman and D. Temple Lang. "Statistical analyses and reproducible research". In: *Bioconductor Project Working Papers* (2004), p. 2.

[6] R.D. Peng, F. Dominici, and S.L. Zeger. "Reproducible epidemiologic research". In: *American Journal of Epidemiology* 163.9 (2006), pp. 783–789.

[7] K.A. Baggerly and K.R. Coombes. "Deriving chemosensitivity from cell lines: Forensic bioinformatics and reproducible research in high-throughput biology". In: *The Annals of Applied Statistics* 3.4 (2009), pp. 1309–1334.

[8] S. Fomel and J.F. Claerbout. "Guest Editors' Introduction: Reproducible Research". In: *Computing in Science & Engineering* 11.1 (2009), pp. 5–7.

[9] V. Stodden. "Enabling reproducible research: Open licensing for scientific innovation". In: *International Journal of Communications Law and Policy, Forthcoming* (2009).

[10] P. Vandewalle, J. Kovacevic, and M. Vetterli. "Reproducible research in signal processing". In: *Signal Processing Magazine, IEEE* 26.3 (2009), pp. 37–47.

[11] D.L. Donoho. "An invitation to reproducible computational research". In: *Biostatistics* 11.3 (2010), pp. 385–388.

[12] D.E. Knuth. "Literate programming". In: *Comput. J.* 27(2) (1984), pp. 97–111.

Chapter 4

High throughput generation of Empirical Phase Diagrams

4.1 Introduction

The time and effort required to generate empirical phase diagrams for individual proteins have somewhat limited the method's general applicability. It typically takes several days or up to 1-2 weeks to collect individual data sets with multiple biophysical techniques, process the large combined data set, and generate an EPD. Reducing the total time to a day or less would allow EPDs to be more routinely used during formulation development as a tool for excipient screening and to enable more reliable comparisons of the stability of higher-order macromolecular states.

Multimode instruments have recently become available that could significantly reduce the time to collect the experimental data used to generate an EPD. The Olis Multiscan (Bogart, GA) (also referred to as "The Protein Machine") is a cuvette-based spectrophotometer that measures circular dichroism, UV-absorbance, fluorescence, turbidity, and light scattering with high level photometric and wavelength accuracy and repeatability. In contrast to most plate readers, this instrument measures full spectra, in addition to incorporating a

temperature controlled sample chamber capable of performing temperature studies from 0 to 100°C.

In this chapter we describe the Olis Multiscan (OM) and its use to generate EPDs for four model proteins including aldolase, bovine serum albumin (BSA), a-chymotrypsin, and lysozyme. These proteins cover a range of molecular weights (14 to 160 kDa), secondary structures (10-67% alpha helical and 10-49% beta sheet), and thermal stabilities (Tm values from 44 to 74°C).[1–6] These model proteins were characterized over a grid of environmental conditions consisting of solution pH values from 3 to 8 and temperatures from 10°C to 85°C. The characterization of each protein was performed over a 12 hour period. At each combination of temperature and pH, the following biophysical measurements were taken: CD at 217, 222, and 235 nm, absorbance from 238 to 343 nm (including optical density measurements from 320 to 340 nm), and intrinsic Trp fluorescence between 255 and 420 nm with 295 nm excitation. DART (described in the previous chapter) was then used to import and regularize the data, filter it, and generate EPDs. The resulting empirical phase diagrams have been interpreted in light of the original raw data, and the phase boundaries and protein behavior were found to be reproducible and similar to those obtained by independent measurements using separate instruments.

4.2 Methods

4.2.1 Materials

Albumin (from bovine serum), aldolase (from rabbit muscle), α-chymotrypsin (from bovine pancreas), and lysozyme (from chicken egg white) were obtained in the form of lyophilized powder from Sigma Life Sciences (St. Louis, MO). All chemicals were of reagent grade and purchased from Fisher Scientific (Pittsburg, PA).

Citrate-phosphate buffer was prepared at 20 mM at pH 3.0, 4.0, 5.0, 6.0, 7.0, and 8.0 from citric acid anhydrous and sodium phosphate dibasic anhydrous. The ionic strength of

each buffer was controlled to I = 0.15 (dimensionless) by the addition of NaCl. For each pH, the lyophilized protein samples were dissolved into 2 mL of H2O and all protein solutions were dialyzed into between 1 and 2 L of citrate-phosphate buffer in Thermo Scientific Slide-a-Lyzer 0.5-3 mL 3500 Da MWCO dialysis cassettes (Waltham, MA). The concentration of each sample was obtained by absorbance spectroscopy with 1 cm path length at 280 nm using an Agilent Technologies 8453 spectrophotometer (Santa Clara, CA) and known extinction coefficients for each model protein. Samples were diluted to 0.2 mg/mL. The citrate-phosphate buffers used for protein dilution and as instrument controls were filtered with a Millipore Millex 0.45 μm syringe filter (Billerica, MA). Samples were stored at 4oC, and measurements were taken within two weeks of reconstituting the lyophilized protein powders with the citrate-phosphate buffers (except for lysozyme which was used within 3 weeks).

4.2.2 High Throughput Spectroscopy

High throughput spectroscopy was performed with an OM equipped with a Quantum Northwest peltier temperature controlled 6-position cuvette turret (Liberty Lake, WA). The Olis Multiscan uses a 150 W xenon arc lamp and dual grating Rapid Scanning Monochromators (RSM-1000) for circular dichroism (CD) and fluorescence excitation (see Figure 4.1). The RSM was set up in slow scanning mode, using a fixed 1.24 mm slit (corresponding to 1.6 nm band pass) and moveable gratings. (For rapid scanning, the gratings would be fixed and a scan disk would be used instead of the fixed slit.) Fluorescence signals are read at 90 degrees from the excitation source through a single grating RSM, using a rotating 1 mm slit (corresponding to a 6.3 nm band pass). The fluorescence monochromator collects a spectrum every 10 ms, and can measure an emission spectrum at each CD excitation wavelength. Absorbance is measured using a separate Avantes system (Eerbeek, The Netherlands) consisting of an Avalamp-DS deuterium light source and an AvaSpec-1024 photodiode array spectrometer that is built directly into the spectrometer system.

Samples were placed in 6Q Spectrosil cuvettes with a 2 mm path length in one direction and 1 cm path length in the other. The cuvettes were positioned so that the 2 mm path was parallel to the excitation beam and Avalamp absorbance beam (see Figure 4.1). Thermal stress was performed from 10-85oC in steps of 2.5oC. Excitation monochromator exit slit width and emission monochromator entrance slit width were both maximized to provide robust signals. These slits control light throughput and have no effect on band passes. Integration times were chosen to yield a cycle of 12 hours per experiment, for the convenience of experimenters. In those 12 hours, equal time was allotted to each of the 3 techniques.

The OM was operated in the following manner during temperature ramp experiments. After temperature equilibration of the sample chamber, the 6-position turret rotates to place one of the sample containing cuvettes into the excitation beam path. For each cuvette, the excitation monochromator scans through a series of excitation wavelengths, spending a user specified time at each wavelength while collecting one CD measurement and a full fluorescence spectrum. After scanning the excitation wavelengths, the next cuvette is rotated into the excitation beam, and this process repeats until each cuvette has been measured. The cuvettes are then scanned again, but this time they are placed into the beam path of the Avantes absorbance subsystem and UV absorbance spectra are measured. Finally, the temperature is raised to the next set point and the entire cycle is repeated. One might ask whether the order of the operations could change the outcome, since the various measurements occur at different time delays after the temperature is increased. For example, CD and fluorescence are measured for the first cuvette shortly after the temperature is raised, whereas the absorbance spectrum of the last cuvette is measured roughly 20 minutes later. At all times, however, each cuvette has had ample time to equilibrate to the temperature before the current one. So the measured conformational state of the protein is somewhere between the equilibrated states at the last temperature and the current temperature. Thus, the order of operations can potentially affect the protein's observed state and transition temperatures, but the effect is limited to an uncertainty equal to the size of the temperature

step.

4.2.3 Circular Dichroism

Circular dichroism (CD) was used to measure molar ellipticity at the wavelengths of 217, 222, and 235 nm, chosen to correspond to typical peaks seen for alpha-helix and beta-structure secondary structures.[7,8] Although Chymotrypsin has no CD peak at 222 nm, Chymotrypsin melting trends were similar for all 3 of the wavelengths chosen. Due to absorbance by the 20 mM citrate-phosphate buffer, far UV CD signals could not be monitored successfully at wavelengths below 215 nm for the majority of protein and pH combinations. Although shorter path lengths would have reduced absorbance by the buffer, they would result in a proportionally diminished fluorescence emission signal. Furthermore, the fluorescence emission measurements are taken parallel to the 1cm cuvette path length, making it difficult to use very short excitation beam path lengths due to reflection and alignment issues. To decrease noise in the data resulting from increased absorbance, CD data was integrated for 19 seconds at each wavelength.

4.2.4 Steady State Intrinsic Trp Fluorescence

The tertiary structure of all proteins was screened using intrinsic fluorescence. Proteins were excited at 295 nm to exclusively (>95%) excite tryptophan residues, as well as 300 nm to investigate red-edge shifts. Emission spectra were recorded between 255 and 420 nm with 18 seconds of integration time per excitation wavelength.

Each (temperature x wavelength) spectral melt matrix was filtered as follows. The matrix was first reconstructed using the top 5 singular vectors given by the Singular Value Decomposition. Each spectrum was then filtered with a third order, 4 nm radius Savitzky-Golay filter followed by a 4 nm radius Gaussian smooth. Then the temperature dimension of the matrix (i.e., the melt at each wavelength) was filtered with a third order, 6°C radius Savitzky-Golay filter followed by a 2.5°C radius Gaussian smooth.

Peak positions were determined from the filtered data using the spectral center of mass method. A red edge shift at each temperature and pH was determined by subtracting the emission peak position with 295 nm excitation from the peak position with 300 nm excitation.

4.2.5 Absorbance and Optical Density Measurements

Absorbance spectra were recorded between 238 and 343 nm with 46 seconds of integration time. The long integration time was required due to the low concentration and path length that were required for compatibility with far-UV CD measurements. The peak near-UV absorbance of BSA for instance, was approximately 0.029 absorbance units. Each (temperature x wavelength) spectral melt matrix was filtered in the same manner as the fluorescence measurements. Second derivative spectra were then calculated with a third order, 4 nm radius, second derivative Savitzky-Golay filter. The mean optical density from 320 to 340 nm was calculated from unfiltered spectra by averaging the optical density values in that range.

4.2.6 Construction of EPDs

EPDs were constructed from each run and were generated separately for each of the four model proteins. The EPDs resulting from the first 12 hour run of each protein are shown in Figure 3. The EPDs from further runs are virtually identical (data not shown). The following biophysical measurements were used: the second derivative near UV absorbance from 275 to 295 nm, mean optical density from 320 to 340 nm, far UV CD at 217, 222, and 235 nm, and fluorescence spectra from 315 to 370 nm. It should be noted that full UV absorbance and fluorescence spectra were used rather than selected peak positions and intensities. Before applying the EPD method, the data was interpolated in the temperature dimension from the original 2.5°C increments to smaller 0.5°C increments. For a detailed description of the EPD method, see Maddux et al.[9]

4.2.7 EPD segmentation

The EPD method assists in the determination of regions of conserved behavior, but does not itself determine transition regions. Instead, human visual assessments have traditionally been used to perform the task of separating regions of the same color on an EPD. A similar classification can also be performed mathematically by the use of cluster analysis. Since it is automatic, cluster analysis may be a valuable tool for high-throughput protein stability characterization using EPDs.

Cluster analysis was performed on the average of 3 runs, separately for each protein, using the same combination of measurements used to generate EPDs. The standardization step in the EPD method results in a list of multidimensional vectors, with one vector for each combination of temperature and pH. K-means clustering was applied to these vectors to find a natural categorization in the high dimensional space, thus dividing the temperature-pH plane into regions of similar measurements. For a description of clustering in general and the K-means algorithm in particular, see Jain 2006.[10]

The phases and transitions temperatures given by K-means clustering did not, however, always match the phases and transitions perceived by visual assessments of the EPDs. To address this issue, the EPDs were segmented using a different method. For each phase in an EPD, a characteristic point in the temperature-pH plane was visually selected to represent that phase. Then for each point in the EPD the nearest characteristic point was determined, where the distance utilized was the Euclidean distance between measurement vectors. The boundaries in the resulting segmented EPD were used to find transition temperatures by averaging the temperature above and below a boundary. These were then averaged over the 3 runs to determine the transition temperatures given below. The error in a transition temperature was calculated by adding (in quadrature) the standard deviation over 3 runs and the quantization error of 2.5°C.

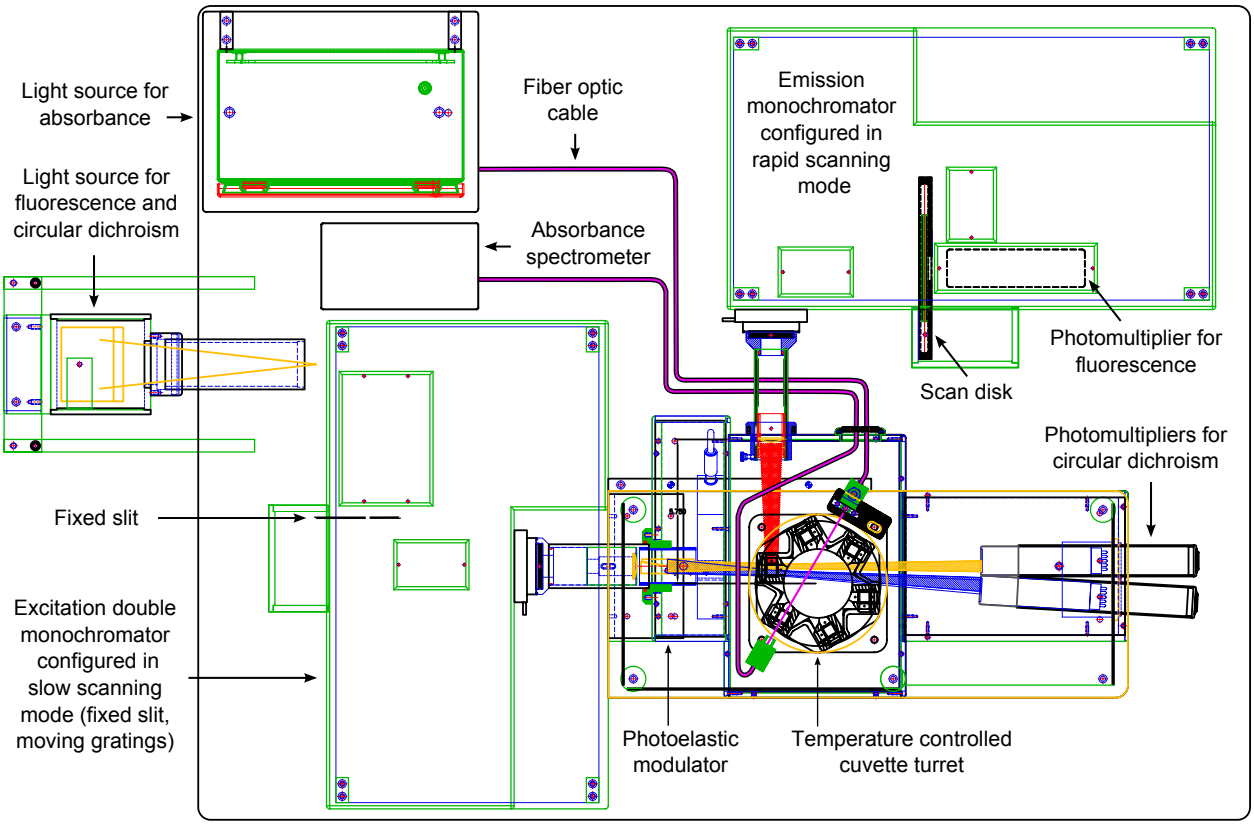


Figure 4.1: Overhead line drawing of the Olis Multiscan (OM), a cuvette-based spectrophotometer that measures circular dichroism, absorbance, and fluorescence with high level photometric and wavelength accuracy and repeatability.

4.3 Results and Discussion

Each EPD (containing data from triplicate runs) required approximately only 1.8 mg of protein, about 8 hours of sample preparation time, about 72 hours of instrument time, and minutes of data analysis time. (The 1.8 mg of protein is the amount used in 3 spectroscopic runs, not including planned overages. In practice the amount of protein required may vary, depending on the cost of the analyte and skill of the experimentalist.) The instrument time could be further reduced since data was collected in triplicate and buffers were measured for every pH and run. The buffer spectra could be measured for only one pH condition using one of the six turret positions. With this experimental setup, an EPD across 5 pH values could be generated using all 3 experimental techniques with only about 12 hours of instrument time and about 0.5 mg of protein. If CD measurements were not used as part of the EPD analysis, the protein concentration and cuvette path length could be increased and instrument integration times could be decreased, allowing perhaps up to 4 EPDs to be generated per day.

If separate instruments had been used, the data collection run described in this chapter would have required 3 times as much protein, because new samples would be needed for each instrument (absorbance, fluorescence, and circular dichroism spectrophotometers). Preparation time would also increase due to the need to prepare more samples and cuvettes for multiple experimental runs. Total instrument time, however, would not increase due to the use of separate instruments, since the OM collects data from the 3 experimental measurements separately. (I.e., the Far UV CD measurements use wavelengths in the 200 to 240 nm range, the fluorescence measurements use excitation wavelengths of 295 and 300 nm, and the absorbance measurements are not simultaneous with CD or fluorescence.) In fact, total instrument time would probably be reduced somewhat by using separate instruments, because path lengths and concentrations could be optimized separately for each instrument.

Representative examples of measurements as a function of pH and temperature from each of the biophysical measurements for the four model proteins are shown in Figure 2. These

results are averaged over 3 runs as a function of temperature and solution pH. For error bars associated with these measurements, see Figures 4.4 - 4.7. Figure 3 shows EPDs summarizing the biophysical data presented in Figure 2. Note that the EPD's were generated using data collected from all wavelengths, not just the representative wavelengths shown in Figure 2. As explained elsewhere, in an EPD the colors themselves have no absolute meaning.[9] Instead, differences in color alone assist in the determination of protein behavior and behavioral transitions.

The EPD for aldolase displays 4 structural states as a function of pH and temperature stress (Figure 3a). At pH 3 and pH 4, the protein was unfolded, as shown by a lack of transitions (color changes) in the EPD and by inspection of the biophysical data itself. At pH 5 to 8, the protein manifested structural changes near 50°C. This apparent phase change is characterized by the melting of secondary and tertiary structure as shown by transitions in absorbance, CD and fluorescence.[7,8,11–17] A few degrees after the onset of melting, the protein self-associated, as shown by increases in optical density. The protein initiated structural changes again at pH values 5 to 8 near 60°C. Although this phase may be partially unfolded, some structure remains, as indicated by the slight recovery in the red shift between 60°C and 80°C.[18,19] The EPD in Figure 3a for aldolase is lacking a structural transition near 30°C at pH 3 and 4, which was found by Hu et al.[20] This difference is probably due to the fact that the EPD in this work does not include near-UV CD data.

The EPD for BSA also displays multiple structural states as a function of pH and temperature stress (Figure 3b). At pH 3, BSA is partially unfolded by the acidic conditions, as indicated by the weak transitions visible in both the EPD and biophysical data. At pH 4, the same phenomenon is observed, though to a lesser extent. The protein had the highest melting temperature at pH 6, with a structural transition observed in the EPD near 65.6 $\pm$ 2.6°C. From pH 5 to 8, the first transition is characterized by melting of the secondary structure as indicated by CD, melting of the tertiary structure indicated by fluorescence changes, and protein association reflected by a slight blue shift in the fluorescence peak position and a

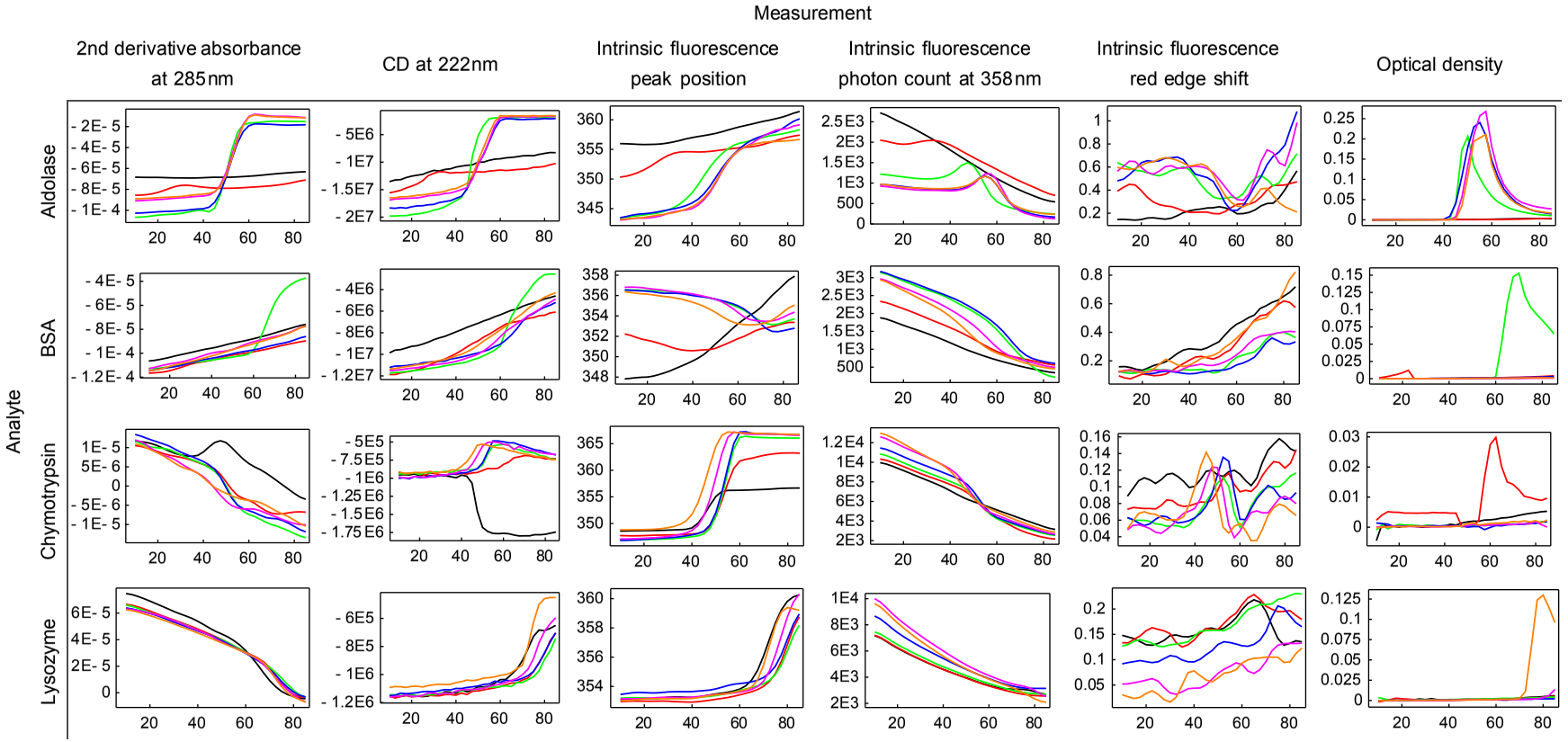


Figure 4.2: Representative biophysical measurements of 4 model proteins collected with the OM. The data includes second derivative near UV absorbance, circular dichroism, intrinsic Trp fluorescence and optical density measurements for the model proteins aldolase, BSA, chymotrypsin, and lysozyme. For error bars, see the supplemental figures. The units for the Y-axis of each column are, respectively: absorbance unit/nm2 , °cm2/dmol, nm, number of photons, emission peak shift per excitation wavelength change (nm/nm), and optical density units. The line colors are black-pH 3, red-pH 4, green-pH 5, blue-pH 6, purple-pH 7, and orange-pH 8. Each figure has temperature (°C) as the horizontal axis.

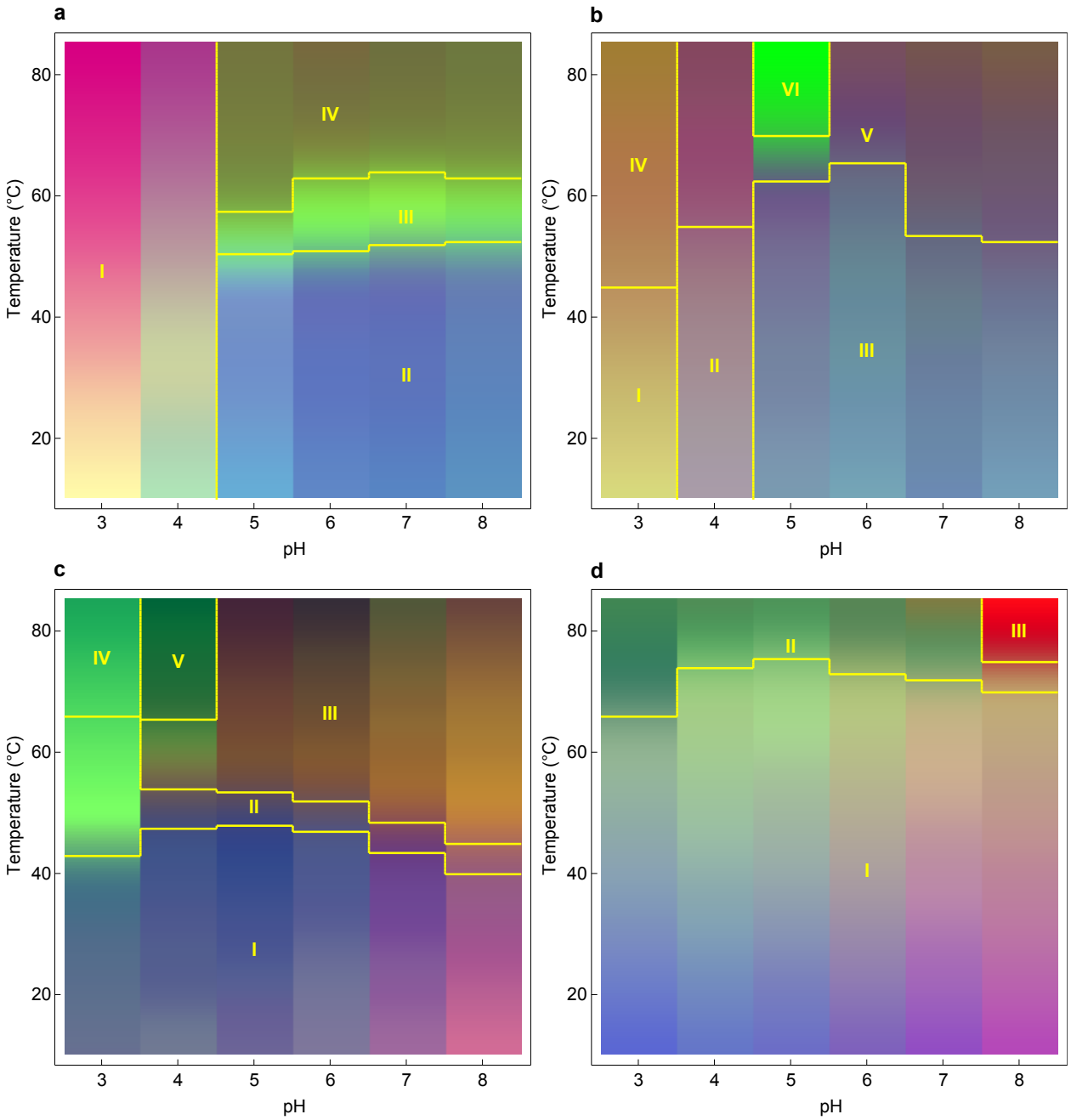


Figure 4.3: Empirical phase diagrams of 4 model proteins: (a) aldolase, (b) BSA, (c) chymotrypsin, and (d) lysozyme. These EPDs summarize the representative biophysical data from Figure 2 (in conjunction with wavelength measurements using the full spectra; see text) and display protein structural responses to temperature and pH perturbations. The experimental techniques used were second derivative near UV absorbance from 275 to 295 nm (full spectra), fluorescence spectra from 315 to 370 nm (full spectra), far UV CD at 217, 222, and 235 nm, and the mean optical density from 320 to 340 nm.

rising red edge shift. At pH 5 the protein undergoes a transition near 68.9 ± 2.6°C in which secondary structure melts more completely, as seen in CD measurements, and the protein more strongly self-associates, as indicated by increases in optical density.

The EPD for chymotrypsin also displays multiple structural states as a function of pH and temperature stress (Figure 3c). Unlike aldolase and BSA, however, chymotrypsin exhibits the same structural state across the entire pH range of 3 to 8 at lower temperatures. At pH 3, the protein's tertiary structure begins to change near 35°C, as shown in the UV second derivative absorbance plot. The secondary structure then begins to change to a more helical state with an onset near 42.8 ± 2.5°C, as suggested by the CD measurements and the EPD. Although chymotrypsin does not display a CD peak at the wavelength of 222 nm used for the plots in Figure 2, the melting trends are the same at all CD wavelengths monitored (data not shown). At pH 4 to 8, the protein's secondary and tertiary structure alter simultaneously near 42.5°C, as shown by CD and fluorescence peak position measurements. During the transition, the red edge shift increases temporarily, consistent with protein association. At pH 4 near 65.6 ± 2.9°C, the protein changes to a state characterized by strong association behavior, as indicated by OD measurements. The color variations visible at lower temperatures (20 to 40°C) result from noise in the data, as they were different for each run. (EPDs for each run are not shown.)

The EPD for lysozyme (Figure 3d), exhibits a single apparent phase from pH 3-8 and 10-60°C. The protein is most stable at pH 4 to 6, undergoing a transition in secondary and tertiary structure near 73°C. The color variations visible at lower temperatures (20 to 60°C) result from a constant slope in the data, visible in Figure 2. At pH 8, the protein begins to strongly self-associate with an onset near 75 ± 2.7°C, as indicated by OD measurements. Analysis of the same four model proteins was conducted by measurements in separate biophysical instruments and virtually identical results were obtained.

4.4 Conclusion

A combined multifunctional spectrometer (the Olis Multiscan) is capable of collecting a variety of protein biophysical data with a single instrument for the construction of EPDs at a much higher throughput than has previously been possible, while maintaining reproducibility and good signal to noise levels. A newly developed software analysis package was combined with the OM instrument to rapidly produce EPDs in a high throughput fashion with minimal sample requirements. The utility of this new methodology was demonstrated by evaluating the conformational stability of four model proteins as a function of solution pH and temperature. The major result of this work is the direct demonstration of the small amount of protein sample needed and short period of time required to generate high resolution EPDs for biophysical characterization of protein conformational stability as a function of solution pH and temperature.

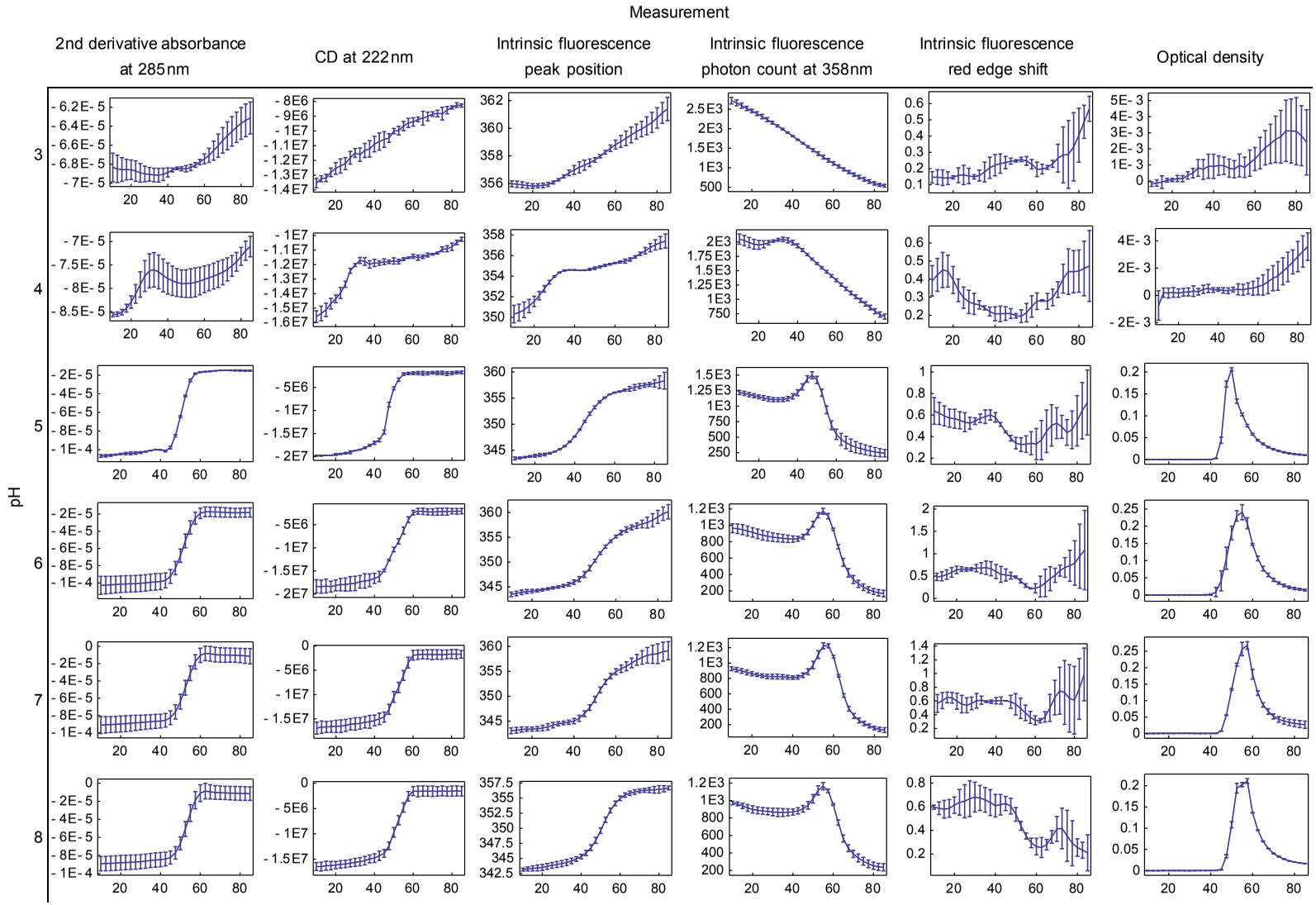


Figure 4.4: Error bars associated with biophysical measurements of Aldolase collected with the OM. The units are for each column are, respectively: absorbance unit/nm2 , °cm2/dmol, nm, number of photons, emission peak shift per excitation wavelength change (nm/nm), and optical density units. Each figure has temperature (°C) as the horizontal axis.

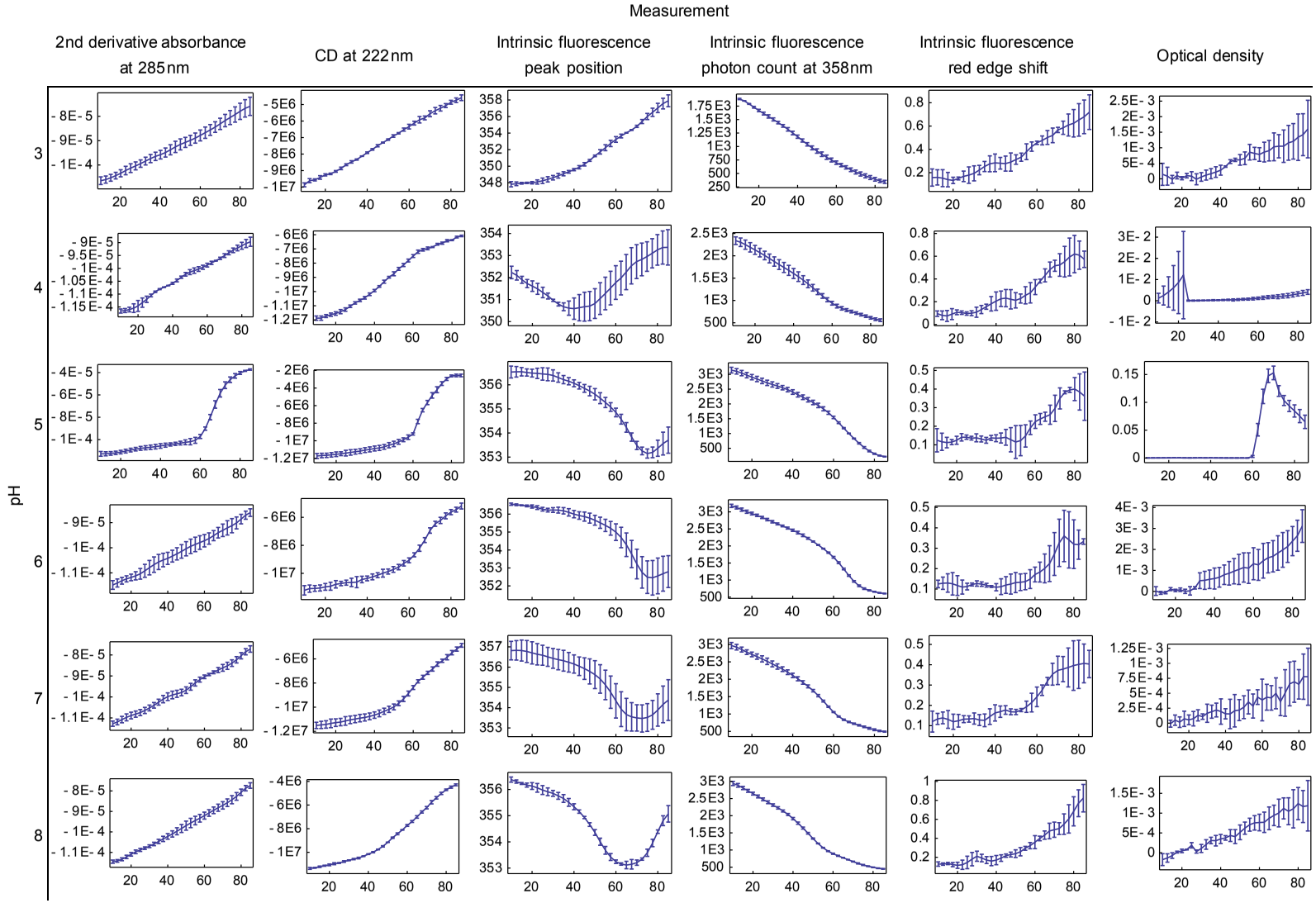


Figure 4.5: Error bars associated with biophysical measurements of BSA collected with the OM. The units are for each column are, respectively: absorbance unit/nm2 , °cm2/dmol, nm, number of photons, emission peak shift per excitation wavelength change (nm/nm), and optical density units. Each figure has temperature (°C) as the horizontal axis.

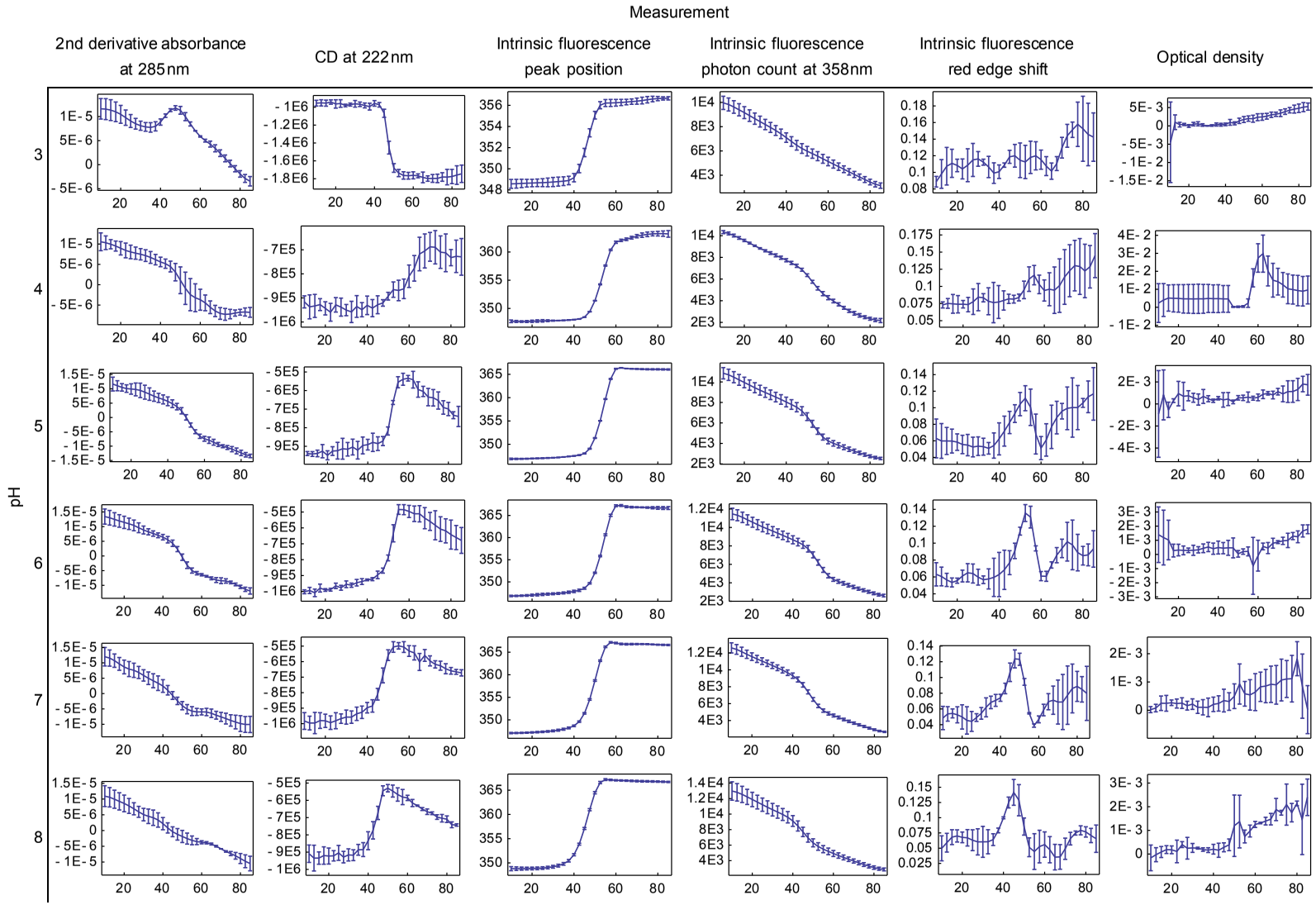


Figure 4.6: Error bars associated with biophysical measurements of Chymotrypsin collected with the OM. The units are for each column are, respectively: absorbance unit/nm2 , °cm2/dmol, nm, number of photons, emission peak shift per excitation wavelength change (nm/nm), and optical density units. Each figure has temperature (°C) as the horizontal axis.

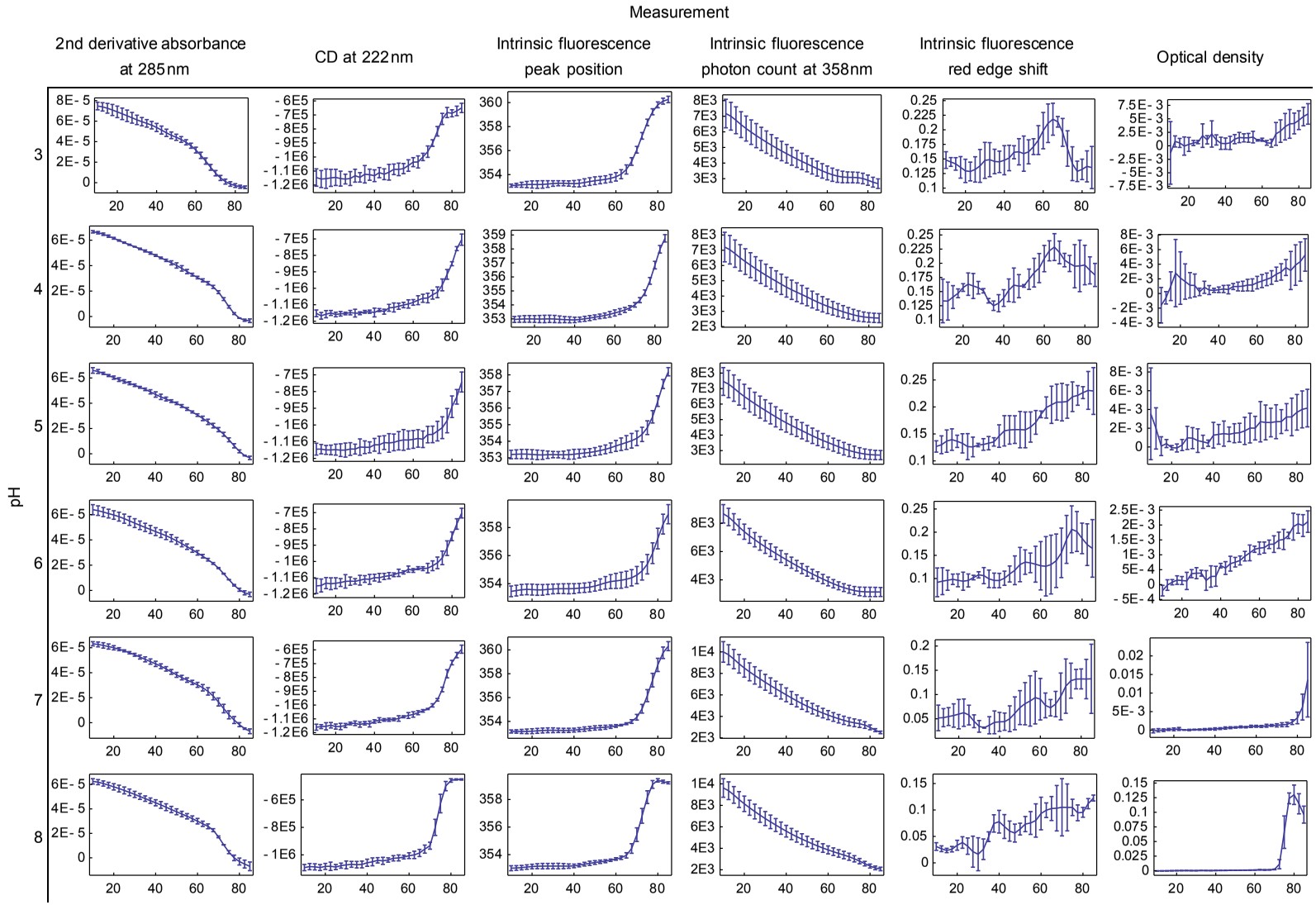


Figure 4.7: Error bars associated with biophysical measurements of Lysozyme collected with the OM. The units are for each column are, respectively: absorbance unit/nm2 , °cm2/dmol, nm, number of photons, emission peak shift per excitation wavelength change (nm/nm), and optical density units. Each figure has temperature (°C) as the horizontal axis.

Bibliography

[1] P.T. Beernink and D.R. Tolan. "Subunit interface mutants of rabbit muscle aldolase form active dimers". In: *Protein Science* 3.9 (1994), pp. 1383–1391.

[2] D.M. Byler and H. Susi. "Examination of the secondary structure of proteins by deconvolved FTIR spectra". In: *Biopolymers* 25.3 (1986), pp. 469–487.

[3] Y. Liu and J.M. Sturtevant. "The observed change in heat capacity accompanying the thermal unfolding of proteins depends on the composition of the solution and on the method employed to change the temperature of unfolding". In: *Biochemistry* 35.9 (1996), pp. 3059–3062.

[4] P. Lozano, T. Diego, and J.L. Iborra. "Dynamic Structure/Function Relationships in the alpha Chymotrypsin Deactivation Process by Heat and pH". In: *European Journal of Biochemistry* 248.1 (1997), pp. 80–85.

[5] K. Murayama and M. Tomida. "Heat-induced secondary structure and conformation change of bovine serum albumin investigated by Fourier transform infrared spectroscopy". In: *Biochemistry* 43.36 (2004), pp. 11526–11532.

[6] L. Sawyer, L.A. Fothergill-Gilmore, and P.S. Freemont. "The predicted secondary structures of class I fructose-bisphosphate aldolases". In: *Biochemical Journal* 249.3 (1988), p. 789.

[7] S.M. Kelly and N.C. Price. "The application of circular dichroism to studies of protein folding and unfolding". In: *Biochimica Et Biophysica Acta-protein structure and molecular enzymology* 1338.2 (1997), pp. 161–185.

[8] S Y Venyaminov and J T Yang. "Determination of protein secondary structure". In: *Circular dichroism and the conformational analysis of biomolecules*. Ed. by G D Fasman. New York: Plenum Press, 1996, pp. 69–107.

[9] N.R. Maddux et al. "Multidimensional methods for the formulation of biopharmaceuticals and vaccines". In: *J. Pharm. Sci.* 100.10 (2011), pp. 4171–4197.

[10] A.K. Jain. "Data clustering: 50 years beyond K-means". In: *Pattern Recognition Letters* 31.8 (2010), pp. 651–666.

[11] LISA A Kueltzo and C R Middaugh. "Ultraviolet absorption spectroscopy". In: *Methods for structural analysis of protein pharmaceuticals* 3 (2005), pp. 1–25.

[12] H. Mach et al. "Ultraviolet absorption spectroscopy". In: *Methods Mol Biol* 40 (1995), pp. 91–114.

[13] H. Mach et al. "Examination of phenylalanine microenvironments in proteins by second-derivative absorption spectroscopy". In: *Archives of biochemistry and biophysics* 287.1 (1991), pp. 33–40.

[14] K Nakanishi, N Berova, and R W Woody. *Circular dichroism - principles and applications*. New York: VCH Publishers Inc., 1994.

[15] N Sreerama et al. "Tyrosine, phenylalanine, and disulfide contributions to the circular dichroism of proteins: Circular dichroism spectra of wild-type and mutant bovine pancreatic trypsin inhibitor". In: *Biochemistry* 38 (1999), pp. 10814–22.

[16] W Jiskoot et al. "Fluorescence Spectroscopy". In: *Methods for structural analysis of protein pharmaceuticals*. Ed. by W Jiskoot and D. Crommelin. Arlington, VA: AAPS Press, 2005, pp. 27–82.

[17] J.R. Lakowicz and B.R. Masters. "Principles of fluorescence spectroscopy". In: *Journal of Biomedical Optics* 13 (2008), p. 029901.

[18] AP Demchenko. "Red-edge-excitation fluorescence spectroscopy of single-tryptophan proteins". In: *European Biophysics Journal* 16.2 (1988), pp. 121–129.

[19] A P Demchenko. "The red-edge effects: 30 years of exploration". In: *Luminescence* 17 (2002), pp. 19–42.

[20] L. Hu et al. "Investigation of Protein Conformational Stability Employing a Multimodal Spectrometer". In: *Analytical Chemistry* 83.24 (2011), pp. 9399–9405.

Chapter 5

Modeling of long term GCSF stability from short term physical data

In this chapter we establish the long-sought connection between the long-term pharmaceutical stability of a protein and short-term data characterizing the protein's physical form. The novelty of our approach comes in using mathematical methods well-suited to bring these broad themes together. Our proof-of-principle analysis will show that features of long-term pharmaceutical stability can indeed be predicted from short-term measurements.

Before explaining the prediction process, we need to review some background facts about the field as it stands now.

5.1 The Expensive Process of Developing Useful Drugs

The development process for new drugs uses experimental methods covering a wide range in cost, from techniques that are expensive, time consuming and difficult to perform, such as those that assess drug safety, efficacy, and long term stability, to techniques that are inexpensive, fast, and easy to perform, such as those that primarily use spectroscopic instruments.

Slower, more expensive techniques typically produce information that is more directly relevant to the optimization of drug designs, formulations, and manufacturing processes. Determining the long term stability of protein drugs is an example of a slow, laborious, and

expensive drug development process. Protein stabilization studies are typically performed with a factorial design of experiments. In this mode, one chooses several variables to explore (these are called factors), such as the concentrations of various excipients and a number of levels for each factor. For instance, a factorial design with 4 factors at 3 levels each results in $3^4 = 81$ formulations to test. One stores these formulations in clinically relevant storage conditions (for example, 2 years at 4°C). At the end of the storage period the formulations are tested to determine the amount of active component remaining, the presence of degraded forms, and the biological activity of the altered drug. One then fits a polynomial model that predicts the end-point measurements from the formulation variables.

The information obtained by faster, less expensive techniques is typically only loosely indicative of the differences observed in more costly studies. Examples of fast techniques include immunological assays to study vaccine efficacy, the use of forced degradation to estimate drug stability and the use of circular dichroism (CD) measurements to determine whether two protein variants are sufficiently similar for therapeutic use. Fast, inexpensive methods allow one to explore design spaces more completely and with greater freedom. More rapid experiments are also necessary during the development and maintenance of manufacturing processes: when an issue or a question arises, one can't wait years for an answer.

Spectroscopy and calorimetry are two commonly employed fast, inexpensive method methods of obtaining information concerning a formulation. These methods have found wide use in the evaluation of protein stability due to the rich information they provide regarding the structural state of proteins and their degradation behavior. Transition temperatures determined by dynamic scanning calorimetry have occasionally been found to be roughly predictive of a protein's long term stability.[1] Spectral similarity methods are frequently applied during the development and maintenance of manufacturing processes, to determine whether a protein is sufficiently similar to its native form.

In this dissertation we have been focused on fast multidimensional methods. Multidimensional analyses incorporating data from many spectroscopic instruments are gaining

popularity due to their ability to simultaneously investigate protein behavior at multiple structure levels. An example of a multidimensional analysis method is the empirical phase diagram (EPD) method, which assists in finding regions in formulation space with conserved protein structure and boundaries between those regions.[2–5] Although spectroscopic and calorimetric methods are fast and they assist in determining protein state and behavior, inference of long term behavior and activity has been somewhat speculative.

5.1.1 Combining Two Strengths

Since both fast and slow methods have their own strengths, it would be desirable to develop techniques that combine speed, low cost, and ease of use with reliable assessment of drug safety, efficacy, and long term stability. Here is where our new approach provides the doorway to a combined approach, in which fast and slow degradation assays are performed on the same factorial design of protein formulations. We will describe new automated mathematical models that use inexpensive data to predict more costly data that possesses direct relevance to drug development. Predictive techniques can be applied in many ways, and great opportunity for creativity exists by varying the predictive method, the data being predicted, and the data being predicted from. In this paper we focus on automated modeling of the long term stability of proteins based on short term spectroscopic measurements.

In more detail, prediction techniques based on least squares methods are applied to generate a model that predicts slow degradation data from fast thermal unfolding data. The approach has three main differences from traditional long-term stability studies. First, the new method is fast, once a predictive model has been determined. With the predictive model in hand, one must only collect fast unfolding data to estimate long term stability information. Second, in the new approach one infers behavior from behavior: a protein's long term behavior for a given formulation from its short term behavior. Traditionally one infers long term behavior by interpolating (via a polynomial model) between long term data points. Interpolation is problematic when applied to the inference of long term protein

behavior, because formulation spaces happen to have high dimensionality. Finally, the main difference between the new approach and traditional short term studies is that the new method is predictive and incorporates a consistent backbone of testability. Traditional testing uses transition temperatures as no more than a rough indication of differences in stability between different formulations. That imprecise knowledge is then used to determine the focus and extent of long term studies. In the new approach, a fully validated, predictive model is developed. The scope and errors of the model are quantified. A strong validation method allows one to extract more information from fast techniques, potentially eliminating a portion of the cost (and waste) of long-term studies.

The idea of predicting properties is not new to the pharmaceutical field.[6–11] For example, Quantitative Structure Activity Relationship (QSAR) models are used to predict the biological activity of drugs from physicochemical descriptor variables. In reference [6], the blood-brain concentration ratio and blood-brain barrier permeability of 61 molecules were modeled using partial least squares and topological and constitutive descriptors of the molecules. Linear regression models are used frequently in the formulation development process. In reference [7], linear regression was used to model pharmaceutical tablet disintegration time and crushing strength from powder properties along with process and composition variables. In reference [8], linear regression was used to develop a method for the simultaneous spectroscopic quantification of caffeine, acetylsalicylic acid and acetaminophen in solution.

Since least-squares analysis is a standard and well-established way of quantifying relationships we employ it here as well. The first feature that is new here is the reduction of high dimensional data to empirically meaningful summaries at an early stage. The self-consistent nature of short-term data summaries such as EPD's is one thing: the real proof of principle comes when that short-term synthesis confronts an independent universe of long-term data. In principle the two might have little relation to one another. We will show, however, that remarkable and indisputable regularities do occur. The application we describe uses spectroscopic measurements for the short-term data, which will be compared to the response

variables of long-term stability measurements.

5.2 Materials and Methods

5.2.1 Long Term Stability Studies

This part of the experiment is quickly summarized here. For details see reference [1]. Granulocyte Colony stimulating factor (GCSF) was studied in a factorial experimental design of formulations, alternating Acetate and Citrate buffer systems and varying buffer concentrations, pH values, and concentrations of Tween80 and HP-β-CD. The factorial design included 24 formulations, but the long term measurements for only 16 of those have been provided. Formulations were numbered 11-26. Parameter values for each formulation are shown in Table 5.1.

Shortly after formulation, thermal unfolding transition midpoints (T_m) were obtained by differential scanning calorimetry (DSC) with a 90 K/hr heating rate.

Isothermal stability studies were then performed by storing the formulations at 40 °C for 3 months, 25 °C for 10 months, and 4 °C for 20-24 months.

Loss of monomeric form, primarily via aggregation, was monitored by size exclusion high performance liquid chromatography (SE-HPLC). The formulations were analyzed at 0, 1, 3, 6, 9, 12, and 20-24 months. A monomer loss rate constant (MLRC) was determined for each formulation by fitting exponential decay to the SE-HPLC data, and is expressed in units of %/month.

Turbidity was measured at the end of the storage period, and is expressed in Formazin Nephelometric Units (FNU).

Chemical stability was assessed at the end of the storage period by use of reverse-phase high performance liquid chromatography (RP-HPLC). The chemical stability is expressed as as a percentage: the area under the chromatogram peak corresponding to intact GCSF in comparison to the total area under the chromatogram.

Table 5.1: Formulation parameters for 16 formulations of GCSF in a factorial design of experiment.

Formulation	Buffer	pH	Buffer concentration (mM)	Tween 80 (%)	HP−Beta−CD (%)
11	None	4.5	0.	0.05	0.
12	None	4.5	0.	0.005	0.
13	None	5.	0.	0.005	0.
14	None	5.	0.	0.05	0.
15	Citrate	4.5	20.	0.005	0.
16	Citrate	4.5	50.	0.05	0.
17	Citrate	5.	20.	0.05	0.
18	Citrate	5.	50.	0.005	0.
19	None	4.	0.	0.	5.
20	None	4.	0.	0.	1.
21	None	4.5	0.	0.	1.
22	None	4.5	0.	0.	5.
23	Acetate	4.	20.	0.	1.
24	Acetate	4.	100.	0.	5.
25	Acetate	4.5	20.	0.	5.
26	Acetate	4.5	100.	0.	1.

Particle counts (PC) were obtained at the end of the storage period by automated particle counting.

The long term results for each formulation are shown in Table 5.2.

Differential scanning calorimetry's effectiveness at predicting long term stability measurements was then evaluated. This was performed for each long term storage condition separately. Here we discuss the long term stability of the formulations stored at 4 °C for 20-24 months. Long term measurements were ranked 1-24 by stability. The resulting rank numbers were averaged and the averages were ranked 1 to 24 to generate a single measure of long term stability. The ranking procedure just described was performed in two passes: first on groups of measurements, then repeated to combine the resulting rankings. DSC transition temperatures were also ranked 1 to 24. A correlation plot of the long term stability ranking versus the DSC ranking was shown in paper [1] and is reproduced here in Figure 5.1.

Table 5.2: Long term stability measurements for 16 formulations of GCSF. Measurements are shown with 2 digits of precision. See text for details.

Formulation	Measurement															
	Chemical Stability (%)	PC, 20 mo, >01um	PC, 20 mo, >10um	PC, 20 mo, >25um	MLRC, 20 mo	Turbidity, 20 mo	PC, 10 mo, >01um	PC, 10 mo, >10um	PC, 10 mo, >25um	MLRC, 10 mo	Turbidity, 10 mo	PC, 03 mo, >01um	PC, 03 mo, >10um	PC, 03 mo, >25um	MLRC, 03 mo	Turbidity, 03 mo
11	62	52	2	2	−3E−4	1	99	2	1	−1.6E−2	0.48	56	1	0	−0.1	1.9
12	1E2	49	4	2	−1E−4	0.7	1.6E2	3	1	1E−3	0.55	1.4E2	2	1	−2.1E−2	0.61
13	1E2	1.8E2	3	0	−7E−4	0.6	2.7E2	2	0	−5E−3	0.52	1E3	14	1	−4E−2	1.2
14	74	91	9	1	−1.1E−3	0.8	3.1E2	18	1	−1.5E−2	0.49	3.5E2	6	2	−0.14	1.2
15	1E2	2E4	1.7E2	12	−9E−4	1.6	1.5E4	3.2E2	26	−3E−3	2.9	1.2E4	3.3E2	1.7E2	−0.16	1.8
16	67	4E3	1.7E2	17	−6E−4	1	2E3	3.4E2	94	−2.7E−2	1.3	1.2E4	3.5E3	2E3	−0.72	14
17	74	3E3	12	3	−1.7E−3	0.8	1.6E4	40	3	−1.7E−2	1.3	3.9E2	18	11	−0.61	0.61
18	1E2	1.4E4	2.9E2	12	−1.1E−3	1.2	9.1E3	9.5E2	1E2	−3E−3	2.9	6.7E3	57	2	−0.16	3.7
19	1E2	7E2	6	2	−3.9E−2	0.7	5.7E2	20	7	−1.5E−2	0.65	1.5E3	14	1	−3.8E−2	0.69
20	1E2	8.3E2	8	1	−5E−4	1.1	7.3E2	3	1	−1.1E−2	0.62	1.3E3	4	0	−3.5E−2	1
21	1E2	1E3	13	3	4E−5	0.9	8.6E2	18	2	−3E−3	0.52	1.8E3	12	0	−2.8E−2	0.94
22	1E2	7.9E2	7	0	−3.8E−2	0.7	1.2E3	10	2	−9E−3	0.76	9.9E2	12	0	−3.9E−2	0.82
23	1E2	8.5E2	63	10	−1.1E−3	1.1	3.7E2	10	3	−6E−3	0.51	2.6E2	9	0	−2E−2	1.5
24	1E2	4.9E3	1E2	10	−3.8E−2	2.9	2.6E3	90	10	−1.8E−2	1.2	8.4E3	1.2E2	26	−4.5E−2	1.2
25	1E2	3.2E3	37	8	−4E−2	1.6	4.6E3	47	12	−1.4E−2	1.8	1.6E3	29	7	−3.7E−2	9.9
26	1E2	1.2E4	2.7E2	23	−2.1E−3	5.2	3.5E3	4.4E2	28	−1.4E−2	3.3	4.7E3	93	12	−6.9E−2	1

5.2.2 Accelerated Stability Studies

Sample preparation

GCSF in 16 different formulation conditions was provided by the authors of [1]. These formulations are numbered GCSF011 thru GCSF026. The protein was thawed and diluted using the respective buffer to obtain the required concentration for each technique.

Far-UV Circular Dichroism (CD) Spectroscopy

CD spectra were acquired using a Jasco J-810 spectropolarimeter (Jasco Inc, Easton, MD) equipped with a 6-position peltier temperature controlled sample cell holder. CD spectra were obtained from 260-200 nm. CD spectra were collected every 2.5 °C over a range of 10 to 87.5 °C .

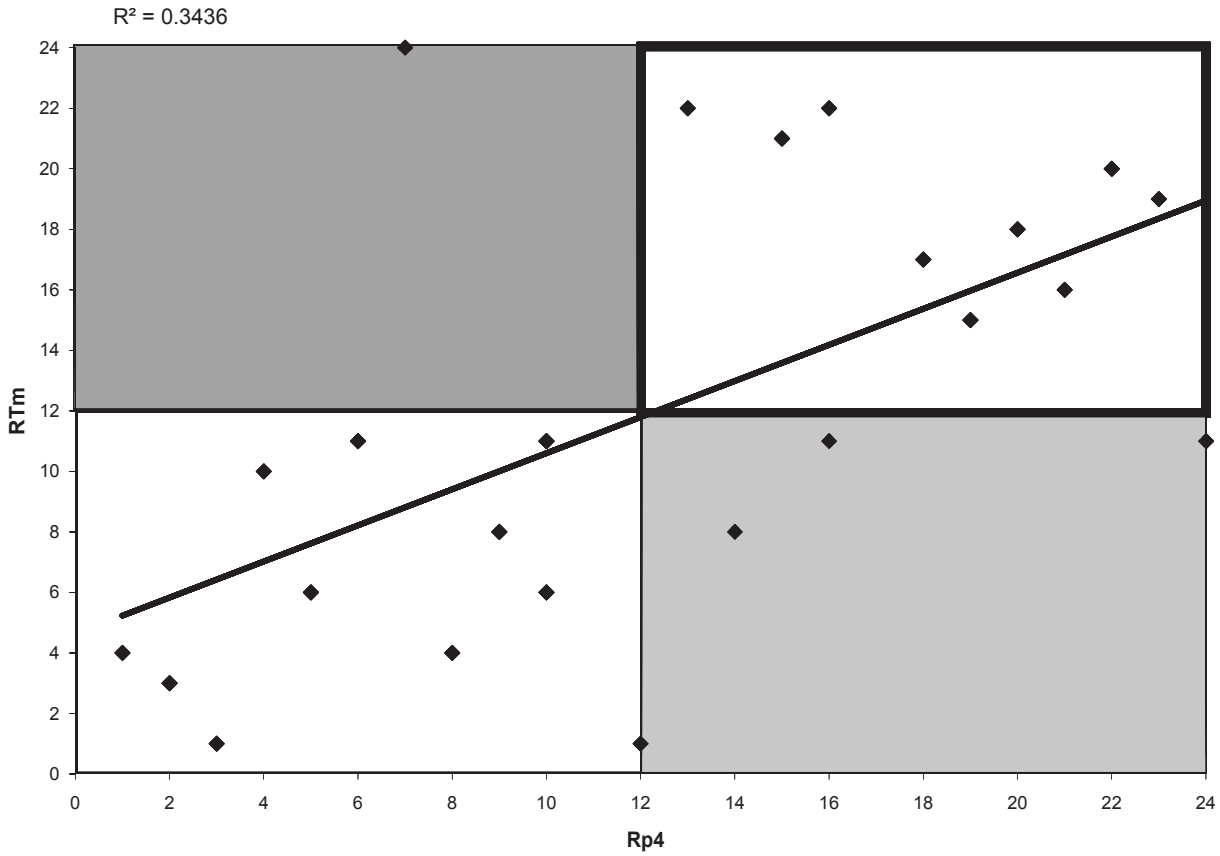


Figure 5.1: Correlation plot of the long term stability of 24 GCSF formulations versus thermal transition midpoints measured by dynamic scanning calorimetry (DSC). The long term stability and DSC measurements of the formulations were each ranked 1 to 24 by a procedure described in the text. The horizontal and vertical axii of the plot are, respectively, the long term stability and DSC rankings.(Reprinted from the European Journal of Pharmaceutics, Ahmed M K Youssef and Gerhard Winter, "A critical evaluation of microcalorimetry as a predictive tool for long term stability of liquid protein formulations: Granulocyte Colony Stimulating Factor (GCSF)", In Press, Copyright 2013, with permission from Elsevier)

Intrinsic Tryptophan (Trp) Fluorescence Spectroscopy and Static light scattering

Intrinsic fluorescence spectra were acquired using a Photon Technology International (PTI) spectrofluorometer (Lawrenceville, NJ) equipped with a turreted 4-position peltier-controlled cell holder. The Trp residues in the protein were excited at 295 nm, and the emission spectra were collected from 310 to 400 nm. Light scattering in the 290-300 nm range was collected simultaneously with fluorescence by use of a second photomultiplier. Fluorescence and light scattering were collected every 2.5 °C over a range of 10 to 87.5 °C .

ANS Fluorescence Spectroscopy

Unfolding of the protein with increasing temperature was also monitored by fluorescence emission of the extrinsic probe 8-Anilino-1-naphthalene sulfonate (ANS). An optimized 15 fold molar excess of ANS was added to the protein solution. The ANS die was excited at 375 nm and the emission spectrum was collected from 400-600 nm every 2.5 °C over the temperature range of 10 to 87.5 °C . Peak intensity of the emission spectra was monitored at 475 nm.

5.3 Analysis

5.3.1 Subtraction of buffer spectra

As shown in Figure 5.2, the buffers for light scattering, intrinsic fluorescence, and ANS fluorescence presented a relatively large spectral signal. In this figure the buffer and sample spectra are plotted for each technique and for formulation 11. Each plot shows the measurement for the technique in question as a function of temperature and wavelength. (The band near 60 °C in ANS fluorescence will be discussed in the subsection on filtering.) A small part of the signal in the buffers is certainly due to the buffer itself. The signal is much larger than what would usually be attributable to buffer alone, however, and may be partly due to a protein contaminate. To avoid introducing this artifact into the sample spectra, subtraction of buffer spectra could have been skipped. Instead, buffer subtraction was performed to ensure that buffer information was not included in the data sets that were used to predict long term behavior.

5.3.2 Filtering

A buffer dependent signal was removed from the ANS spectra prior to filtering. As can be observed in the ANS spectrum plots of Figure 5.2, the spectrum drops to near zero at some

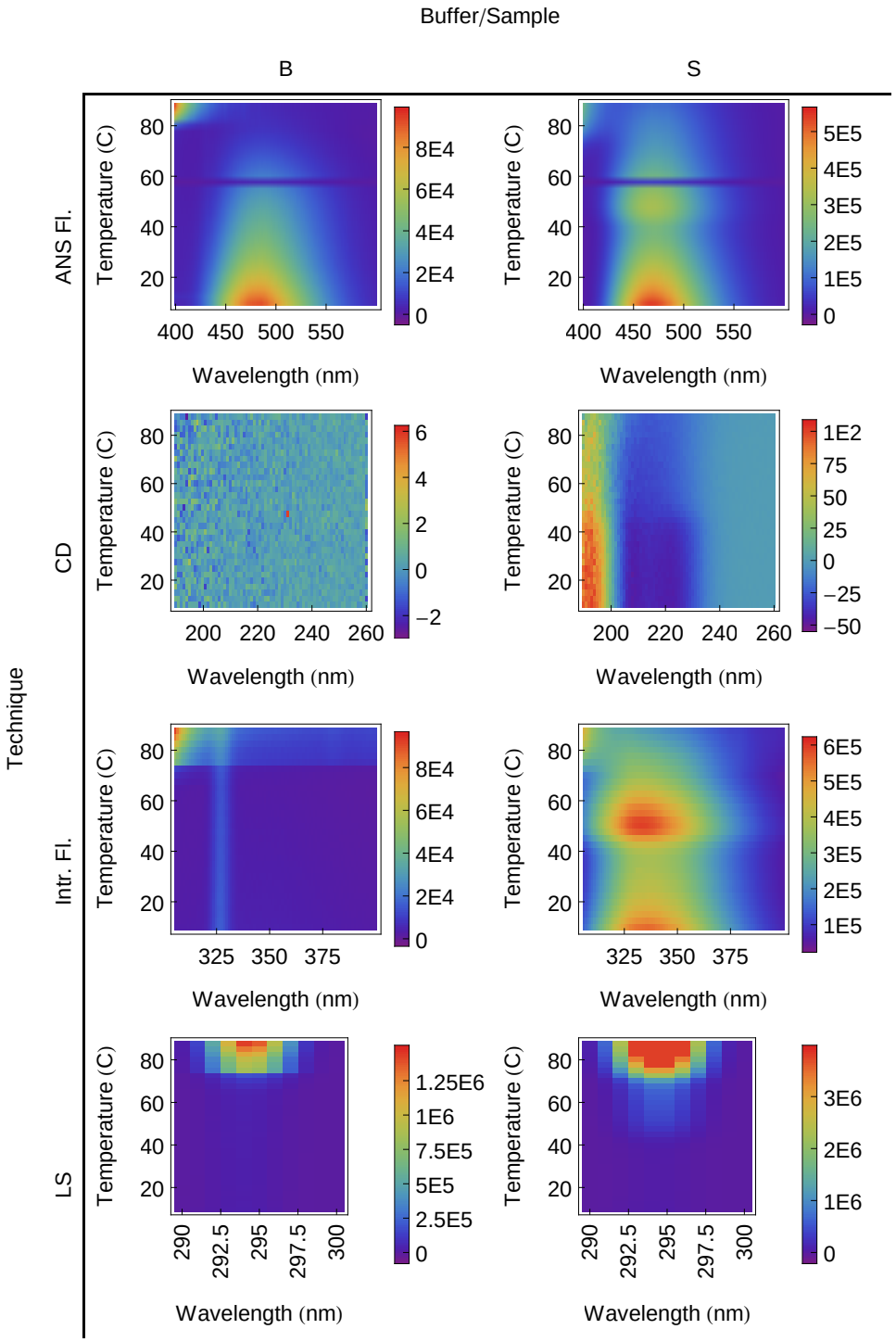


Figure 5.2: Plots of short term spectroscopic measurements for formulation 11. The buffers presented a relatively large spectral signal. A small part of the signal in the buffers is presumably due to the buffer itself. The signal is much larger than what would usually be attributable to buffer alone, however, and may be partly due to protein contamination. The band near 60 °C in the ANS fluorescence plot is discussed in the text.

temperatures. We don't know the cause of this transient behavior. It occurs in both the buffer and sample spectra, so is assumed to originate in the buffer. This artifact was removed to ensure that buffer information was not included in the data sets that were used to predict long term behavior. It was removed by replacing the spectrum of an affected temperature by the average of the two spectra at neighboring temperatures.

The spectra were then filtered with a Savitzky-Golay filter of radius 4nm, order 3, followed by a Gaussian smooth of radius 4nm. The melts at each wavelength (i.e., the temperature dimension) were filtered with a Savitzky-Golay filter of radius 3 °C , order 3, followed by a Gaussian smooth of radius 3 °C .

Spectral regions of large peaks were then selected to lessen the effect of noise on later analysis steps. Circular dichroism was restricted to the 205-260nm range, intrinsic fluorescence to the 310-365nm range and light scattering to the 290-300nm range. Entire ANS fluorescence spectra were left alone.

5.3.3 Determination of transition temperatures

Transition temperatures were determined automatically using the derivative method. First, each spectral melt (possessing wavelength and temperature dimensions) was reduced to a one dimensional melt by averaging over wavelengths. The melt was then filtered with a 3rd derivative Savitzky-Golay filter of radius 5 °C , order 3. As illustrated in Figure 5.3, this operation results in peaks corresponding to the transition onset, midpoint, and endset. This figure shows the zeroth, first, second, and third derivatives of intrinsic fluorescence melts for formulations 13 and 14. The zero derivative melts are the fluorescence intensity averaged over wavelength. A peak finding algorithm was applied to the third derivative melts, finding peaks greater than a predetermined size relative to the standard deviation of the 3rd derivative melt. The peak finder chooses the lowest temperature such peak, then finds the two neighboring peaks of opposite sign. Transition onset, midpoints, and endsets were found in this manner for all formulations and measurement techniques, and are shown

Table 5.3: Transition temperatures that were determined automatically using the derivative method. See Figure 5.3 for a summary of the method.

Transition type (C)

Formulation	Endset				Midpoint				Onset			
	Technique				Technique				Technique			
	ANS Fl.	CD	Intr. Fl.	LS	ANS Fl.	CD	Intr. Fl.	LS	ANS Fl.	CD	Intr. Fl.	LS
11	51	52	52	53	44	45	45	46	36	36	36	38
12	65	68	66	66	58	62	60	59	52	54	51	52
13	62	64	61	58	52	57	53	51	43	48	46	44
14	48	52	48	66	40	44	40	58	32	34	34	51
15	62	64	63	63	55	56	56	56	48	49	49	49
16	58	57	60	58	47	48	48	46	40	39	40	39
17	60	60	60	59	50	49	50	49	42	42	42	41
18	61	61	61	60	54	54	54	54	47	48	48	47
19	78	76	76	79	71	69	69	71	64	61	61	64
20	78	77	78	80	70	70	70	72	64	62	62	66
21	70	70	68	68	62	64	62	62	56	56	56	56
22	72	73	70	71	64	66	64	64	58	60	57	57
23	68	69	68	68	62	62	62	62	56	56	55	56
24	64	66	64	64	58	59	58	58	52	52	51	51
25	71	67	66	67	60	60	60	60	54	54	53	53
26	63	66	63	62	57	60	56	56	51	52	50	50

in Table 5.3

5.3.4 Construction of data set to predict

The data to predict consisted of the long term measurements. As will be seen, many of these measurements could not be modeled well. Two adjustments were made to improve modeling.

First, the particle count measurements were distributed over several orders of magnitude, yet we attempted to model them from short term spectral measurements, which were distributed more evenly on a linear scale. To aid in predicting particle counts from spectra, the base 10 logarithm of the particle counts was used.

Second, we averaged several long term measurements to reduce the influence of error. The effect of error could not be investigated quantitatively since the data in this project was only measured once. Log particle counts were averaged over storage conditions and

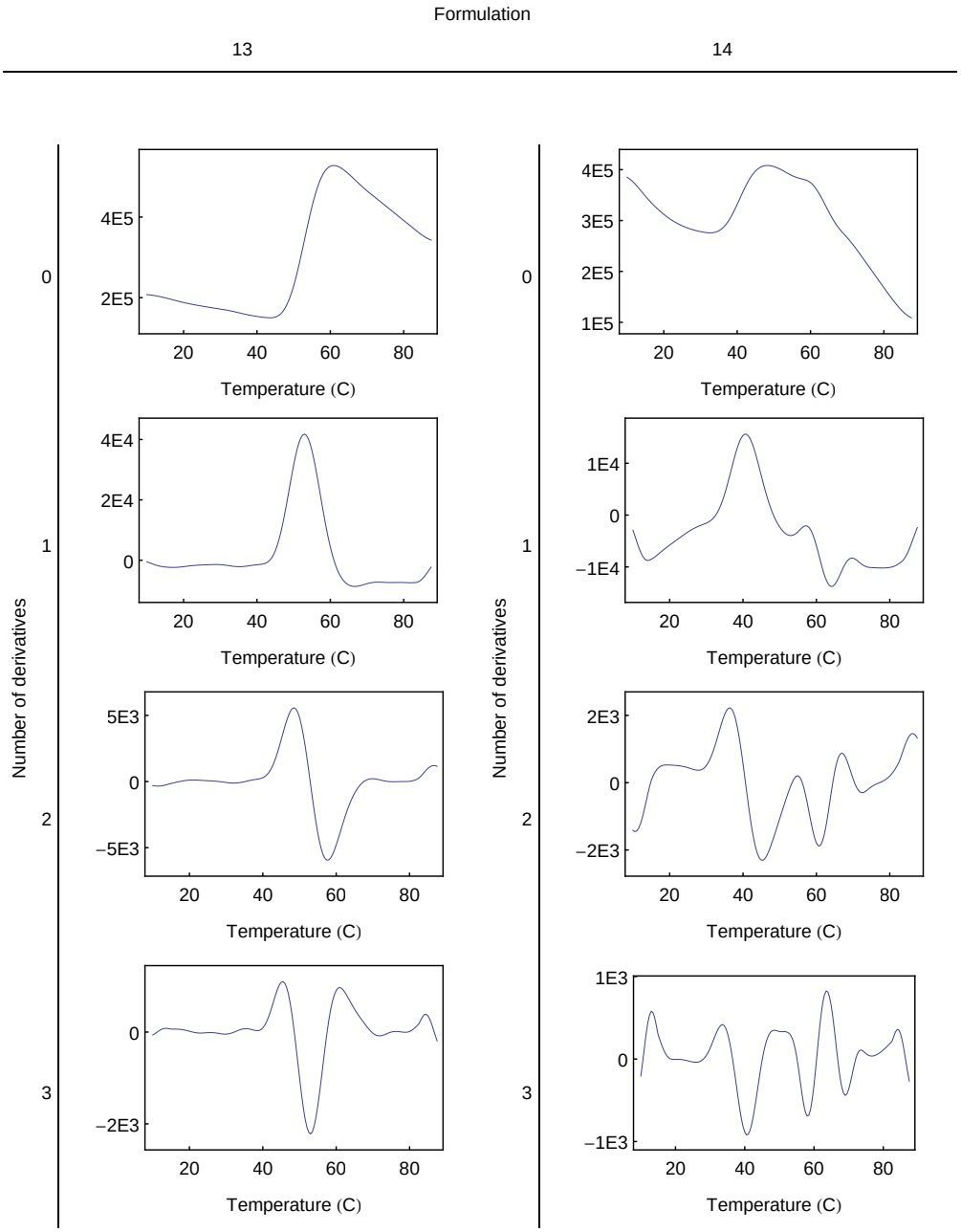


Figure 5.3: Derivatives of intrinsic fluorescence melts for formulations 13 and 14. The melts with no derivative are the fluorescence intensity averaged over wavelength. The third derivative of a melt results in peaks corresponding to the transition onset, midpoint, and endset. The positions of these peaks were determined with a peak finding algorithm, and the resulting transition temperatures are shown in Table 5.3.

size ranges, obtaining a single mean particle count. The separate particle counts were kept in the data set to be predicted. Turbidities and MLRC's were also averaged over storage conditions, and the mean values inserted in the data set to be predicted.

5.3.5 Construction of data sets from which to predict stability

Long term measurements were predicted from several types of short term data and the quality of the resulting predictions were compared. The short term data sets from which predictions are made will be called predictor data sets. Five of them were constructed, and are described below.

Before discussing the construction of various predictor data sets, two operations must be defined that will be referred to repeatedly. The first operation, which we call flattening, consists of combining dimensions in an array to make a single dimension. For instance, we will have a $16 \times 32 \times 11$ dimensional array of light scattering measurements (16 formulations, 32 temperatures, and 11 wavelengths). Flattening the wavelength and temperature dimensions results in a 16×352 dimensional array, where the second dimension contains positions for each combination of temperature and wavelength.

The second operation we call standardization. The experimental techniques used in this work (circular dichroism, intrinsic fluorescence, ANS fluorescence, and light scattering) return data with differing scales and number of wavelengths. Circular dichroism measurements, for instance, were approximately in the range 0 to -50 millidegrees of ellipticity, whereas light scattering data ranged from 0 to approximately 10^6 photons/s. If prediction techniques were applied to the raw data, CD data would have much less influence on the resulting models than light scattering data, since peak CD measurements are approximately 10^4 times smaller than peak light scattering measurements. Even if the measurements from all experimental techniques were rescaled to be in the same range, an experimental technique that returns very few measurements compared to another technique would have less influence on the resulting models. To account for the influence of scales and numbers of

measurements, the data was rescaled as follows. The result of a preprocessing method is a list of arrays, one array for each measurement technique. Each array is two dimensional, possessing a formulation dimension and a "measurement" dimension resulting from flattening other dimensions returned by the preprocessing method. Each array was standardized by dividing by its Frobenius norm:

$$D_{ij}^{standardized} = \frac{D_{ij}}{\sqrt{\sum_{ij} |D_{ij}|^2}}, \tag{5.1}$$

where D_{ij} is data resulting from a preprocessing method, i indexes formulations and j indexes measurements for a given formulation.

The third operation we call concatenation. Preprocessing, flattening, and standardization steps result in a list of 2 dimensional arrays, one array for each experimental technique. What is desired for a given preprocessing method is a single array with formulation and measurement dimensions. To achieve this, for each formulation the data from all the measurement techniques are combined along the measurement dimension. For instance, we will have a 16×71 array of circular dichroism measurements (16 formulation, 71 wavelengths), a 16×11 light scattering array, a 16×95 intrinsic fluorescence array, and a 16×201 ANS fluorescence array. Concatenating these arrays along the wavelength dimension results in a single array with dimensions $16 \times (71 + 11 + 95 + 201)$, or 16×378.

Low temperature spectra

Good long term predictions from low temperature spectra would be particularly interesting, since such measurements are both convenient and inexpensive. A protein's low temperature spectrum can be measured in minutes or less, whereas a full thermal unfolding study requires at least an hour to perform. In addition, if the protein has not undergone major structural alteration, then it can be recovered by dialysis and reused. This can be useful in the beginning stage of the development of a new protein drug, when there is often very little of the protein

available.

The spectra of GCSF at low temperature were obtained by averaging thermal melt spectra over the temperature range 10 - 35 °C . An average was used instead of the spectrum at 10 °C in order to increase the signal to noise ratio. Averaging over temperatures resulted in a list of arrays, one array for each measurement technique (CD, intrinsic fluorescence, and ANS fluorescence), each array possessing formulation and wavelength dimensions. These arrays were standardized, and then concatenated along the wavelength dimension.

Light scattering data was not used, since the low temperature, non-aggregated form of GCSF did not present a large light scattering signal. Inclusion of light scattering data was found to lower the quality of predictions from low temperature spectra.

Thermal melts

Each spectral melt (possessing wavelength and temperature dimensions) was reduced to a one dimensional melt by averaging over wavelengths. The resulting arrays were standardized, and then concatenated along the temperature dimension.

Transition temperatures

Transition temperatures were determined by the derivative method, as described above. Arrays were standardized, and then concatenated along the "Transition type" dimension. Concatenation resulted in a single two dimensional array with a formulation dimension and a dimension containing the transition onset, midpoint, and endset for each measurement technique.

Transition regions

Spectral melts were restricted to a range of temperatures around the thermal transition temperatures. The thermal transition temperatures used were the thermal transition midpoints averaged over the spectroscopic measurement techniques. A single transition temperature

was thus used for each formulation. For a given formulation, spectra were restricted to within ±10 °C of that formulation's transition temperature. Plots of the resulting data are shown in Figure 5.4. The temperature and wavelength dimension of the arrays were then flattened, and the arrays were then standardized and concatenated along the measurement dimension, yielding a single 2 dimensional array with measurement and temperature dimensions.

This manner of preprocessing the data is based on the accepted idea that the behavior of proteins near thermal transitions, in particular the relative behavior of different types of protein structure, provides information about unfolding mechanisms. Extracting transition regions for use in prediction algorithms is therefore a way of automating traditional comparative structural analyses. It is a "registering" operation, where one takes data sets that do not match up when overlaid, and modifies them so that critical points do match up. For example, facial recognition algorithms begin by rotating, shifting, and resizing images so that facial features of different images match up when overlaid.

All predictors

A predictor data set was made by combining into a single array all of the predictor data sets defined above. This was done by standardizing the arrays, and then concatenating them along their measurement dimension.

5.3.6 Prediction methods

An input predictor matrix X_{ij}, where $i = 1...m$ indexes formulations and $j = 1..n$ indexes predictor measurements, was centered so that each column had zero mean:

$$M_j^{(X)} = \frac{1}{m} \sum_i X_{ij} \tag{5.2}$$

$$\overline{X}_{ij} = X_{ij} - M_j^{(X)}. \tag{5.3}$$

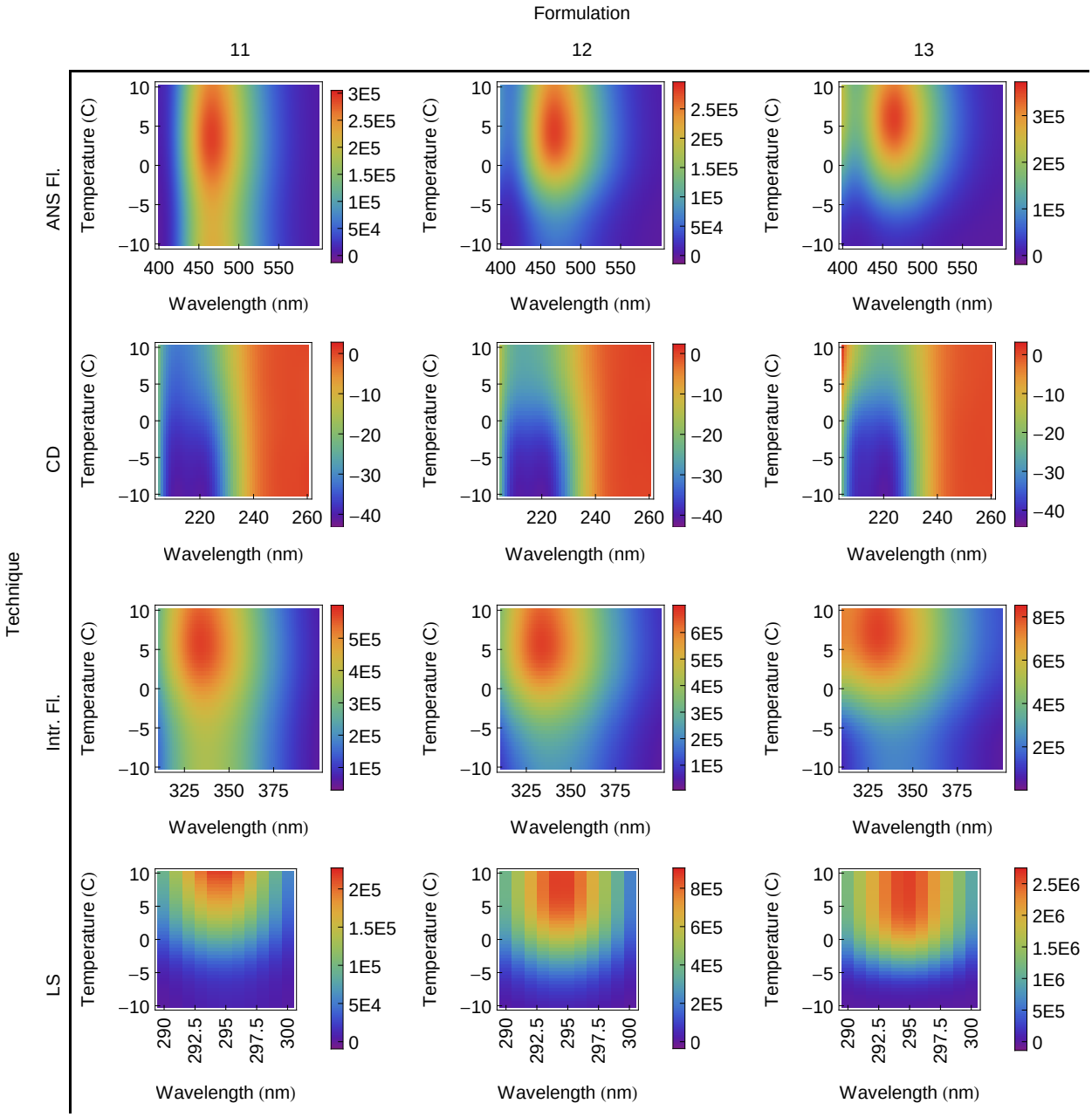


Figure 5.4: Plots of spectral melts that have been restricted to a range of temperatures around the thermal transition temperatures. Each temperature axis zero value corresponds to the thermal transition temperature for the associated formulation.

The input long term observation matrix Y_{ik}, where $i = 1...m$ indexes formulations and $k = 1...r$ indexes long term observations, was centered so that each column had zero mean:

$$M_k^{(Y)} = \frac{1}{m} \sum_i Y_{ij} \tag{5.4}$$

$$\overline{Y}_{ij} = Y_{ik} - M_k^{(Y)} \tag{5.5}$$

Least squares fits

The linear model

$$\sum_j \overline{X}_{ij} T_{jk} = \overline{Y}_{ik} \tag{5.6}$$

was fit to the data using least squares (LSQ, to distinguish it from Light Scattering). $\overline{X}$ is the standardized predictor matrix, $\overline{Y}$ is the matrix of standardized long term measurements, T is the model determined by least squares, i indexes formulations, j indexes predictor measurements, and k indexes long term measurements.

The fitted least squares model was applied to test data $X_j^{(test)}$ in the following manner. The data was first centered using the mean values found previously:

$$\overline{X}_j^{(test)} = X_j^{(test)} - M_j^{(X)}. \tag{5.7}$$

The predicted observations were then found with

$$Y_k^{(test)} = \sum_j \overline{X}_j^{(test)} T_{jk} + M_k^{(Y)} \tag{5.8}$$

Radial Basis Function Network (RBFN) fits

A standardized predictor matrix $\overline{X}_{ik}$ was first transformed via

$$\widetilde{X}_{ij} = e^{-\frac{\sum_k (\overline{X}_{ik} - \overline{X}_{jk})^2}{m\lambda^2}}, \tag{5.9}$$

where i and j index formulations and k indexes predictor measurements. Roughly speaking, the effect of this transformation is to make locations in $\overline{X}$ correspond to directions in $\widetilde{X}$. The width parameter λ was set at 1.0, chosen manually by checking fit results for all predictors and long term measurements (as shown in Table 5.4).

The linear model

$$\sum_j \widetilde{X}_{ij} U_{jk} = \overline{Y}_{ik} \tag{5.10}$$

was then fit to the data using least squares. U is the model determined by least squares, i and j index formulations, and k indexes long term measurements.

The fitted RBFN model was applied to test data $X_j^{(test)}$ in the following manner. The test data was first centered using the mean values found previously, as shown in Equation 5.7. $\overline{X}_j^{(test)}$ was then transformed via

$$\widetilde{X}_j^{(test)} = e^{-\frac{\sum_k (\overline{X}_k^{(test)} - \overline{X}_{jk})^2}{m\lambda^2}}, \tag{5.11}$$

where j and k index formulations. The predicted observations were then found with

$$Y_k^{(test)} = \sum_j \widetilde{X}_j^{(test)} U_{jk} + M_k^{(Y)}. \tag{5.12}$$

5.3.7 Leave one out cross validation

Cross validation was performed for each combination of predictor matrix and prediction method. The model was fitted with one of the formulations omitted. The model was then

tested on this formulation, giving long term predictions $\mathrm{Y}_k^{(test)}$, where k indexes predictor measurements. This was repeated for each formulation, yielding a matrix $Y_{ik}^{(test)}$ of long term predictions, where $i = 1...m$ indexes formulations.

5.3.8 Estimate of fit likelihood

We compared predictions $Y_{il}^{(test)}$ to observations Y_{il}. In this subsection, i is variable and indexes formulations, whereas l is constant and corresponds to a specific long term observation such as chemical stability. Correlation plots of the best 16 fits are shown in Figure 5.6. The plot labels indicate the measurement predicted, the data used to predict from, and the prediction method. Each point is a formulation, its horizontal position is the observed long term measurement, its vertical position is the predicted long term measurement and the horizontal and vertical axii extend over the same range.

The likelihood of all the fits was estimated using the Pearson correlation coefficient permutation (PCC) test. The Pearson correlation coefficient $PCC(Y_{il}^{(test)}, Y_{il}^{(observed)})$ of two vectors is defined as:

$$PCC(Y_{il}^{(test)}, Y_{il}^{(observed)}) = \frac{1}{m-1} \sum_i \left(\frac{Y_{il}^{(test)} - M^{(test)}}{S^{(test)}} \right) \left(\frac{Y_{il}^{(observed)} - M^{(observed)}}{S^{(observed)}} \right) \tag{5.13}$$

where

$$M^{(test)} = \frac{1}{m} \sum_i Y_{il}^{(test)}, \tag{5.14}$$

$$S^{(test)} = \sqrt{\frac{\sum_i (Y_{il}^{(test)} - M^{(test)})^2}{m-1}}, \tag{5.15}$$

and likewise for $M^{(observed)}$ and $S^{(observed)}$.

A null distribution for the observations was constructed by generating 5000 random permutations of $Y_{il}^{(observed)}$, permuting in the formulation index i. A null distribution for the PCC was then computed by finding the PCC of $Y_{il}^{(test)}$ with each of the 5000 permutations of $Y_{il}^{(observed)}$. Fit likelihoods are reported as σ values, found by dividing $PCC(Y_{il}^{(test)}, Y_{il}^{(observed)})$ by the standard deviation of the null distribution for the PCC. Negative σ values correspond to predictions that are anti-correlated with observations.

Figure 5.5 shows the histogram of a PCC null distribution constructed for the chemical stability of GCSF (as predicted from thermal melts using a least squares fit). The red line is the Pearson correlation coefficient of predictions and observations, $PCC(Y_{il}^{(test)}, Y_{il}^{(observed)})$.

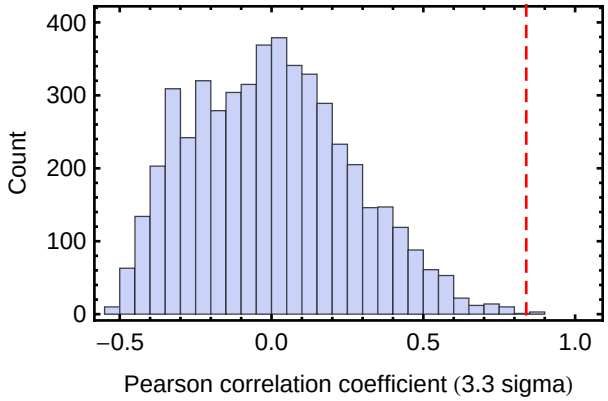


Figure 5.5: A measure of goodness of fit, the Pearson correlation coefficient, compared to a null distribution constructed for the measure. The histogram shows a null distribution for the Pearson correlation coefficient between predictions and observations of chemical stability of GCSF. Chemical stability was predicted from thermal melts using a least squares fit. The null distribution was generated using 5000 permutations of observations of the chemical stability. The red line is the Pearson correlation coefficient between predictions and observations, showing a fit likelihood of 2.7 sigma. See the text for a more complete description.

5.4 Results and Discussion

Leave one out cross validation was performed on each combination of prediction method (LSQ and RBFN), predictor data set (low temperature spectra, thermal melts, transition temperatures, transition regions, and all data sets together), and long term measurement. The likelihood of the resulting fits was determined with the Pearson correlation coefficient

Table 5.4: Fit likelihood values of fits resulting from leave-one-out cross validation for the least squares prediction method. Refer to Figure 5.5 and the text for information on how the fit likelihoods were determined. Significance values greater than 2 are highlighted in yellow, and those greater than 3 are highlighted in green.

Measurement	Predictor: All predictors	Low T spectra	Melts	Transition T's	Trans. Regions
Chemical Stability (%)	3.2	3.4	3.2	1.6	2.4
Log PC, 03 mo, >01um	−0.5	2.3	−0.2	0.2	0.
Log PC, 03 mo, >10um	0.2	2.1	0.5	−0.1	2.3
Log PC, 03 mo, >25um	1.2	2.5	1.5	−0.2	2.6
Log PC, 10 mo, >01um	1.6	0.	0.	0.1	1.6
Log PC, 10 mo, >10um	1.2	0.3	1.7	0.7	3.3
Log PC, 10 mo, >25um	0.4	1.4	1.2	0.8	1.7
Log PC, 20 mo, >01um	2.7	0.4	2.5	0.4	2.3
Log PC, 20 mo, >10um	0.9	0.1	1.3	1.3	2.8
Log PC, 20 mo, >25um	1.5	0.2	1.6	1.2	0.8
Log PC, mean	1.5	1.1	2.	0.7	3.1
MLRC, 03 mo	3.5	2.1	3.	2.9	2.2
MLRC, 10 mo	2.8	1.5	2.6	−0.1	3.3
MLRC, 20 mo	1.6	−1.5	−0.6	1.	2.4
MLRC, mean	3.4	2.3	2.6	3.1	2.3
Turbidity, 03 mo	−0.9	−1.6	−0.6	0.3	−1.1
Turbidity, 10 mo	1.3	−0.3	1.6	−1.3	1.6
Turbidity, 20 mo	1.4	0.5	1.3	−0.1	1.4
Turbidity, mean	−0.3	1.1	−0.2	−0.3	0.4

permutation test. The fit likelihoods are shown in Tables 5.6 and 5.6. The fit likelihoods in these tables are measured in units of the standard deviation of the null distributions constructed for each permutation test. Correlation plots of the top 16 best fits are shown in Figure 5.6.

Of all the separate predictors, transition regions provided the largest number of fit likelihoods greater than 2 sigma. This was true for both LSQ and RBFN prediction. For LSQ prediction, combining all predictors into one gave improved results. For RBFN prediction, combining all predictors into one yielded results similar to prediction from transition regions. Standardization of the the predictors will need to be refined so that they contribute more equally when combined. RBFN and LSQ prediction gave similar results for prediction using

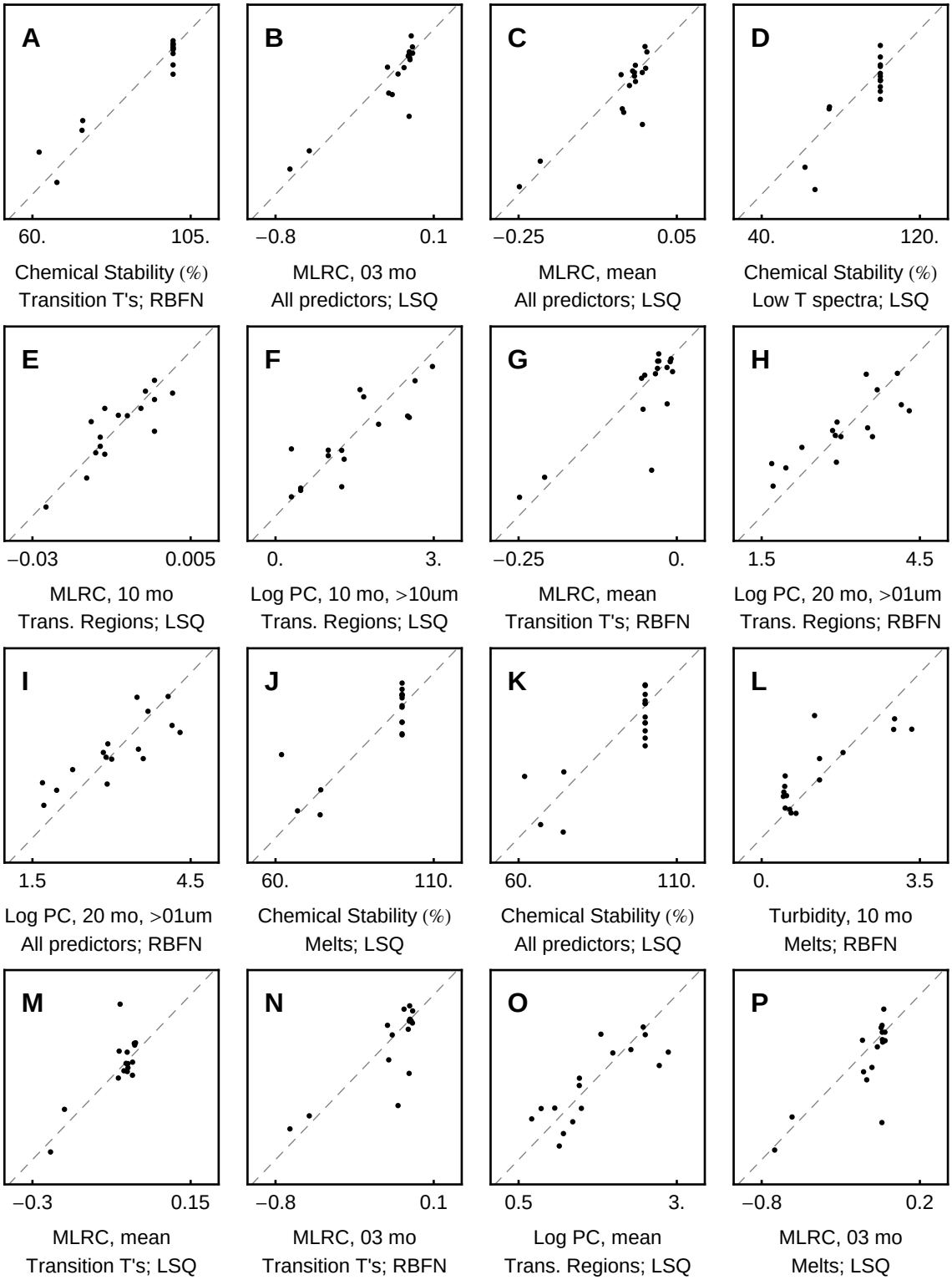


Figure 5.6: Correlation plots of fits resulting from leave-one-out cross validation. The plot labels indicate the measurement predicted, the data used to predict from, and the prediction method. Each point corresponds to a formulation. The horizontal and vertical axii correspond to observed and predicted measurements, respectively, and extend over the same range. The likelihood of each fit was estimated using the Pearson coefficient permutation test, and the 16 best fits are shown here in descending order.

Table 5.5: Fit likelihood values of fits resulting from leave-one-out cross validation for the radial basis function network prediction method. Refer to Figure 5.5 and the text for information on how the fit likelihoods were determined. Significance values greater than 2 are highlighted in yellow, and those greater than 3 are highlighted in green.

Measurement	Predictor: All predictors	Low T spectra	Melts	Transition T's	Trans. Regions
Chemical Stability (%)	2.5	2.7	2.7	3.8	2.4
Log PC, 03 mo, >01um	0.6	1.5	1.2	−0.6	0.6
Log PC, 03 mo, >10um	1.5	1.7	1.4	−0.7	1.5
Log PC, 03 mo, >25um	1.6	1.6	1.5	0.5	1.6
Log PC, 10 mo, >01um	3.	0.4	2.6	1.8	3.
Log PC, 10 mo, >10um	2.4	0.6	2.1	0.4	2.4
Log PC, 10 mo, >25um	1.3	1.2	1.9	−0.4	1.3
Log PC, 20 mo, >01um	3.2	1.3	2.6	1.2	3.2
Log PC, 20 mo, >10um	1.9	0.6	2.	−0.1	1.9
Log PC, 20 mo, >25um	1.5	0.8	2.4	−0.1	1.5
Log PC, mean	2.3	1.5	2.5	0.4	2.3
MLRC, 03 mo	2.5	0.1	1.6	3.1	2.5
MLRC, 10 mo	1.1	1.	2.	1.8	1.
MLRC, 20 mo	2.6	−1.8	0.7	0.5	2.6
MLRC, mean	2.5	0.5	1.5	3.2	2.4
Turbidity, 03 mo	−1.5	−0.1	−0.6	−0.5	−1.5
Turbidity, 10 mo	2.4	0.3	3.1	−0.2	2.4
Turbidity, 20 mo	1.2	1.7	2.2	1.3	1.2
Turbidity, mean	0.1	1.5	1.	−0.9	0.

the combined predictor.

Least squares prediction from low temperature spectra performed nearly as well as prediction from transition regions. This is somewhat surprising, since low temperature spectra consist of conformational information primarily concerning only the non-degraded form, whereas the transition region data contains information on the susceptibility of various levels of protein structure to alteration by temperature.

What would the fit results look like for a larger factorial design of formulations? Including more formulations could improve the fits by inclusion of phenomena easily predicted by the short term measurements. On the other hand, inclusion of phenomena not easily predicted by the short therm measurements would worsen the fits. It's likely that both situations will

occur simultaneously. The former situation may occur in the following manner. Extremes in formulation parameters were avoided in the factorial design of formulations, since the long term behavior might be more easily predictable for those conditions.[1] Inclusion of those extremes may improve the fits. The latter situation (worsening of fits with added formulations) could occur if the factorial design of formulations investigated compounds that strongly affect the chemical stability of proteins. In such a study, short term measurements of chemical stability would be useful, such as measurements of protein spectral changes during titration of destabilizing agents.

Not all determinants of protein stability will be accessible in a factorial design of formulations. For example, pharmaceutical proteins are the product of variable manufacturing processes. Will the methodology being proposed still work if it is developed using protein from one manufacturing process, and then used on protein from another manufacturing process? This question is not answerable using the data in this chapter. We note, however, that the proposed methodology employs a super set of traditional methods, using predictive modeling to join the strengths of short term and long term assessments of protein stability. Thus, it need not perform worse than traditional methods of assessing comparability, since these use short term and long term assessments separately and do not typically employ predictive modeling.

One might be concerned that prediction mistakes will only be minimized with respect to the training data set. This is, however, the case with modeling in general and science itself. Models predict the results of experiments that have already been performed, provide insight into mechanisms, and are replaced when conflicting data arrives. The prevalence of prediction errors can be investigated by cross validation, wherein a model is first fitted to a subset of a data set and then tested on a subset of the data disjoint from the subset it was trained on. (Cross validation is analogous to generating and testing hypotheses, as found in the scientific method.) Formulation errors resulting from application of this new methodology can be managed in the same manner as formulation errors are currently

managed: by comprehensive testing of final candidate formulations.

5.5 Conclusion

Based on results shown for the small data set described here, it appears that the long term stability of proteins can at least in some cases be modeled from short term spectroscopic measurements. The best predictions of long term behavior were obtained from spectroscopic measurements in the neighborhood of thermal transitions. (Such neighborhoods are shown in Figure 5.4.) This agrees with the accepted idea that the behavior of proteins near thermal transitions provides information on unfolding mechanisms. This manner of extracting transition regions for use in prediction algorithms is a way of automating traditional comparative structural analyses, and appears to be effective.

The method could be used either before or after a 2-3 year isothermal stability study. In both cases, at the beginning of the study one would perform comprehensive spectroscopic characterization of all formulations used in the long term study. After performing a 2-3 year isothermal stability study of a protein or vaccine, the resulting predictive models could be used to explore the formulation space more fully. For example, since the method predicts behavior from behavior, one could investigate the effect of new combinations of formulation parameters. This can be difficult to do with polynomial modeling of long term behavior from formulation parameters.

If applied prior to a full 2-3 year isothermal stability study, the method would be used to direct and focus the long term study. One would perform a shorter long term study lasting several months and obtain the predictive models. Spectroscopic techniques would then be used to explore large regions of formulation space for promising candidates to be included in a 2-3 year study.

The method can be extended to include measurements of biological activity. For example, various assays of biological activity after long term storage could be modeled from similar

assays before storage.

Bibliography

[1] Ahmed MK Youssef and Gerhard Winter. "A critical evaluation of microcalorimetry as a predictive tool for long term stability of liquid protein formulations: Granulocyte Colony Stimulating Factor (GCSF)". In: *European Journal of Pharmaceutics and Biopharmaceutics* (2013).

[2] L A Kueltzo et al. "Derivative absorbance spectroscopy and protein phase diagrams as tools for comprehensive protein characterization: A bGCSF case study". In: *J. Pharm. Sci.* 92 (2003), pp. 1805–20.

[3] H Fan et al. "Solution behavior of IFN-β-1a: An empirical phase diagram based approach". In: *J. Pharm. Sci.* 94 (2005), pp. 1893–911.

[4] N.R. Maddux et al. "Multidimensional methods for the formulation of biopharmaceuticals and vaccines". In: *J. Pharm. Sci.* 100.10 (2011), pp. 4171–4197.

[5] N.R. Maddux et al. "An Improved Methodology for Multidimensional High-Throughput Preformulation Characterization of Protein Conformational Stability". In: *J. Pharm. Sci.* 101 (2012), pp. 2017–2024.

[6] Juan M Luco. "Prediction of the brain-blood distribution of a large set of drugs from structurally derived descriptors using partial least-squares (PLS) modeling". In: *Journal of chemical information and computer sciences* 39.2 (1999), pp. 396–404.

[7] Johan A Westerhuis and Pierre MJ Coenegracht. "Multivariate modelling of the pharmaceutical two-step process of wet granulation and tableting with multiblock partial least squares". In: *Journal of chemometrics* 11.5 (1999), pp. 379–392.

[8] RD Bautista et al. "Simultaneous spectrophotometric determination of drugs in pharmaceutical preparations using multiple linear regression and partial least-squares regression, calibration and prediction methods". In: *Talanta* 43.12 (1996), pp. 2107–2115.

[9] Héctor C Goicoechea et al. "Complementary use of partial least-squares and artificial neural networks for the non-linear spectrophotometric analysis of pharmaceutical samples". In: *Analytical and bioanalytical chemistry* 374.3 (2002), pp. 460–465.

[10] Ulf Norinder and Thomas Osterberg. "Theoretical calculation and prediction of drug transport processes using simple parameters and partial least squares projections to latent structures (PLS) statistics. The use of electrotopological state indices". In: *Journal of pharmaceutical sciences* 90.8 (2001), pp. 1076–1085.

[11] Marcelo M Sena et al. "Direct determination of diclofenac in pharmaceutical formulations containing B vitamins by using UV spectrophotometry and partial least squares regression". In: *Journal of pharmaceutical and biomedical analysis* 36.4 (2004), pp. 743–749.

Appendix A

DART script used in Chapter 5

A.1 Import short term data

A.1.1 Import circular dichroism (CD) data

First we import all the CD files without any modifications. The function scanFolder[] recurses through a specified folder, importing instrument files of the specified type (Jasco in this case) into DART arrays. It returns a list of DART arrays, one for each file found in the folder. It would recurse through folders and subfolders if they existed, but in this case the folder only contains files.

```
cd = scanFolder[
    importPath <> "shortTerm\\exported\\CD\\", "*.txt",
    importJasco
];
```

The function dimInfo[] prints out the dimensions in an array, which helps make the scripts more readable and easier to debug.

```
dimInfo[cd[[1]]]
```

Name	Unit	Length	Scale
fileLocation		1	"GCSF 11-12-13 04302009A10.txt"
Wavelength	nm	71	260.,259., ... ,191.,190.

Some filenames did not follow the same format as others, and were missing a space. Inserting the missing space will help simplify the regular expressions used later on.

```
cd = operateDimension[
    StringReplace[#, "14-15-1604" -> "14-15-16 04"]&,
```

```
    cd,
    "fileLocation"
];
```

The “fileLocation” dimension only has 1 position, and is used as a label for each array. We now parse it with a regular expression, returning the formulation, cell, and temperature information found in the file location. Note that parse is automatically applied separately to each DART array in the variable “cd”.

```
cd = parse[cd, "fileLocation", "GCSF (\\S+) \\d+([A-F])(\\d+)\\.txt", 3];
```

Rename the dimensions just extracted. These functions also are automatically applied to each DART array in the list.

```
cd = renameDimension[cd, "parsedDim1", "Formulation"];
cd = renameDimension[cd, "parsedDim2", "Cell"];
cd = renameDimension[cd, "parsedDim3", "Temperature"];
cd = makeNumericDims[cd, {"Temperature"}];
cd = dropSingletons[cd, {"fileLocation"}];
```

Replace dimensions found in the filenames with their meanings. This operation is data entry, and can’t be avoided since the information being applied is only in lab notebooks and nowhere in the files. First, set it up.

```
fromDimNames = {"Formulation", "Cell"};
toDimNames = {"Formulation", "Buffer/Sample"};
toDimUnits = {"", ""};

fromToPositions = {
 {"11-12-13", "A", 11, "B"}, {"11-12-13", "B", 11, "S"},
 {"11-12-13", "C", 12, "B"}, {"11-12-13", "D", 12, "S"},
 {"11-12-13", "E", 13, "B"}, {"11-12-13", "F", 13, "S"},
 {"14-15-16", "A", 14, "B"}, {"14-15-16", "B", 14, "S"},
 {"14-15-16", "C", 15, "B"}, {"14-15-16", "D", 15, "S"},
 {"14-15-16", "E", 16, "B"}, {"14-15-16", "F", 16, "S"},
 {"17-18-19", "A", 17, "B"}, {"17-18-19", "B", 17, "S"},
 {"17-18-19", "C", 18, "B"}, {"17-18-19", "D", 18, "S"},
 {"17-18-19", "E", 19, "B"}, {"17-18-19", "F", 19, "S"},
 {"20-21-22", "A", 20, "B"}, {"20-21-22", "B", 20, "S"},
 {"20-21-22", "C", 21, "B"}, {"20-21-22", "D", 21, "S"},
 {"20-21-22", "E", 22, "B"}, {"20-21-22", "F", 22, "S"},
 {"23-24-25", "A", 23, "B"}, {"23-24-25", "B", 23, "S"},
 {"23-24-25", "C", 24, "B"}, {"23-24-25", "D", 24, "S"},
 {"23-24-25", "E", 25, "B"}, {"23-24-25", "F", 25, "S"},
```

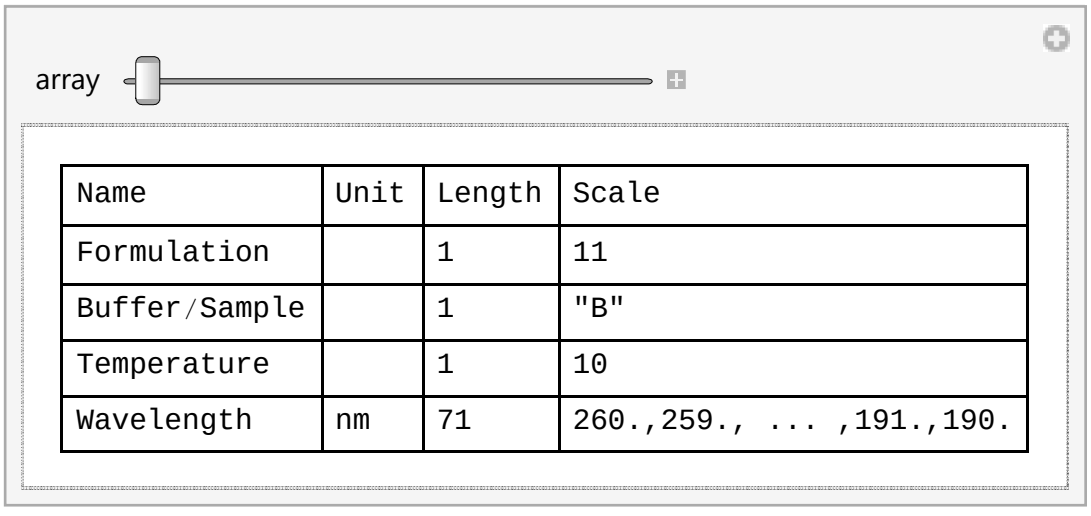
```
 {"26", "A", 26, "B"}, {"26", "B", 26, "S"}
};

fromToPositions = Map[Take[#,2]->Take[#, -2]&, fromToPositions]
```

```
{{11-12-13, A} → {11, B}, {11-12-13, B} → {11, S},
 {11-12-13, C} → {12, B}, {11-12-13, D} → {12, S}, {11-12-13, E} → {13, B},
 {11-12-13, F} → {13, S}, {14-15-16, A} → {14, B}, {14-15-16, B} → {14, S},
 {14-15-16, C} → {15, B}, {14-15-16, D} → {15, S}, {14-15-16, E} → {16, B},
 {14-15-16, F} → {16, S}, {17-18-19, A} → {17, B}, {17-18-19, B} → {17, S},
 {17-18-19, C} → {18, B}, {17-18-19, D} → {18, S}, {17-18-19, E} → {19, B},
 {17-18-19, F} → {19, S}, {20-21-22, A} → {20, B}, {20-21-22, B} → {20, S},
 {20-21-22, C} → {21, B}, {20-21-22, D} → {21, S}, {20-21-22, E} → {22, B},
 {20-21-22, F} → {22, S}, {23-24-25, A} → {23, B}, {23-24-25, B} → {23, S},
 {23-24-25, C} → {24, B}, {23-24-25, D} → {24, S}, {23-24-25, E} → {25, B},
 {23-24-25, F} → {25, S}, {26, A} → {26, B}, {26, B} → {26, S}}
```

Now we replace the dimensions.

```
cd = replaceDimensions[
    cd, fromDimNames, toDimNames, toDimUnits, fromToPositions
];
cd = makeNumericDims[cd, {"Formulation"}];
```

```
dimInfo[cd]
```



Name	Unit	Length	Scale
Formulation		1	11
Buffer/Sample		1	"B"
Temperature		1	10
Wavelength	nm	71	260.,259., ... ,191.,190.

Now that all the arrays are labeled properly, they can be merged into a single array.

```
cd = merge[cd];
```

Add a label for later

```
cd = addSingletons[cd, {newDimension["Technique", "", "CD"]}];
```

```
dimInfo[cd]
```

Name	Unit	Length	Scale
Technique		1	"CD"
Formulation		16	11,12, ... ,25,26
Buffer/Sample		2	"B","S"
Temperature		32	1,2, ... ,31,32
Wavelength	nm	71	190.,191., ... ,259.,260.

```
cd0 = cd;
```

A.1.2 Import ANS fluorescence data

See comments in the first subsection.

```
ans = scanFolder[
    importPath <> "shortTerm\\exported\\ANS\\",
    "*.xls",
    importPTI
];
```

```
ans = parse[ans, "fileLocation", "(\\d+)([SB])\\.xls", 2];
ans = renameDimension[ans, "parsedDim1", "Formulation", ""];
ans = renameDimension[ans, "parsedDim2", "Buffer/Sample", ""];
ans = makeNumericDims[ans, {"Formulation", "Temperature"}];
ans = dropSingletons[ans, {"fileLocation"}];
```

```
ans = merge[ans];
```

```
ans = addSingletons[ans, {newDimension["Technique", "", "ANS Fl."]}];
```

```
dimInfo[ans]
```

Name	Unit	Length	Scale
Technique		1	"ANS Fl."
Formulation		16	11,12, ... ,25,26
Buffer/Sample		2	"B","S"
Temperature		32	1,2, ... ,31,32
Wavelength		201	399.95,400.95, ... ,598.95,599.95

```
ans0 = ans;
```

A.1.3 Import intrinsic fluorescence and light scattering data

See comments in the first subsection.

```
ils = scanFolder[
    importPath <> "shortTerm\\exported\\Intrinsic, Light Scattering\\",
    "*.xls",
    importPTI
];
```

```
ils = parse[ils, "fileLocation", "(\\d+)([SB])([IL])\\.xls", 3];
ils = renameDimension[ils, "parsedDim1", "Formulation"];
ils = renameDimension[ils, "parsedDim2", "Buffer/Sample"];
ils = makeNumericDims[ils, {"Formulation", "Temperature"}];
ils = dropSingletons[ils, {"fileLocation"}];
```

```
intr = select[ils, "parsedDim3", "I"];
intr = dropSingletons[intr, {"parsedDim3"}];
intr = merge[intr];
intr = select[intr, "Wavelength", #>=305&];
intr = addSingletons[
    intr,
    {newDimension["Technique", "", "Intr. Fl."]}
];
```

```
ls = select[ils, "parsedDim3", "L"];
ls = dropSingletons[ls, {"parsedDim3"}];
```

```
ls = merge[ls];
ls = select[ls, "Wavelength", #<=300&];
ls = addSingletons[
    ls,
    {newDimension["Technique", "", "LS"]}
];
```

```
dimInfo[ls]
```

Name	Unit	Length	Scale
Technique		1	"LS"
Formulation		16	11,12, ... ,25,26
Buffer/Sample		2	"B","S"
Temperature		32	1,2, ... ,31,32
Wavelength		11	290.,291., ... ,299.,300.

```
intr0 = intr;
ls0 = ls;
```

A.1.4 Import DSC data

Note: Even though DSC is short term data, it's in the long term data folder because it was measured by our collaborators in Germany.

```
rowRange = {2,17};
colRange = {3,3};
rowHeaderPos = 2;
colHeaderPos = 1;
```

```
dsc = importExcel[
    importPath <> "longTerm\\all.xls",
    rowRange, colRange, rowHeaderPos, colHeaderPos
];
dsc = renameDimension[dsc, "row", "Formulation"];
dsc = renameDimension[dsc, "column", "Measurement"];
```

Get the formulation numbers

```
dsc = operateDimension[
    ToExpression[StringTake[#, -3]]&,
```

```
    dsc,
    "Formulation"
];
```

```
dsc0 = dsc;
```

A.2 Import long term data

A.2.1 Import formulation parameters

```
rowRange = {2,17};
colRange = {4,8};
rowHeaderPos = 2;
colHeaderPos = 1;
```

```
formulations = importExcel[
    importPath <> "longTerm\\all.xls",
    rowRange, colRange, rowHeaderPos, colHeaderPos
];

formulations = renameDimension[formulations, "row", "Formulation"];
formulations = renameDimension[formulations, "column", "Parameter"];
```

Get the formulation numbers

```
formulations = operateDimension[
    ToExpression[StringTake[#, -3]]&,
    formulations,
    "Formulation"
];
```

```
dimInfo[formulations]
```

Name	Unit	Length	Scale
Formulation		16	11,12, ... ,25,26
Parameter		5	"Buffer","pH","Buffer concentration (mM)","Tween 80 (%)","HP-Beta-CD (%)"

A.2.2 Export table of formulation parameters

```
tab = elementFunction[Style[#, FontFamily->"Times"]&, formulations];
tab = operate[
    table,
    tab,
    {"Formulation", "Parameter"}
];
```

```
p = data[tab];
exportPlot["formulations", p];
```

A.2.3 Import stability measurements

```
rowRange = {2,17};
colRange = {9,24};
rowHeaderPos = 2;
colHeaderPos = 1;
```

```
longTerm = importExcel[
    importPath <> "longTerm\\all.xls",
    rowRange, colRange, rowHeaderPos, colHeaderPos
];

longTerm = renameDimension[longTerm, "row", "Formulation"];
longTerm = renameDimension[longTerm, "column", "Measurement"];
```

Get the formulation numbers

```
longTerm = operateDimension[
    ToExpression[StringTake[#, -3]]&,
```

```
    longTerm,
    "Formulation"
];
```

```
dimInfo[longTerm]
```

Name	Unit	Length	Scale
Formulation		16	11,12, ... ,25,26
Measurement		16	"Chemical Stability (%)","PC, 20 mo, >01um ", ... ,"MLRC, 03 mo","Turbidity, 03 mo"

```
longTerm0 = longTerm;
```

A.2.4 Export table of measurements

```
tab = elementFunction[
    Style[niceNumber[#], FontFamily->"Times",10]&,
    longTerm
];
tab = operate[
    table[#, False, False, True]&,
    tab,
    {"Formulation","Measurement"}
];
```

```
(* browse[tab] *)
```

```
p = data[tab];
exportPlot["longTermMeas", p];
```

A.3 Process short term data

A.3.1 Correct temperature dimension scale and add units to scales

Note: right after merge[], the temperature scale is guaranteed to be ordered 1-32, and buffer/sample is guaranteed to be ordered A-B.

```
temperatureScale = Range[10, 87.5, 2.5];
```

```
arrays = {ans0, intr0, ls0, cd0};
arrays = replaceScale[arrays, "Temperature", temperatureScale];
arrays = renameDimension[arrays, "Temperature", "Temperature", "C"];
arrays = renameDimension[arrays, "Wavelength", "Wavelength", "nm"];
{ans, intr, ls, cd} = arrays;
shortTerm = arrays;
```

A.3.2 Subtract the buffer

The buffer spectra have a signal that we don't understand. It could be protein contaminate.

Plot the buffer and sample spectra side by side

```
p = {ans, intr, ls, cd};
p = operate[arrayPlot, p, {"Wavelength","Temperature"}];
p = merge[p];
p = operate[table[#, False, True]&, p, {"Technique", "Buffer/Sample"}];
```

```
(* browse[p] *)
```

Export plot of buffer/sample comparison for a single formulation

```
p = select[p, "Formulation", 11];
p = dropSingletons[p, {"Formulation"}];
p = data[p];
```

```
(* Export the plot *)
exportPlot["compareBufferSample", p];
```

I'm going to subtract the buffer, so that the claim can't be made that the prediction is directly influenced by the buffer formulation. The information to predict from needs to be coming strictly from the protein's state.

```
arrays = {ans, intr, ls, cd};
arrays = subtractPosition[arrays, "Buffer/Sample", "B"];
arrays = dropSingletons[arrays, {"Buffer/Sample"}];
```

```
{ans, intr, ls, cd} = arrays;
```

A.3.3 Make sure formulation dimensions are the same, short term vs. long term

This is to check that they have the same formulations. They are certainly sorted, because of the use of merge during the imports.

```
x1 = scale[dimension[longTerm, "Formulation"]]
x2 = scale[dimension[shortTerm[[1]], "Formulation"]]
x1==x2
```

```
{11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26}
```

```
{11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26}
```

```
True
```

```
p = {ans, intr, ls, cd};
p = operate[arrayPlot, p, {"Wavelength","Temperature"}];
p = merge[p];
```

```
(* browse[p] *)
```

A.3.4 For ANS fluorescence, remove an instrument artifact where the recorded spectra goes to zero at some temperatures.

```
trigger = 2;
ans = operate[
    removeJumps[#, trigger]&,
    ans,
    {"Temperature", "Wavelength"}
];
```

A.3.5 Filter the thermal denaturation spectral melts

```
tr = 3;   (* temperature radius *)
```

```
wr = 4;   (* wavelength radius *)
```

```
cdFiltered = filterMelt[cd, tr, wr];
```

```
ansFiltered = filterMelt[ans, tr, wr];
```

```
intrFiltered = filterMelt[intr, tr, wr];
```

```
lsFiltered = filterMelt[ls, tr, wr];
```

A.3.6 Select regions of spectra that contain peaks and are not noisy

```
cdFiltered = select[cdFiltered, "Wavelength", # >= 205&];
```

```
intrFiltered = select[intrFiltered, "Wavelength", # >= 310&];
```

```
lsFiltered = select[lsFiltered, "Wavelength", # <= 300&];
```

A.3.7 Get the spectra of the native form

The spectra are very noisy, so will be averaging over low temperatures. The protein denatures after about 40C.

```
cdSpectrum = select[cdFiltered, "Temperature", #<=35&];
cdSpectrum = operate[mean, cdSpectrum, {"Temperature"}];
```

```
ansSpectrum = select[ansFiltered, "Temperature", #<=35&];
ansSpectrum = operate[mean, ansSpectrum, {"Temperature"}];
```

The ANS spectrum for formulation 19 appears to be flipped at low temperatures. We don't know why. Flip it back.

```
ansSpectrum = merge[
    {
        ansSpectrum,
        -select[ansSpectrum, "Formulation", 19]
```

```
    }
];
```

```
intrSpectrum = select[intrFiltered, "Temperature", #<=35&];
intrSpectrum = operate[mean, intrSpectrum, {"Temperature"}];
```

```
lsSpectrum = operate[mean, lsFiltered, {"Temperature"}];
```

A.3.8 Get the melt curves, using the mean of the spectra over wavelength.

```
cdMelt = select[cdFiltered, "Wavelength", #>=210 && #<=225&];
cdMelt = operate[mean, cdMelt, {"Wavelength"}];
```

```
ansMelt = operate[mean, ansFiltered, {"Wavelength"}];
```

```
intrMelt = operate[mean, intrFiltered, {"Wavelength"}];
```

For intrinsic fluorescence, we could also get the peak position. It won't be used, though, since the results look almost the same as intrMelt.

intrPeakPositionMelt = operate[centerOfMass, intrFiltered, {"Wavelength"}];

```
lsMelt = operate[mean, lsFiltered, {"Wavelength"}];
```

```
melts = merge[{cdMelt, ansMelt, intrMelt, lsMelt}];
```

A.4 Get transition temperatures

A.4.1 Function to get transition temperatures

```
getTransitionT[
    melt_dartArray, minTransitionTemp_, triggerSTDEV_, peakPosNeg_:-1
] :=
Module[{x, ts, minT, tx, sgn,stdev,i,j,tmidpoint,tonset,tendset,out},
    (* Get the 3rd derivative.  *)
    (* This makes transitions look like a large negative peak with small
```

```
    positive peaks on either side *)
x = sgFilter[melt, "Temperature", 3, 5, 3];

(* Drop the low temp data, since it's messed up by the FIR filter. *)
x = select[x, "Temperature", #>=minTransitionTemp&];

(* Interpolate to .5C increment *)
minT = scale[dimension[x, "Temperature"]][[1]];
x = interpolate[x, minT, .5];

(* Get the raw array and scale positions *)
ts = scale[dimension[x, "Temperature"]];
x = data[x];

(* Make the largest peak always be positive *)
x = x * peakPosNeg;

(* Get the first peak in x that has magnitude > trigger *)
stdev = StandardDeviation[x];
For[i=1, i<= Length[x]-1, i++,
    If[x[[i]] > triggerSTDEV * stdev && x[[i+1]] < x[[i]], Break[]];
];
tmidpoint = ts[[i]];

(* Get the negative peak just before the positive peak *)
For[j=i, j>2, j--,
    If[x[[j-1]] > x[[j]], Break[]];
];
tonset = ts[[j]];

(* Get the negative peak just after the positive peak *)
For[j=i, j<Length[x]-1, j++
    If[x[[j+1]] > x[[j]], Break[]];
];
tendset = ts[[j]];

out = newArray[
        {
            newDimension[
                "Transition type",
                "C",
                {"Onset", "Midpoint", "Endset"}
            ]
        },
        {tonset, tmidpoint, tendset}
```

```
    ];

    Return[out];
];
```

Taking the third derivative of a melt makes peaks corresponding to the transition onset, midpoint and endset

```
x = addSingletons[intrMelt, {newDimension["Number of derivatives", "", 0]}];
x = operate[interpolate[#, 10, .5] &, x, "Temperature"];
```

```
ders = {x};
For[i=1, i<=3, i++,
    xd  = sgFilter[x, "Temperature", 3, 5, i];
    xd = replaceScale[xd, "Number of derivatives", {i}];
    ders = Append[ders, xd];
];

p = merge[ders];
```

A.4.2 Export plot of melt derivatives for a single formulation

```
p = operate[listPlot[#, ImageSize->half]&, p, "Temperature"];
p = operate[table[#, True]&, p, "Number of derivatives"];
```

```
p = select[p, "Formulation", #==13 || #==14&];
p = operate[table, p, "Formulation"];
```

```
(* browse[p] *)
```

```
p = dropSingletons[p, {"Technique"}];
p = data[p];
```

```
(* Set figuresPath, a variable used in plotExport.m *)
figuresPath = NotebookDirectory[] <> "figures\\";
exportPlot["meltDerivatives", p];
```

A.4.3 Get the transition temperatures

```
discardBelowT = 30;
trigger = .4;
```

```
cdTT = operate[
    getTransitionT[#, discardBelowT, trigger]&,
    cdMelt, "Temperature"
];
```

```
ansTT = operate[
    getTransitionT[#, discardBelowT, trigger]&,
    ansMelt, "Temperature"
];
```

```
intrTT = operate[
    getTransitionT[#, discardBelowT, trigger]&,
    intrMelt, "Temperature"
];
```

```
lsTT = operate[
    getTransitionT[#, discardBelowT, .3]&,
    lsMelt, "Temperature"
];
```

Join them into 1 array

```
tts = merge[{cdTT, ansTT, intrTT, lsTT}];
```

Round the transition temperatures to integer values

```
tts = dataFunction[Round, tts];
```

A.4.4 Make a table of results and export

```
tab = elementFunction[Style[#, FontFamily->"Times"]&, tts];
tab = operate[table, tab, {"Formulation", "Technique"}];
```

```
tab = operate[table, tab, {"Transition type"}];
```

```
(* browse[tab] *)
```

```
p = data[tab];
exportPlot["transitionTs", p];
```

A.5 Process long term data

A.5.1 Get log particle counts from the long term data

The particle counts vary over several orders of magnitude, so the log particle count is more appropriate for linear prediction.

```
log = select[longTerm0, "Measurement", StringCount[#, "PC"] > 0 &];
others = select[longTerm0, "Measurement", StringCount[#, "PC"] == 0 &];
log = elementFunction[If[# > 0, Log[10, #], 0]&, log];
log = operateDimension["Log " <> #&, log, {"Measurement"}];
longTerm = merge[{others, log}];
```

A.5.2 Get mean of some of the long term measurements

We will get the mean of some of the measurement types because the separate measurements can't be predicted well. I think the data is probably noisy. It didn't come with error bars.

Get mean of log particle counts.

```
m = select[longTerm, "Measurement", StringCount[#, "PC"]>0&];
m = operate[mean, m, "Measurement"];
m = addSingletons[
    m,
    {newDimension["Measurement", "", {"Log PC, mean"}]}
];
longTerm = merge[{longTerm, m}];
```

Get mean of MLRC's

```
m = select[longTerm, "Measurement", StringCount[#, "MLRC"]>0&];
```

```
m = operate[mean, m, "Measurement"];
m = addSingletons[m, {newDimension["Measurement", "", {"MLRC, mean"}]}];
longTerm = merge[{longTerm, m}];
```

Get mean of turbidities

```
m = select[longTerm, "Measurement", StringCount[#, "Turbidity"]>0&];
m = operate[mean, m, "Measurement"];
m = addSingletons[
    m,
    {newDimension["Measurement", "", {"Turbidity, mean"}]}
];
longTerm = merge[{longTerm, m}];
```

A.6 Make datasets from which to predict stability

A.6.1 Function to combine arrays into a single 2 dimensional array

```
combineSpectra[arrays_, predictorName_] := Module[{x},
    (* In each array, combine dimensions other then "Formulation"
    into a single dimension named "Measurement". *)
    x = fuseOtherDimensions[
        arrays,
        {"Formulation"}, "Measurement"
    ];

    (* Divide each technique's matrix by its Frobenius norm *)
    x = operate[
        #/Norm[Flatten[#]]&,
        x,
        {"Formulation", "Measurement"}
    ];

    (* Combine the arrays by concatenating them
    along the "Measurement" dimension *)
    x = concatenate[x, "Measurement"];

    (* Label the data *)
    x = addSingletons[
        x,
```

```
        {newDimension["Predictor", "", predictorName]}
    ];

    Return[x];
];
```

A.6.2 Set up prediction from spectra at low temperature

```
fromSpectra = combineSpectra[
    {cdSpectrum, intrSpectrum, ansSpectrum},
    "Low T spectra"
];
dimInfo[fromSpectra]
```

Name	Unit	Length	Scale
Predictor		1	"Low T spectra"
Measurement		347	1,2, ... ,346,347
Formulation		16	11,12, ... ,25,26

A.6.3 Set up prediction from melt curves

```
fromMelts = combineSpectra[
    {cdMelt, intrMelt, ansMelt, lsMelt},
    "Melts"
];
```

A.6.4 Set up prediction from transition temperatures

In this dataset, standardize as usual.

```
fromTTs = combineSpectra[
    {tts},
    "Transition T's"
];
```

A.6.5 Set up prediction from transition regions (TR's)

Interpolate the spectra along the temperature dimension

```
startT = 10;
incrementT = .5;
```

```
spectra = {cdFiltered, ansFiltered, intrFiltered, lsFiltered};
spectra = operate[
    interpolate[#, startT, incrementT]&,
    spectra,
    {"Temperature"}
];
```

The dimensions of all of these arrays are Wavelength x Temperature x Formulation, and only the wavelength dimension varies between arrays.

```
dimInfo[spectra[[1]]]
```

Name	Unit	Length	Scale
Technique		1	"CD"
Formulation		16	11,12, ... ,25,26
Wavelength	nm	56	205.,206., ... ,259.,260.
Temperature	C	156	10.,10.5, ... ,87.,87.5

```
dimInfo[tts]
```

Name	Unit	Length	Scale
Technique		4	"ANS Fl.","CD","Intr. Fl.","LS"
Formulation		16	11,12, ... ,25,26
Transition type	C	3	"Endset","Midpoint","Onset"

Pick out the transition regions using transition temperatures

```
getRegions[spectra_dartArray, temperatures_dartArray, range_] :=
Module[
    {
        formulation, transitionT, minT, maxT, incrementT,
```

```
        newTempScale, out
    },
    formulation = position[dimension[spectra, "Formulation"]];

    transitionT = select[temperatures, "Formulation", formulation];
    transitionT = data[transitionT][[1]];

    {minT, maxT} = transitionT + range;
    incrementT = startStopIncrement[dimension[spectra, "Temperature"]][[3]];
    newTempScale = Range[range[[1]], range[[2]], incrementT];

    out = select[spectra, "Temperature", # >= minT&];
    maxT = scale[dimension[out, "Temperature"]][[Length[newTempScale]]];
    out = select[out, "Temperature", # <= maxT&];
    out = replaceScale[out, "Temperature", newTempScale];

    Return[out];
];
```

```
transitionTs = select[tts, "Transition type", "Midpoint"];
transitionTs = dropSingletons[transitionTs, "Transition type"];
transitionTs = operate[mean, transitionTs, "Technique"];
```

```
range = {-10, 10};
TRs = operate[
    getRegions[#, transitionTs, range]&,
    spectra, {"Temperature", "Wavelength", "Formulation"},
    threadDimensions->{"Formulation"}
];
```

Export a plot of the transition regions

```
p = operate[arrayPlot, TRs, {"Wavelength","Temperature"}];
p = merge[p];
p = select[p, "Formulation", {11,12,13}];
p = operate[table[#, False, True]&, p, {"Technique", "Formulation"}];
```

```
(* browse[p] *)
```

```
exportPlot["transitionRegions", data[p]];
```

Make predictor dataset

```
fromTransitionRegions = combineSpectra[
    TRs,
    "Trans. Regions"
];
```

A.6.6 Set up prediction from all the techniques

```
fromAll =
    {
        fromSpectra, fromMelts, fromTTs,
        fromTransitionRegions
    };
fromAll = dropSingletons[fromAll, {"Predictor"}];
fromAll = combineSpectra[fromAll, "All predictors"];
```

A.7 Leave-one-out cross validation

A.7.1 Make lists of prediction methods and predictors

```
predictionMethods = {{leastSquares, "LSQ"}, {rbfn, "RBFN"}};
```

```
predictors = {
    fromSpectra, fromMelts,
    fromTTs, fromTransitionRegions,fromAll
};
```

A.7.2 Do cross validation, for each predictor and each prediction method

```
t0 = AbsoluteTime[];
allResults = {};
forEach[method, predictionMethods,
    {methodFunction, methodName} = method;
    forEach[predictor, predictors,
```

```
        (* Thread cross validation over the data,
        even though the only dimension in the dataset other than Formulation
        Measurement is Predictor, and it's singleton for each array.  Effect
        operate[] is only used here in order to ignore the singleton dimensi
        testResult = operate[
            leaveOneOut[#, longTerm, methodFunction]&,
            predictor,
            {"Formulation", "Measurement"}
        ];

        (* Add a label for the prediction method *)
        testResult = addSingletons[
            testResult,
            {newDimension["Method", "", methodName]}
        ];

        allResults = Append[allResults, testResult];
    ];
];
t1 = AbsoluteTime[];
Print[t1-t0, "s"];
```

```
334.1089036s
```

A.7.3 Merge cross validation results

```
allResults = merge[allResults];
```

```
dimInfo[allResults]
```

Name	Unit	Length	Scale
Method		2	"LSQ","RBFN"
Predictor		5	"All predictors","Low T spectra","Melts","Transition T's","Trans. Regions"
Formulation		16	11,12, ... ,25,26
original/predicted		2	"Leave one out prediction","Observed"
Measurement		19	"Chemical Stability (%)","Log PC, 03 mo, >01um", ... ,"Turbidity, 20 mo","Turbidity, mean"

A.7.4 Get the likelihood of each cross validation run, using the Pearson correlation coefficient permutation test

```
pccSigma = operate[
    correlationSigma[#, 5000]&,
    allResults,
    {"Formulation", "original/predicted"}
];
```

Round to 2 digits of precision

```
pccSigma = dataFunction[Round[#, .1]&, pccSigma];
```

A.8 Plot results

A.8.1 Plot a histogram of the Pearson correlation coefficients for a prediction

```
x = select[allResults,
    "Method", "LSQ",
```

```
    "Predictor", "Melts",
    "Measurement", "Chemical Stability (%)"
];
```

```
p = data[correlationSigma[x, 5000, True]];
sig = data[correlationSigma[x, 5000, False]];
sig = Round[sig, .1];
```

```
p=Show[p,
    Frame->True,
    ImageSize->half,
    FrameStyle->Directive[FontFamily->"Arial", fontSize, thickLine],
    FrameLabel->{"Pearson correlation coefficient ("
        <> ToString[sig] <> " sigma)","Count"
    }
];
```

```
exportPlot["pccHistogram", p];
```

A.8.2 Make a table of the likelihoods, highlighting likelihoods > 3 sigma

```
tab = elementFunction[
    Style[
        ToString[#],
        FontFamily->"Times",
        Background-> If[#≥3,
            Green
        ,
            If[#≥2,
                Yellow
            ,
                White
            ]
        ]
    ]&,
    pccSigma
];
```

```
tab = operate[
    table,
```

```
    tab,
    {"Measurement", "Predictor"}
];
```

```
(* browse[tab] *)
```

Export tables of likelihoods

```
p = data[select[tab, "Method", "LSQ"]][[1]];
exportPlot["LSLikelihoods", p];

p = data[select[tab, "Method", "RBFN"]][[1]];
exportPlot["RBFNLikelihoods", p];
```

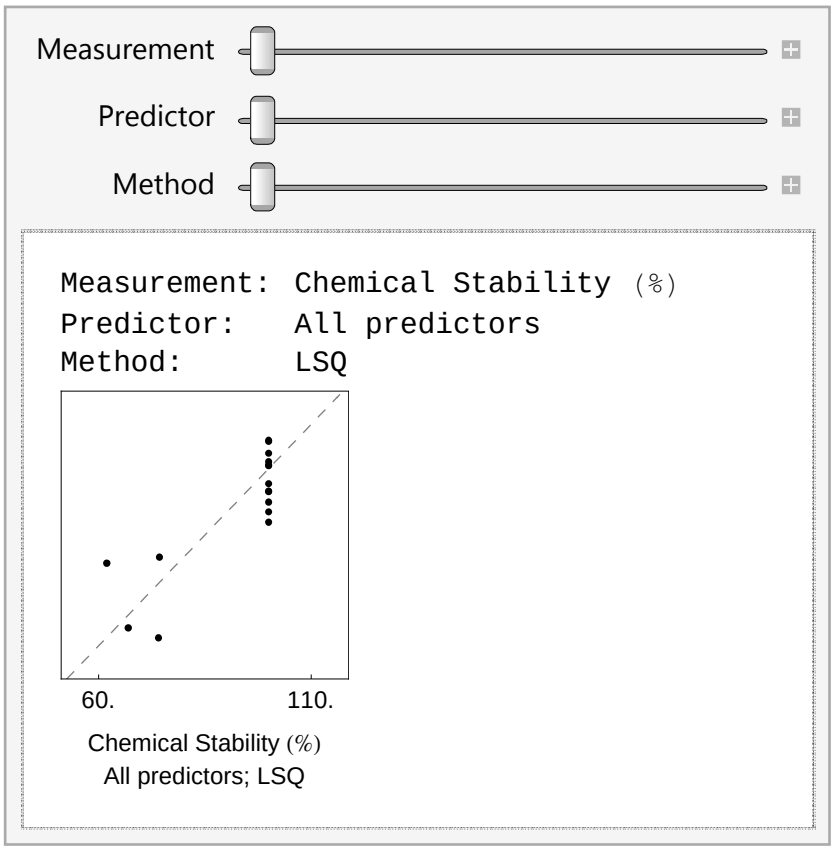
A.8.3 Make correlation plots

```
makePlotLabel[array_] :=
    position[dimension[array, "Measurement"]] <> "\n" <>
    position[dimension[array, "Predictor"]] <> "; " <>
    position[dimension[array, "Method"]]
```

```
threadDims = {"Measurement","Predictor","Method"};
```

```
plots = operate[
    plotCorrelation[#, makePlotLabel[#], None]&,
    allResults,
    Join[{"original/predicted", "Formulation"}, threadDims],
    threadDimensions->threadDims
];
```

```
browse[plots]
```



A.8.4 Export correlation plots of best 16 fits

Make sure that the dimensions and dimension scale positions are in the same order in the array of plots and the array of correlation significances. This only needs to be done since we're taking raw data from the arrays.

```
p = merge[plots];  (* This sorts the dimension scale positions *)
p = transpose[p, {"Measurement", "Predictor", "Method"}];

c = merge[pccSigma];
c = transpose[c, {"Measurement", "Predictor", "Method"}];
```

```
p = Flatten[data[p]];
c = Flatten[data[c]];
```

```
ord = Reverse[Ordering[c]];
c = c[[ord]];
```

```
p = p[[ord]];
p = Take[p, 16];
cplots = p;
```

Export correlation plots

```
For[i=1, i<=Length[cplots], i++,
    p = cplots[[i]];
    letter = CharacterRange["A", "Z"][[i]];
    p = Show[p, ImageSize->72(*points/inch*)*1.35(*inches*),
        FrameStyle->Directive[FontFamily->"Arial", 9, thickLine]
    ];
    p = inset[p, letter, {.08, .92}, Black];
    exportPlot["bestFit" <> letter, p];
];
```

Appendix B

Functions used in Appendix A

B.1 Functions for performing regression analysis

B.1.1 Least squares fit

```
leastSquares[fromIn_, toIn_] := Module[
    {fromMean, from, scale, rescale, toMean, to,t, function, n},

    (* Get the mean row of the "from" training dataset, and
        subtract it from the training and test data *)
    fromMean = Mean[fromIn];   (* A row *)
    from = Map[#-fromMean&, fromIn];

    (* Get the mean value of the "to" training dataset,
        and subtract it from the training and test data *)
    toMean = Mean[toIn];
    (* toMean is either a row or a numerical value, depending on whether
        toIn is a matrix or column vector *)
    to = Map[#-toMean&, toIn];

    (* from.t = to *)
    t = PseudoInverse[from].to;
    function = With[
        {a = fromMean, b = toMean, c = t},
        (#-a).c + b&
    ];
    Return[function];
];
```

Arguments of leastSquares[] function

fromIn is a matrix with dimensions
    (sample x input observable)
toIn is a matrix with dimensions
    (sample x output observable)
    or a column vector of dimension (sample)
leastSquares[] returns a function that transforms input vectors to output vectors.

B.1.2 Radial basis function network

This function externally works the same as the leastSquares[] function above. I.e. it has the same arguments and the same type of return value.

```
rbfn[fromIn_, toIn_, width_:1] := Module[
    {fromMean, toMean, from, scale, rescale, exp,
        expMean, to, diffs, n, transform, function},

    (* Center columns around their mean *)
    fromMean = Mean[fromIn];
    from = N[Map[#-fromMean&, fromIn]];

    (* Rescale columns to have RMS 1 *)
    scale = Map[StandardDeviation, Transpose[fromIn]];
    rescale = 1/scale;
    from = Map[#*rescale&, from];

    (* Center the measurements to predict around their mean *)
    toMean = Mean[toIn];
    to = Map[#-toMean&, toIn];

    (* Get gaussian similarity matrix *)
    diffs =  Table[
        Norm[from[[i]]-from[[j]]],
        {i,1,Length[from]},
        {j,1,Length[from]}
    ];
    n = Length[from[[1]]];
    (* The rows in "from" have stdev 1.
        Need to divide overlaps by n. *)
    exp = Exp[-diffs^2/(n width^2)];

    (* Solving: from.t = to *)
    transform = PseudoInverse[exp].to;
```

```
    function = With[
        {a=fromMean,b=rescale,c=from,d=width,e=transform,f=toMean},
        rbfnPredict[#,a,b,c,d,e,f]&
    ];
    Return[function];
];
```

The next function, rbfnPredict[] , is used inside rbfn[] to make the function that rbfn[] returns.

```
rbfnPredict[
    from_, fromMean_, rescale_, refPoints_,
    width_, transform_, toMean_
] :=
Module[
    {rescaled,diffs,n,exp,out},
    rescaled = (from-fromMean) * rescale;
    diffs =  Table[
        Norm[rescaled-refPoints[[j]]],
        {j,1,Length[refPoints]}
    ];
    n = Length[from];
    exp = Exp[-diffs^2/(n width^2)];
    out = exp.transform + toMean;
    Return[out];
];
```

B.1.3 Function to perform leave one out cross validation

```
leaveOneOut[fromIn_dartArray, toIn_dartArray, predictionFunction_] :=
Module[
    {
        sampleDim, obsDim, dimOrder, from, to, inOut, i,
        fromTrain, toTrain, fromTest, toTest, meas,
        measToTrain, measToTest, f, out
    },
    (* Get the dimension scales.
    There should be just two in each array. *)
    sampleDim = otherDimensions[fromIn, "Measurement"][[1]];
    obsDim = dimension[toIn, "Measurement"];

    (* Transpose so that samples are associated to the
        left matrix dimension, measurements to the right *)
    dimOrder = {sampleDim, obsDim};
```

```
    from = transpose[fromIn, dimOrder];
    to = transpose[toIn, dimOrder];

    (* Get the data *)
    from = N[data[from]];
    to = N[data[to]];

    (* Do prediction *)
    inOut = {};
    For[i=1, i<=Length[from], i++,
        fromTrain = Drop[from, {i}];
        toTrain = Drop[to, {i}];

        fromTest = from[[i]];
        toTest = to[[i]];

        f = predictionFunction[fromTrain, toTrain];
        inOut = Append[inOut, {toTest, f[fromTest]}];
    ];

    out = newArray[
                {
                    sampleDim,
                    newDimension[
                        "original/predicted",
                        "",
                        {"Observed", "Leave one out prediction"}
                    ],
                    obsDim
                },
                inOut
            ];
    Return[out];
];
```

B.2 Pearson correlation coefficient permutation test

B.2.1 A correlation function that avoids dividing by zero

```
correlation[vec1_, vec2_] := Module[{v1,v2,n1,n2},
    v1 = vec1 - Mean[vec1];
    v2 = vec2 - Mean[vec2];
```

```
    n1 = Norm[v1];
    n2 = Norm[v2];

    (* Return zero correlation if one of the vectors has zero variance. *)
    (* This is to avoid dividing by zero.  *)
    (* Would use Mathematica's Correlation[] if it weren't for this.  *)
    If[n1<10^-10 || n2<10^-10, Return[0]];
    Return[v1.v2/(n1 n2)];
];
```

B.2.2 Pearson correlation coefficient permutation test for two lists of numbers

```
correlationSigma[
    predicted_List, original_List,
    nPermutations_, returnPlot_:False
] :=
Module[
    {permutedCorrelations,pccStd,pccObs,plotRange,out},
    permutedCorrelations = Table[
        correlation[RandomSample[predicted], original],
        {nPermutations}
    ];
    pccStd = StandardDeviation[permutedCorrelations];
    pccObs = Correlation[predicted, original];
    plotRange = Max[{pccStd,pccObs}]*1.3;
    If[returnPlot,
        out = Histogram[
            permutedCorrelations,
            PlotRange->{{Automatic, plotRange}, {All,All}}
        ];
        out = Show[out,
            Epilog->{
                Thickness[Medium], Dashed, Red,
                Line[{{pccObs, 0}, {pccObs, 5000}}]
            }
        ];
    ,
        out = If[pccStd == 0, 0, pccObs/pccStd];
    ];
    Return[out];
];
```

B.2.3 Pearson correlation coefficient permutation test for a DART array containing two lists of numbers

```
correlationSigma[array_dartArray, nPermutations_, returnPlot_:False] :=
Module[{arr,dat,out},
    arr = transpose[array,
        Join[
            {"original/predicted"},
            otherDimensions[array, "original/predicted"]
        ]
    ];
    dat = data[arr];
    out = correlationSigma[
        Flatten[dat[[1]]],
        Flatten[dat[[2]]],
        nPermutations,
        returnPlot
    ];
    out = newArray[{}, out];
    Return[out];
];
```