Mechanistic Analysis and Quantification of Gastrointestinal Motility: Physiological Variability and Plasma Level Implications

by

Arjang Talattof

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Pharmaceutical Sciences)
in The University of Michigan
2015

Doctoral Committee:

Professor Gordon L. Amidon, Chair
Research Professor Gregory E. Amidon
Professor Kerby Shedden
Professor Duxin Sun

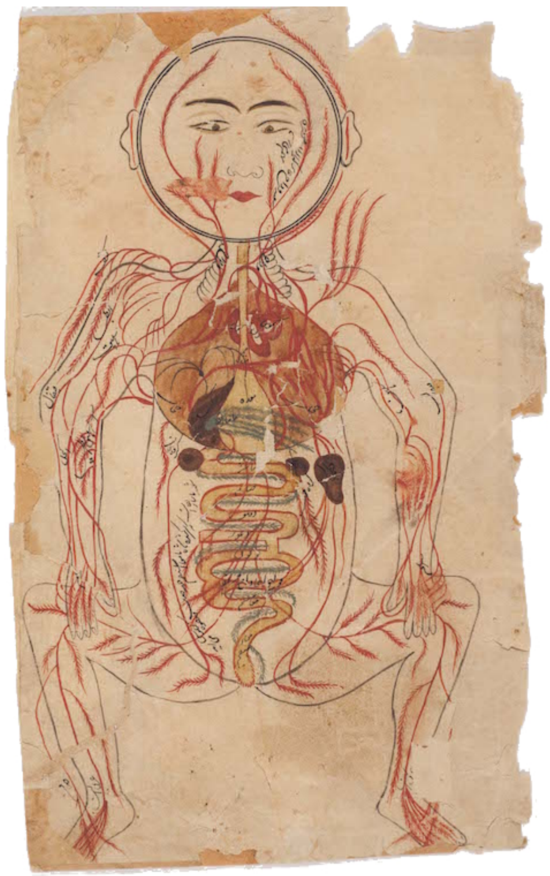


Illustration from a Persian treatise on human anatomy
“Tashrīh-ī Mansūrī (Mansūr’s Anatomy)”
Ibn Ilyās, Mansūr ibn Muhammad, fl. 1384.

“Scientific truth should be presented in different forms, and should be regarded as equally scientific, whether it appears in the robust form and the vivid colouring of a physical illustration, or in the tenuity and paleness of a symbolic expression.”
James Clerk Maxwell

“I can calculate the movements of stars, but not the madness of men.”
Isaac Newton

To my family, whose enduring and unconditional love has been a pillar of support throughout my life.

ACKNOWLEDGEMENTS

It is impossible to overstate my gratitude to all those who have made possible this project. I first wish to thank my advisor, Dr. Gordon L. Amidon. He has been a tremendous source of inspiration and support from the first day. He was always willing to set time aside out of his busy schedule (managed so gracefully by Ms. Gail Benninghoff) to discuss not just progress in my work but also more generally provide support and advice in all aspects of my graduate studies. His enthusiasm for my project drove me to work even harder. His insights, both scientifically and professionally, have helped shape me as a researcher, and his dedicated quest for knowledge is truly awe-inspiring.

I am very thankful to my committee members. Dr. Gregory E. Amidon, Dr. Kerby Shedden (Department of Statistics), and Dr. Duxin Sun all provided wonderful advice and insight and helped me expand upon various aspects of my projects both in committee sessions and individual meetings. Also instrumental in my success at the College of Pharmacy were Dr. Frank J. Ascione, Ms. Patrina T. Hardy, Dr. David E. Smith, and Dr. Steven P. Schwendeman who provided valuable advice and support during my graduate career. Fellow collaborators on the overall project were instrumental in development and progress, helping contextualize my work in the larger scheme. In particular, Jason Baker, Dr. James Brasseur, Ann Fiorito, Dr. William Hassler, Dr. Mark Koenigsknecht, Dr. Deanna Mudie, and Dr. Yasuhiro Tsume all contributed in various ways and am indebted for their time and support. I am appreciative of the Rackham Graduate School for grants to attend conferences as well as Dr. Gustavo R. Rosania for his support.

Judy Price provided the Mathematica template for the bioequivalence visualization plots in Chapter I. I would like to extend my appreciation to those who put together the LaTeX template used in writing my dissertation. I am also grateful to Dr. Robert Morris (NASA Ames Research Center) who spent time at an art gallery in San Jose, CA, discussing the application of Gaussian processes in his work on robotics and artificial intelligence. Through this interaction I realized the potential for new statistical methods used for signal analysis in Chapter III.

Much gratitude is due to past and present graduate students and researchers as well as members of the administrative and support staff of the College of Pharmacy. Funding for this project was provided by the University of Michigan's College of Pharmacy and the U.S. Food and Drug Administration (HHSF223201310144C).

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF FIGURES vii

LIST OF TABLES xii

LIST OF APPENDICES xiii

ABSTRACT xiv

CHAPTER

I. Gastrointestinal Motility Variation and Implications for Plasma Level Variations 1

1.1 Introduction 1
1.2 Methods 3
1.2.1 Fasted state GI transit variability 3
1.2.2 Developing a periodic function 4
1.2.3 Intra- and inter-patient variation and bioequivalence 5
1.2.4 Model validation 6
1.3 Results and Discussion 8
1.3.1 Dose time dependence and plasma levels 8
1.3.2 Example cases 8
1.3.3 Bioequivalence implications 10
1.4 Conclusion 13
1.5 Tables and figures 14
1.6 References 35

II. Fasted State Dissolution and Transit of Ibuprofen 39

2.1 Introduction 39
2.1.1 Physiological properties of the gastrointestinal tract 40

2.2 Methods . . . 42
2.2.1 Fluid and particle transit . . . 43
2.2.2 Disintegration and dissolution mass balance . . . 45
2.2.3 Ibuprofen and phenol red . . . 45
2.3 Results and discussion . . . 46
2.4 Conclusion . . . 48
2.5 Tables and figures . . . 50
2.6 References . . . 59

III. Methodology for Statistical Analysis of Antroduodenal Manometry Signals . . . 64

3.1 Introduction . . . 64
3.2 Methods . . . 66
3.2.1 Data acquisition . . . 66
3.2.2 Wavelet transformations . . . 67
3.2.3 Kernel density estimators . . . 68
3.2.4 Gaussian processes . . . 69
3.3 Results and discussion . . . 70
3.3.1 Peak detection . . . 70
3.3.2 Kernel smoothing . . . 71
3.3.3 Regression . . . 72
3.4 Conclusion . . . 73
3.5 Figures . . . 74
3.6 References . . . 81

IV. Conclusions . . . 84

4.1 Summary . . . 84
4.2 Future considerations . . . 85
4.2.1 Mass balance analysis . . . 85
4.2.2 Motility signal processing . . . 86
4.2.3 Physiological studies . . . 87
4.3 References . . . 89

APPENDICES . . . 91

LIST OF FIGURES

Figure

1.1 Plots illustrating the continuous gastric emptying rate function (top) and the phase-associated lag time function (bottom). 14

1.2 Experimentally determined[20] emptying rates (top) and lag times (bottom) at the begging of the three phases. 15

1.3 Schematic of the GI tract with a pulsatile, time-dependent emptying rate from the stomach compartment into the intestinal compartments. Intestinal transit rates are phase-shifted such that the pulse initiating in the stomach travels through the GI tract over a two-hour time period[8]. 15

1.4 Effect of dose time t_0. Left: early dosing, that corresponds to phase I, with a low gastric emptying rate and long lag time; the volumetric lag time dependence results in a considerable difference in the emptying and appearance in plasma. Right: late dosing in phase III where the gastric emptying rate has increased considerably and the lag time is nearly zero; there is a negligible difference between the 50 mL and 200 mL volumes since all gastric content is emptying rapidly and immediately. 16

1.5 Dependence of gastric emptying on volume and dose time (i.e. phase). 17

1.6 A comparison of simulated (green) and experimentally determined[20] (red) gastric emptying rates and lag times for 50 mL (left) and 200 mL (right) volumes. 18

1.7 Simulated emptying half-times t_{50} (green dots) compared to measured citepOberle1990 t_{50} of 50 mL (blue) and 200 mL (yellow) volumes. . 18

1.8 Left, the predicted gastric emptying results for a 120-minute range of dose time t_0 (green shaded region, mean prediction as dashed green line) with superimposed volumetric emptying data from subjects measured by MRI[25]. Emptying patterns show both first-order (center, 75%) and non-first order (right, 25%) corresponding to the distribution of simulated emptying patterns. 19

1.9 Accordance of experimental[36] and simulated gastric emptying and intestinal transit time ranges. The black diamonds represent the 95% CI about the means, the white bars within the colored regions are the medians; the colored regions represent the 25-75 percentiles; and the whiskers span the entire range. The difference between the means is not statistically significant (Mann-Whitney $p = 0.79$ and $p = 0.62$ for gastric emptying and intestinal transit times, respectively). . . . 19
1.10 Left: probability density of simulated (green) and experimental[36] intestinal transit times for solutions and dosage forms. Right: small intestinal transit flow using previously reported compartmental model [35] (red) and the current model (mean as dashed line; predicted range as green shaded region). 20
1.11 BCS Class I with high p_{eff} showing volumetric effect on plasma levels. 21
1.12 BCS Class III constant p_{eff} and volumetric effect on plasma levels. 21
1.13 BCS Class III location-dependent p_{eff} and volumetric effect on plasma levels. 22
1.14 Examples of predicting plasma level variation for short plasma elimination half-life compounds. 23
1.15 Simulated BCS Class I BE trials. The black horizontal bars represent the reference 80-125% range. The vertical bars are individual BE simulations with 24 virtual subjects each, indicating the C_{max} mean 90% CI. In the left column, the mean C_{max} is used while the median is considered in the right column. 24
1.16 Simulated BCS Class III BE trials using constant permeation. The vertical bars are individual BE simulations with 24 virtual subjects each, indicating the C_{max} mean 90% CI. In the left column, the mean C_{max} is used while the median is considered in the right column. . . 25
1.17 Simulated BCS Class III BE trials using location-dependent permeation. The vertical bars are individual BE simulations with 24 virtual subjects each, indicating the C_{max} mean 90% CI. In the left column, the mean C_{max} is used while the median is considered in the right column. 26
1.18 Variations in bioavailability for BCS Class I (a), BCS Class III with constant permeation (b), and BCS Class III with locational permeation (b). Bioavailability as a function of dose time t_0 (green) based on mean emptying rates and lag times compared to a population with varied emptying rates and lag times (dotted red) for each of the three simulations. 27
1.19 Bioavailability CV for BCS Class I (a) and BCS Class III with locational permeation (a) simulations with respect to plasma elimination half-life and parameter variation (as % of mean). 29
1.20 C_{max} CV for BCS Class I (a) and BCS Class III with locational permeation (b) simulations with respect to plasma elimination half-life and parameter variation (as % of mean). 30

1.21 Expected failure rate for BCS Class I (a) and BCS Class III with locational permeation (b) simulations with respect to plasma elimination half-life and parameter variation (as % of mean). 31
1.22 Simulated BCS Class I half-life dependence of T_{max} and C_{max} distributions for 50 mL and 200 mL volumes. Overlayed on the C_{max} plots are various distributions, highlighting the lack of normality following a log transformation. 32
1.23 Simulated BCS Class III half-life dependence of T_{max} and C_{max} distributions for 50 mL and 200 mL volumes using constant permeation. Overlayed on the C_{max} plots are various distributions, highlighting the lack of normality following a log transformation. 33
1.24 Simulated half-life dependence of T_{max} and C_{max} distributions for 50 mL and 200 mL volumes using location-dependent permeation. Overlayed on the C_{max} plots are various distributions, highlighting the lack of normality following a log transformation. 34
2.1 Incorporating dissolution mass balance into the GI transit system. A solid dose is administered with a liquid volume. Gastric emptying drives fluid and solid out of the stomach. Fluid is replenished via gastric secretion. Disintegration and dissolution take place in all compartments based on the physiological environment and drug physiochemical properties. 50
2.2 Visualizing gastric secretion rate relative to stomach volume and gastric emptying rate. 52
2.3 The effect of particle size on emptying rate. 53
2.4 Effective boundary layer thickness h_{eff} (Equation 2.2). 53
2.5 Ibuprofen dissolution under different conditions. 53
2.6 Reported[58–60] and calculated ibuprofen solubility. 54
2.7 Phenol red solution transit through GI tract when dosed during (a) early phase I; (b) phase II; (c) early phase III; and (d) late phase III. 55
2.8 Ibuprofen plasma profiles for early phase I (blue), phase II (red), early phase III (purple), and late phase III (green) showing the variation in C_{max} and especially T_{max}. 56
2.9 Ibuprofen particulate and solution transit through GI tract when dosed during (a) early phase I; (b) phase II; (c) early phase III; and (d) late phase III. 57
2.10 Predicted (green) and experimental (orange) values for C_{max} (top), T_{max} (middle), and AUC (bottom) for 200, 400, and 800mg doses of ibuprofen. The colored boxes represent the 25-75 percentiles, the whiskers span the entire range, and the fuchsia circles are outliers. . 58
3.1 Schematic of tube placement with motility channels in the stomach (4), duodenum (3), proximal jejunum (2), and distal jejunum (1). . 74
3.2 Manometry signal from a single duodenal channel. 74
3.3 Left: original manometry tracing from Figure 3.2. Center: Histogram of pressure (mmHg) occurrences in the signal. Right: Fourier transform of the signal in Figure 3.2 showing lack of identifiable periodicity. 75

3.4 CWT-based peak detection results for three different antroduodenal manometry channels. The detected peaks are shown as red dots overplayed on top of the signals. Horizontally adjacent are histogram plots showing detected peak densities along the time axis. 75
3.5 Temporal distribution of detected peaks (red ticks) represented as a histogram plot. 76
3.6 Each detected peak (red bar) is associated with a Gaussian basis function (gray curve). 76
3.7 Kernel density estimations using different smoothing bandwidths. . 77
3.8 Using different kernel functions to estimate peak density: quartic/biweight, cosine, Epanechnikov, Gaussian, triangular, and tri-weight . 77
3.9 The distribution of detected peaks is estimated (green curve) by summing the basis functions (and normalizing so the functions integrates to 1 as a proper densities). The red tick marks are the time positions of peaks and the grey histogram shows their temporal distribution. . 78
3.10 Prediction of average pressure function as determined by the Gaussian process regressions. Regions of low (black), medium (blue), and high (red) expected pressures are overplayed on the original signal (light blue). 78
3.11 Top: overlay of the original signal with identified regions corresponding to the different phases (I in green, II in blue, and III in red). The durations are labeled above. Center: a transition plot, with the detected peaks plotted in red below. Bottom: the kernel density estimation (light red) of detected peak times (red tick marks). . . . 79
3.12 Phase III pressure peaks are preceded by short spikes that are not seen in either phases I or II. Top row contains 40-minute spans of signals in phases I-III. Middle row shows the boxed regions from top row in greater detail. The black arrows correspond to pressure peaks in the respective regions while the red arrows indicate the pre-spikes. Bottow row is further enlargement of a 1-minute period that shows pre-spikes occurring only in phase III. 80
B.1 (A-B) Images adapted from[2] and (C)[18]. (D-F) The FFTs of the template images (G-H) Detected peaks in the Fourier domain (J-H) Reconstructions based on the detected peaks. 157
B.2 (A) Normalized Fourier domain histograms of Figure B.1C and (B) Figure B.4F. 158
B.3 (A) Image from Figure B.1B. (B) A circularly cropped. (C-D) B rotated 15° and 95°, respectively. (E-F) The detected peaks from the FFTs of images A-D. (I) The peaks that are maintained under rotation. (J) Reconstructions based on the maintained peaks 158

B.4 Detected regions of interest in images. (A), (B), and (C) The highlighted boxes show regions where peaks were detected in images of intestinal villi of treated mice. The red regions in each image are magnified in (D), (E), and (F). Detected points are shown in (G), (H), and (I). The detected points were used to reconstruct the images in (J), (K), and (L), in which regular patterns are seen. 159
B.5 (A) Contiguous regions were selected from figure B.4C, magnified in (B) and (C); the Fourier transforms are shown in (D) and (E), along with corresponding reconstructions showing hexagonal patterns highlighted in blue in (F) and (G), respectively. 160
B.6 (A) Selected region in a crystalloid object. (B) Inverse Fourier transform of the peaks in the FFT, superimposed on the same selected region as in A. (C) Repetitive pattern in the crystalloid object, with the unit cell indicated by purple arrows. (D) Quadrants in a hexagonal crystalloid were peaks were detected in the FFT. Superimposed are the enlarged unit cells of each quadrant. (E) Plot of the two shortest directions and the angle of separation. The colors of the circles correspond to the colored quadrants in D. The average measure for edge 1 was 21.99±0.07 nm, edge 2 11.14±0.15 nm, and the angle separating them 87.80±0.41 degrees. 161

LIST OF TABLES

Table

1.1 Mass balance parameters 14
1.2 Properties of example compounds with short plasma elimination half-lives used for simulation validations. 20
1.3 Variation in BCS Class I simulations 27
1.4 Variation in BCS Class III (cont. abs.) simulations 28
1.5 Variation in BCS Class III (loc. abs.) simulations 28
2.1 GI fluid component concentrations & properties, liquid volumes, and geometry during the fasted state[37] 51
2.2 Ibuprofen physiochemical properties 52
2.3 Phenol red physiochemical properties 52

LIST OF APPENDICES

Appendix

A. Sample Python Code for Motility, Dissolution, and Signal Analyses . . 92

B. A Machine Vision Approach to Visualize and Analyze Intracellular Crystalloid Objects in Transmission Electron Microscope Images . . . 146

ABSTRACT

Mechanistic Analysis and Quantification of Gastrointestinal Motility: Physiological Variability and Plasma Level Implications

by

Arjang Talattof

Chair: Gordon L. Amidon

The oral route of administration is still by far the most ubiquitous method of drug delivery. Development in this area still faces many challenges due to the complex inhomogeneity of the gastrointestinal environment. In particular, gastric emptying and gastrointestinal motility is not predictable and so dosing occurs randomly with respect to these physiological variables. The goal of this research is to present a mass balance analysis that captures this variation, highlighting the effects of motility and exploring how it ultimately impacts plasma levels and the relationship to bioequivalence.

A mechanistic analysis is first developed describing the underlying fasted state cyclical motility and how the contents of the gastrointestinal tract are propelled. This physiologically based approach allows the estimation of potential absorption ranges based on uncontrolled variation. Validation of the simulations is based on reported gastric emptying profiles and volumetric emptying as well as previous experimental works on gastrointestinal transit times, and the bioequivalence implications of such variation are also considered. Next, a dissolution model is presented to account for

the dynamics of physiological conditions along the gastrointestinal tract, including small volumes and variable pH profiles. Predicting the extent of dissolution along with transit profiles of dissolved and particulate content is crucial to approximating absorption. Ibuprofen and phenol red are used as example cases.

Finally, a method for refining the gastrointestinal transit model is critical for ensuring accuracy, and a methodology is presented for extracting relevant information from intubation studies. Gastrointestinal manometry can be thought of as a stochastic process in which the indeterminacy of state transition times belies absolute periodicity of the system. To account for this inherent randomness, the use of statistical computing can identify and characterize the different phases of the gastrointestinal cycle. Specifically, a Gaussian process is used as a robust regression method to model the time-dependent evolution of the signal. As further validation, using a pressure peak detection method based on continuous wavelet transforms and subsequently a kernel density estimator as a smoothing function, regression-based motility phase classification corresponds expected pressure peak density estimates.

CHAPTER I

Gastrointestinal Motility Variation and Implications for Plasma Level Variations

1.1 Introduction

The human stomach can be divided into three distinct muscular components, each contributing in part to emptying: the fundus (proximal stomach) that relaxes and receives content; the antrum (distal stomach) that grinds and titrates large matter into small particles; and the pylorus, a region of high pressure that prevents emptying of solids during antral contractions[1,2]. The small intestine, comprised of the duodenum, jejunum, and ileum, varies from 3.5-9.8m in length with an inner diameter of 2.5-3 cm[3,4]. Gastrointestinal (GI) motility and the associated migrating motor or myoelectric complex (MMC) play a crucial role in transporting ingested material from the stomach through the intestines and into the colon by means of segmental and peristaltic contractions[5]. The propagating wave of peristalsis is regulated by hormones, paracrine signaling, and the autonomous nervous system, while segmentation is carried out by longitudinal muscle relaxation and circular muscle contraction thereby mixing GI contents with digestive enzymes and ensuring composition uniformity and sufficient epithelial contact for absorption[6]. The gastric emptying rate is controlled by gastric distention promoting emptying and intestinal stimuli slowing emptying[7]. Contractile activity propels matter from the stomach into the small bowel where segmental contractions beginning in the duodenum reach the terminal ileum in approximately 2 hours[8]. The MMC is defined by three distinct phases: phase I is an inert period with little activity; phase II features sporadic contractions gradually ascending in magnitude but with little net forward movement of gastric contents; and phase III is characterized by powerful, high frequency contractile bursts that promote emptying of contents where peak flow rates are observed[9]. The lengths of the phases can vary greatly, phase I lasting between 20-90 minutes, phase II between 35-135 minutes, and phase III between 2-15 minutes[10]. As the contractile activity propagates it becomes less spatiotemporally organized resulting in slower propulsion rates in the distal small bowel[11]. It is thus important to acknowledge these physiological effects on oral drug products and their intestinal residence times, which can impact the extent of absorption in the fasted state. Such temporal variations must be taken into account to explain frequent and irregular plasma concentration profiles that cannot be accounted for using compartmental, first-order rate models[12,13]. Indeed, mechanistic approaches have been previously employed to help explain variable absorption and double-peak phenomena in pharmacokinetic profiles[14,15]. A compartment-based, continuously stirred reactor tank mass balance system is presented here to analyze high solubility compounds with both high and low permeability (BCS Classes I & III) and elucidate the effects of GI motility on drug plasma profile variations during the fasted state. Both the gastric emptying and the intestinal transit rates are time-dependent functions reflecting the various phases of GI motility.

1.2 Methods

1.2.1 Fasted state GI transit variability

The process of human gastric emptying and intestinal transit are traditionally represented by first-order approximations. For volumes of 240 to 800 mL, experimental measurements of gastric emptying half-time varied from 8 to 18 minutes[16–19]. Oberle et al. demonstrated the emptying half-time dependence on MMC phase for 50 and 200 mL volumes[20]. Various numbers of compartments have been employed to model the GI tract using constant transit rates: a two-compartment model with well-mixed tanks[21]; a single-compartment model used to account for dissolution rate-limited absorption[22]; a four-compartment model incorporating bile secretion effects [23]; and representing the intestinal tract from stomach to colon as nine consecutive compartments[24].

There are several physiological factors that cannot be captured using these classical approaches. Gastric emptying patterns are heavily dependent on the MMC phases[20]. Volumetric emptying from the stomach was shown to be non first-order, bi-phasic in 25% of subjects evaluated[25]. MMC propagation can also result in the orad transit of GI content[26–28] which, when coupled with the formation of small water volume packets along the intestinal tract (as opposed to a continuous region of fluid) [25], raises the issue of segmental, back and forth mixing of contents and its impact on GI transit times. Indeed, the small intestine water volumes themselves in fasted humans ranged from 80 to 300 mL[29–32]. The propagation of MMCs along the small bowel has profound effects on the intestinal residence times which vary greatly when measured experimentally[33].

To tackle some of these more complicated sources of variation, Higaki et al. presented a two-phase model incorporating lag time, demonstrating the value of a time-dependent absorption coefficient in prediction nonlinear properties of transit and absorption. Langguth et al. used a single compartment but with transit as a periodic step function corresponding to the MMC phases[34]. Herein, a multi-compartment approach is used based on previously described, phase-dependent gastric emptying rates [20], with a continuous function describing the transition between MMC phase states which propagate along the entire GI tract (i.e. across all compartments).

1.2.2 Developing a periodic function

The cyclical gastric emptying rate and lag time are described by a Fourier series approximation (Equation 1.1) and a sigmoidal decay function (Equation 1.2), respectively, illustrated in Figure 1.1. The parameters, described in Table 1.1, reflect the experimentally measured gastric emptying rates and delay times for 50 mL and 200 mL volumes[20] (Figure 1.2). The two equations are used to construct a periodic, time-dependent mass balance analysis for high solubility (BCS Class I and III) compounds (Figure 3.1). Dose time refers to the time of dosing relative to the phase cycle, i.e. $k_{ge}(t + t_0)$ and $t_{lag}(t + t_0)$ for some $t_0 \geq 0$. Two examples of dose time selection are illustrated in Figure 1.4.

$$k_{ge}(t) = \rho_2 \Big(\sum_{k=1}^{N} \big(\frac{(-1)^k sin(-\theta k(t - \tau))}{k} + \rho_1 \big) \Big)^{\rho_3} + s \tag{1.1}$$

$$t_{lag}(t) = c_1 - \frac{c_2}{c_3 + c_4 \ exp(-c_5 Mod[t_0, 120] + c_6)} \tag{1.2}$$

The compartments are treated as continuous stirred-tank reactors (CSTRs), balancing the influx/outflux of mass or concentration. Drug-containing volume enters the stomach at dose time t_0 and empties, according to the cyclical, time-dependent emptying rate into a series of intestinal compartments with two outflows each–an effective permeation rate into the plasma compartment and an intestinal transit rate (equal to a phase shifting of the gastric emptying rate, i.e. $k_{int_N}(t) = k_{ge}(t + \tau)$ for some time offset τ). This is equivalent to a contractile wave propagating along the

GI tract. A 3-dimensional plot in Figure 1.5a shows the percent emptied from the stomach compartment versus time t and dose time t_0.

Plasma elimination half-lives of7, 14, 30, 60, and 240 minutes are used. The multiple intestinal compartments reflect the different regions on the small intestine. Yu et al. previously described GI transit using a 7-compartment model which best reflected the average intestinal transit time of 199 minutes[35–37]. Similarly, here each intestinal compartment is assigned transit rate $k_{int_N}(t)$, an effective permeation rate $perm_N$, and a back-mixing rate $Q_N(t)$ for each of the 3 pairs after the first intestinal compartment to reflect segmental contractions that can move luminal content in the orad direction[26–28]. Permeation rates can vary in different compartments, thus allowing consideration of location-dependent intestinal absorption as yet another factor in plasma level variation.

1.2.3 Intra- and inter-patient variation and bioequivalence

Bioequivalence (BE) is defined as the mean 90% confidence interval (CI) of a test product falling within the 80-125% range about the mean of the reference product. To account for variability between simulated BE trials, 24 virtual patients are generated from a uniformly distributed range of ±10% about the parameter values used in generating the functions. For each virtual patient, a simulation is carried out over a 2-hour range of dose time t_0. Resultant plasma profiles and the BE metrics of maximum plasma concentration C_{max}, time of maximum plasma concentration T_{max}, and bioavailability AUC are considered versus dose time, highlighting the effects of inter- and intra-patient variation. The population reference means and medians are calculated from a sample of 10,000 values while the 6-, 12-, and 24-subject test values are randomly selected from within that sample.

1.2.4 Model validation

1.2.4.1 Gastric emptying patterns

Using non-disintegrating dosage forms during the fasted state, Weitschies et al. demonstrated a broad time range of 1 to 185 minutes for gastric emptying time with median and mean of 21 minutes and 37 minutes, respectively[38]. Simulations are done over a range of dose times from $t_0 = 0\ min$ (initial phase I) until $t_0 = 120\ min$ (terminal phase III). The causatum range of gastric emptying patterns of the three phases and stomach emptying half-times are in accordance with reported gastric emptying patterns (Figures 1.5b, 1.6, 1.7)[20].

Mudie et al. noninvasively measured emptying of 240 mL water in fasted healthy humans using magnetic resonance imaging (MRI). They quantified the distribution of water volumes into packets along the intestinal tract and showed that baseline (resting) volumes varied greatly among different subjects during the fasted state[25]. Normalizing to resting volumes, Figure 1.8 shows the emptying patterns overlayed on the range of simulated predictions. In the study by Mudie et al., 75% of the subjects displayed first-order emptying patterns while 25% had non-first order, biphasic emptying. These ratios are similar in the simulations (80% and 20%, respectively). Mean simulated gastric emptying time is comparable to that of reported 100 mL solutions containing non-absorbable diethylenetriaminepentaacetic acid (DTPA) labeled with technetium 99m administered to test subjects and monitored via scintigraphy[36] (Figure 1.9, $p = 0.79$).

1.2.4.2 Intestinal transit

The simulated intestinal transit time is also similar to reported values (Figure 1.9, $p = 0.62$). Davis et al. measured labeled solution transit times by imaging anterior and posterior abdomen sites and quantifying radioactivity of labeled solution[36]. In the simulations, a non-absorbable (permeation rates $p_{eff} = 0$) dose administered at a given dose time t_0 empties from the stomach; entrance of the solution into the first compartment (when the concentration exceeds 1%) marks the starting time while exit from the final compartment (when concentration reaches below 1%) marks the endpoint, the difference between the two being the intestinal transit time. Back-mixing is optimized to reflect measured intestinal transit times. A 120-minute range of dose times is used for each simulation of intestinal transit using different back-mixing values, either constant for all three such that $Q(t) = 0.02\ min^{-1}$ (equaling the slow, phase I forward rate); or proportional to the intestinal transit rate functions (Equation 1.3).

$$Q_N(t) = \alpha\ k_{int_{(2N-1)}}(t) \text{ for } N \in \{1, 2, 3\} \text{ and } \alpha \in [0.01, 0.5] \tag{1.3}$$

The distributions of simulated intestinal transit rates is optimized to reflect the reported distributions via a Mann-Whitney test comparing the mean simulated and experimental times, yielding a backflow parameter of $\alpha = 15\%$. The simulated and experimental intestinal residence time distributions are shown in Figure 1.9 and agree with previous results[33,35–37,39–47]. The experimental distribution includes both inter- and intra-patient variation since it is an agglomeration of over 400 human studies, while the simulated distribution captures only intra-patient variation (i.e. that due to gastric emptying and motility phase). A comparison of small intestine residence time predictions to the compartmental model proposed by Yu et al.[35] is shown in Figure 1.10. The advantage of this method is the ability to capture variations due to motility phases: the previous model yields, for any particular formulation dosing, a single residence time irrespective of how many simulations are carried out. The current model yields instead a prediction range that reflects the effect of dose time and evolving gastric emptying rate–as well as the related MMC propagation along the GI tract–and still generates a mean residence time in accordance with experimental values.

1.3 Results and Discussion

1.3.1 Dose time dependence and plasma levels

Previously, Kaus et al. showed in simulations that C_{max} was susceptible to changes in gastric emptying for high permeability compounds[48]. Here the effective permeability rates for all intestinal compartments are on the order of 10^{-3} cm/s for BCS Class I simulations. Thus the rapid absorption means the plasma elimination half-life is the rate-limiting step even in very short half-life scenarios. A slow but constant permeation on the order of 10^{-5} cm/s is used for BCS Class III simulations where the volumetric effect is less pronounced and variation in the plasma profile versus dose time is mitigated. Restricting the slow permeation to initial intestinal compartments (no permeation thereafter), however, results in slow absorption only in the first three intestinal compartments, thus gastric emptying also plays a significant role in how quickly the contents are presented and surpass the sites of absorption. The 50 mL volume is more susceptible to plasma level variations as a function of dose time in the BCS Class I simulations while in the BCS Class III simulations both the 50 mL and 200 mL volumes are affected (Figures 1.11,1.12,1.13).

1.3.2 Example cases

Three compounds with relatively short plasma elimination half-lives are presented as example cases (Table 1.2).

1.3.2.1 Fluvastatin

Fluvastatin, used for treatment of hypercholesterolemia and cardiovascular disease, has a reported plasma elimination half-life of 1-3 hours and is highly soluble (33 mg/ml) and permeable ($cLogP$ = 4)[49–52]. In the study by Tse et al., 24 healthy male subjects received single doses of fluvastatin which were rapidly absorbed from the GI tract though showed low bioavailability due to high first-pass metabolism[52]. The experimental results and standard deviations are shown in red in Figure 1.14a. After correcting for volume of distribution, the simulation using a short plasma elimination half-life yields the mean predicted plasma profile (dashed green line) and upper/lower bounds for the plasma levels (shaded region) consistent with the inter-subject variations reported by in the clinical study, illustrating the potential to reproduce accurate variations in a population.

1.3.2.2 Fluorouracil

Fluorouracil is a BCS Class III compound (reported solubility of 11,000 mg/mL, $cLogP$ = -0.66) with a very short, 7-16 minute plasma elimination half-life used to treat breast, ovary, and GI tract cancers[53–57]. Phillips et al. dosed the compound intravenously and then orally in patients[55]. Accounting for bioavailability and volume of distribution, Figure 1.14b shows the reported plasma concentrations (red) overlayed on top of the simulated envelope of maximal and minimal plasma levels (shaded region). In addition to capturing the variation of the study population, the dose time is approximated to individual plasma profiles.

1.3.2.3 Diethylcarbamazine

Used for treatment of filariasis, diethylcarbamazine (DEC) is an anthelmintic BCS Class III compound (reported solubility of 750 mg/mL, $cLogP$ = 1.62) with a plasma elimination half-life of 8 hours[58–61]. In a small study of 12 healthy volunteers, Bolla et al. gave single doses to the subjects in a crossover study at either 0600h or 1800h[58]. They reported statistically significant differences in plasma levels between the two cohorts, consistent with previous studies highlighting circadian effects on motility[62–64].

Figure 1.14c shows both the 0600h and 1800h groups in red and blue, respectively, superimposed over the predicted envelope of maximal and minimal plasma levels using the a 480-minute plasma elimination half-life. The predicted range (light green) spans the variation of both groups, and the mean simulated plasma curve (dashed green line) transects the mean reported values.

1.3.3 Bioequivalence implications

1.3.3.1 BCS Class I

By perturbing the physiological parameters over a ±10% uniformly distributed range about the means, considerable variation is seen in the resultant BE simulations. Figures 1.15, 1.16, and 1.17 illustrates the effect of increasing plasma elimination half-life and volume of co-administered liquid for simulations of BCS Class I, BCS Class III with constant permeation, and BCS Class III with regional permeation, respectively. With very short, 7-minute elimination half-life, 25% of BCS Class I BE studies are expected to fail when considering the mean test C_{max} 90% CI versus the reference 80-125% range. With a 50 mL volume, gastric emptying accounts for 59% of the 80-125% interval for 7-minute plasma elimination half-life drugs, and this decreases to 5% for 240-minute plasma elimination half-life drugs. Similarly, with a 200 mL volume, the percent of the 80-125% range covered decreases from 22% to 3% for 7- to 240-minute plasma elimination half-lives. Thus longer plasma elimination half-lives and increased volumes mitigate variation caused by gastric emptying. Using the test C_{max} medians rather than the means, it is expected that 98% of BCS Class I BE studies will fail for short elimination half-life drugs in 50 mL volumes, and even with 200 mL volumes 10% will still fail. The percent of the 80-125% range attributed to gastric emptying variation increases dramatically as well: 8-134% for 50 mL and 5-32% for 200 mL volumes. The results are summarized in Table 1.3.

1.3.3.2 BCS Class III

In the case of BCS Class III simulations with constant and low effective permeation in the intestinal compartments, the variations are far less extreme. No BE trials are expected to fail, while only 4-13% (50 mL volume) and 3-14% (200 mL volume) of the 80-125% reference range are attributable to gastric emptying variation. These rates do not increase significantly when the median 90% CIs are instead considered (Table 1.4).

Perhaps counterintuitively, however, for BCS Class III simulations with locational permeation, increases in volume and plasma elimination half-life yield greater variation. For 240 min plasma elimination half-lives, 4% and 14% of the BE trials are expected to fail using 50 mL and 200 mL volumes, respectively. Furthermore, 14-38% (50 mL volume) and 27-57% (200 mL volume) of the reference 80-125% range are due to variations in gastric emptying. These rise to 16-47% and 42-113% for 50 mL and 200 mL volumes, respectively, when the median CIs are used rather than the means (Table 1.5). This reversal in trend is likely due to the change in the rate-limiting step: BCS Class I compounds are readily absorbed along the entire intestinal tract and thus the longer half-lives exempt gastric emptying as a source of variation. That is, no matter how quickly or slowly the drug solution reaches the site of absorption, uptake occurs rapidly enough and the comparatively longer elimination half-lives allow sufficient time to achieve the same maximum concentration and bioavailability. Conversely, BCS Class III absorption profiles can depend heavily on intestinal location. Thus with short plasma elimination half-lives, the drug is quickly cleared from the system and so variation remains low (as absorption is the rate-limiting step). However, as plasma-elimination half-life increases and becomes the rate-limiting step, the effect of gastric emptying is now emphasized to a greater degree: slowly emptying and transiting content has time to be absorbed while faster moving drug concentrations necessarily miss, at least in part, the site of absorption. Since the 50 mL volume is more concentrated, enough drug may be absorbed and available systemically, while the 200 mL is more dilute and thus the potential to only partially absorb or miss entirely is reflected even more severely in the C_{max} variations.

1.3.3.3 Extending parameter variation and plasma elimination half-life

The current model assumes a uniform ±10% variation about the mean physiological parameters, underpredicting the extent of experimental variation especially for phase I lag times and phase III gastric emptying rates[20]. The results are also thus far relevant to only short plasma-elimination half-lives. Therefore further simulations are run with extended plasma elimination half-lives up to 24 hours and incrementally increased parameter variations from ±10% to ±75%. The coefficient of variation (CV) for bioavailability increases both with greater parameter variation as well as longer plasma elimination half-life in the BCS Class I simulations, however, it is not as dramatic in the BCS Class III simulations where CV is constant or indeed decreasing with greater parameter variation (Figure 1.19). BCS Class I C_{max} CV increases with increased parameter variation however decreases as plasma elimination half-life is increased, while BCS Class III C_{max} CV increases with both greater parameter variation as well as greater plasma elimination half-life (Figure 1.20). Only for a very short, 7-minute plasma elimination half-life is there a significant expected rate of BE failure in the BCS Class I simulations. However, for a 200 mL volume in the BCS Class III model, longer plasma elimination half-life and greater parameter variation both contribute to increased expected BE failures, consistent with the previous explanation (Figure 1.21).

1.3.3.4 Statistical considerations

The distribution of pharmacokinetic parameters is generally assumed to be lognormal[65–73]. While gastric emptying is but one of a multitude of factors influencing plasma levels, its contribution in isolation appears to be asymmetrically -biased (i.e. an extreme value distribution). For the BCS Class I and III simulations (with both constant and locational absorption), the log-transformed population C_{max} values do not display normality whether using a 50 mL or 200 mL volume. The population histograms are overlayed with normal (red) and either Gumbel or Weibull (blue) approximations (Figures 1.22, 1.23, 1.24). It is perhaps reasonable to consider non-parametric tests where applicable and to employ both the mean and median, at the very least, when seeking to determine BE since the mean is more impervious to so-called fat tailed distributions while the median accounts for these outliers.

1.4 Conclusion

GI motility is an important physiological process that has not been evaluated regarding fasted-state oral absorption kinetics and BE implications. The results of this study indicate that, for short elimination half-life drugs with fast absorption, failures of BE are expected based on gastric emptying variation alone, and that for low absorption drugs, failures may continue with even extended plasma elimination half-lives. The framework for a mechanistic, bottom-up analysis is presented here. Further studies are needed on GI motility to better determine the statistics of gastric emptying in fasted and fed states and the corresponding implications for BE trial design and refinement of the 80-125% range.

1.5 Tables and figures

Table 1.1: Mass balance parameters

Parameter	Description
ρ_1	Length and amplitude of phase I
ρ_2	Length and amplitude of phase II relative to phase I
ρ_3	Length and amplitude of phase III
N	Number of sine functions
ϕ	Half the cycle frequency
s	Initial phase I rate of gastric empting
c_i	Constants from experimental averages[20]

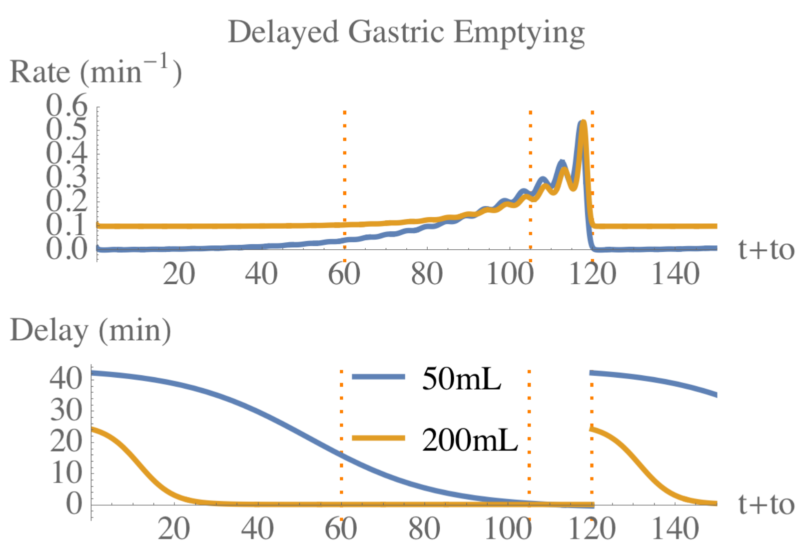


Figure 1.1: Plots illustrating the continuous gastric emptying rate function (top) and the phase-associated lag time function (bottom).

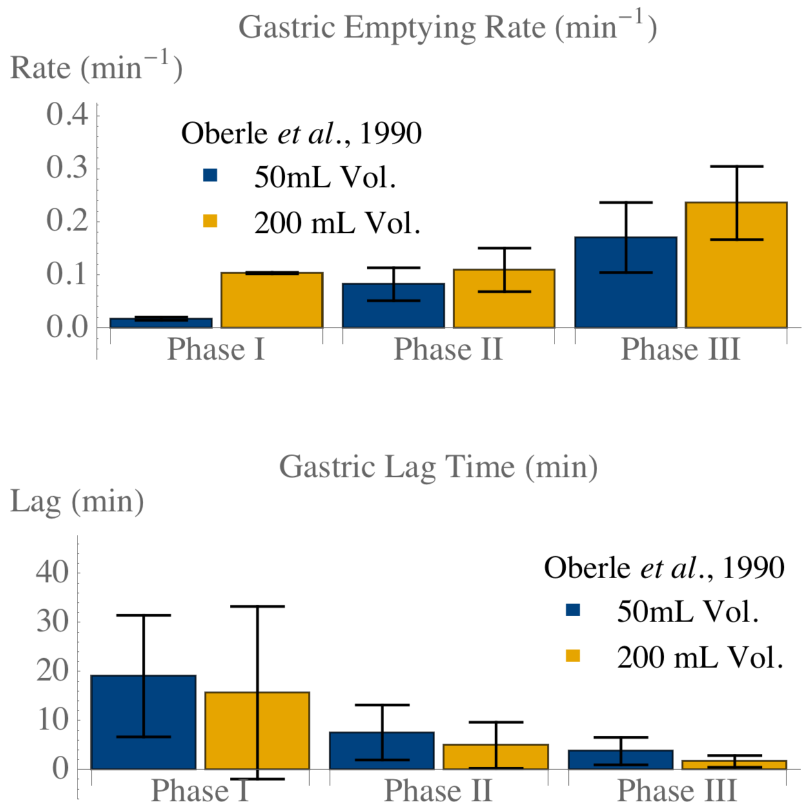


Figure 1.2: Experimentally determined[20] emptying rates (top) and lag times (bottom) at the begging of the three phases.

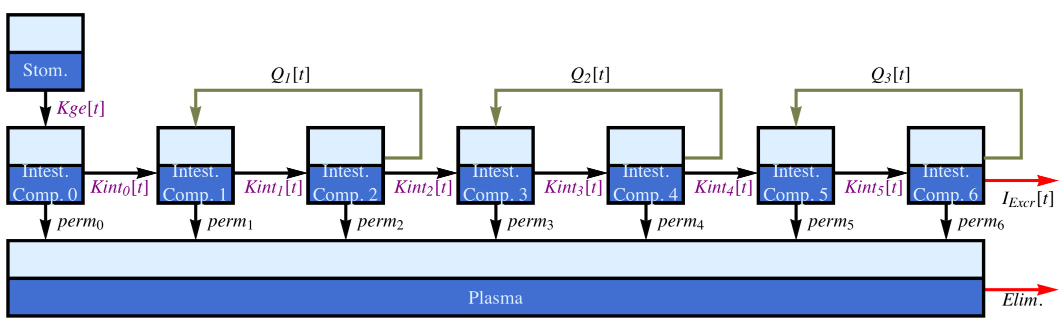


Figure 1.3: Schematic of the GI tract with a pulsatile, time-dependent emptying rate from the stomach compartment into the intestinal compartments. Intestinal transit rates are phase-shifted such that the pulse initiating in the stomach travels through the GI tract over a two-hour time period[8].

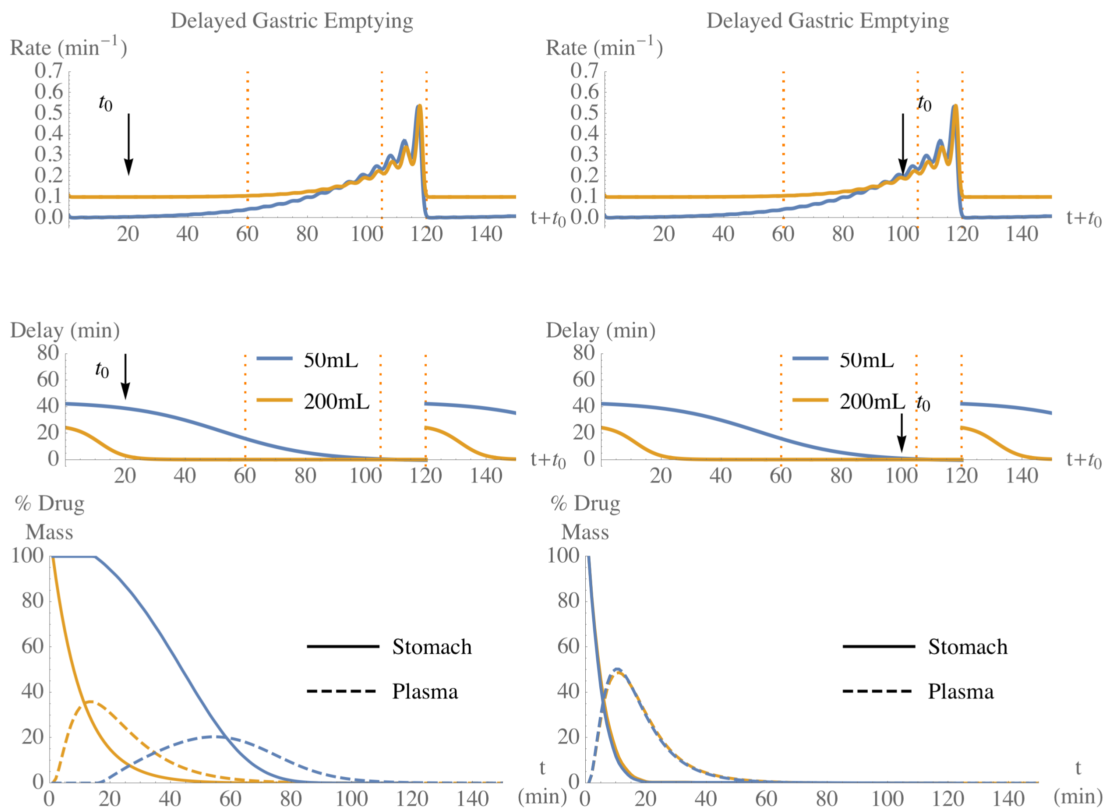
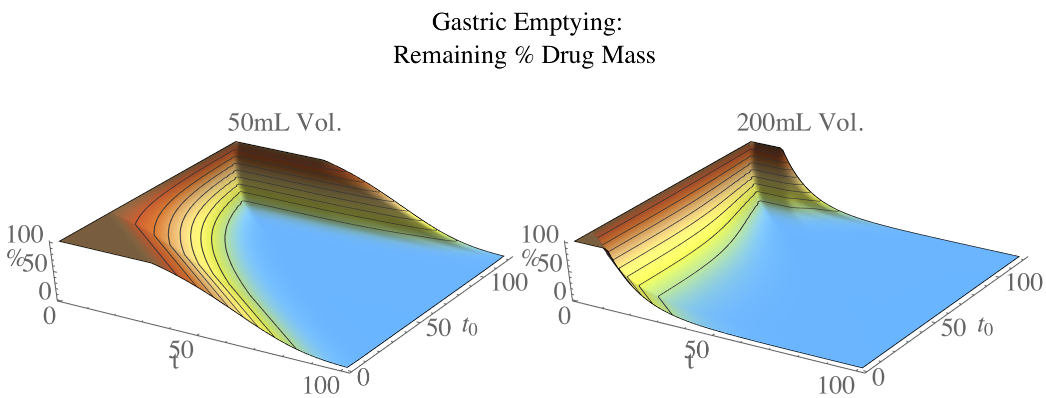
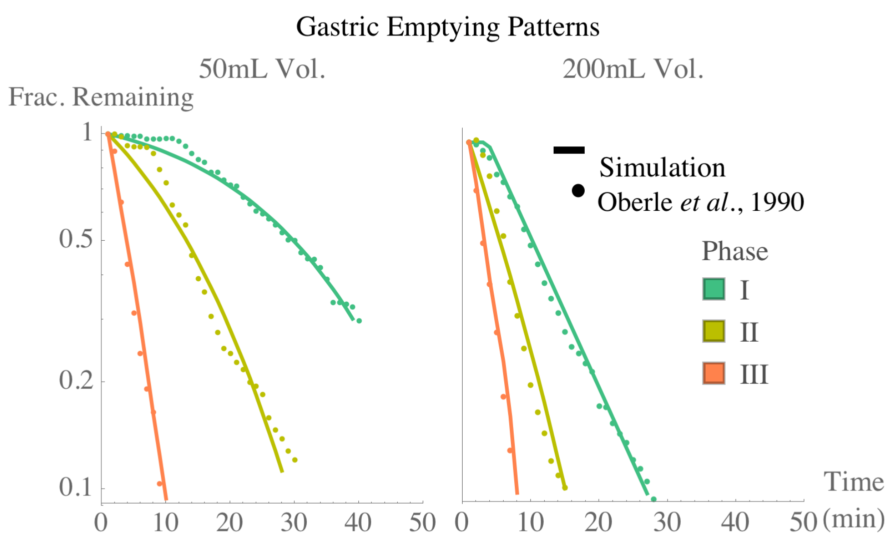


Figure 1.4: Effect of dose time t_0. Left: early dosing, that corresponds to phase I, with a low gastric emptying rate and long lag time; the volumetric lag time dependence results in a considerable difference in the emptying and appearance in plasma. Right: late dosing in phase III where the gastric emptying rate has increased considerably and the lag time is nearly zero; there is a negligible difference between the 50 mL and 200 mL volumes since all gastric content is emptying rapidly and immediately.



(a) Percent emptied from stomach compartment with respect to time t and dose time t_0 for 50 mL (left) and 200 mL (right) liquids, illustrating a volume dependence.



(b) Gastric emptying patterns corresponding to measured rates[20] for 50 mL (left) and 200 mL (right) volumes.

Figure 1.5: Dependence of gastric emptying on volume and dose time (i.e. phase).

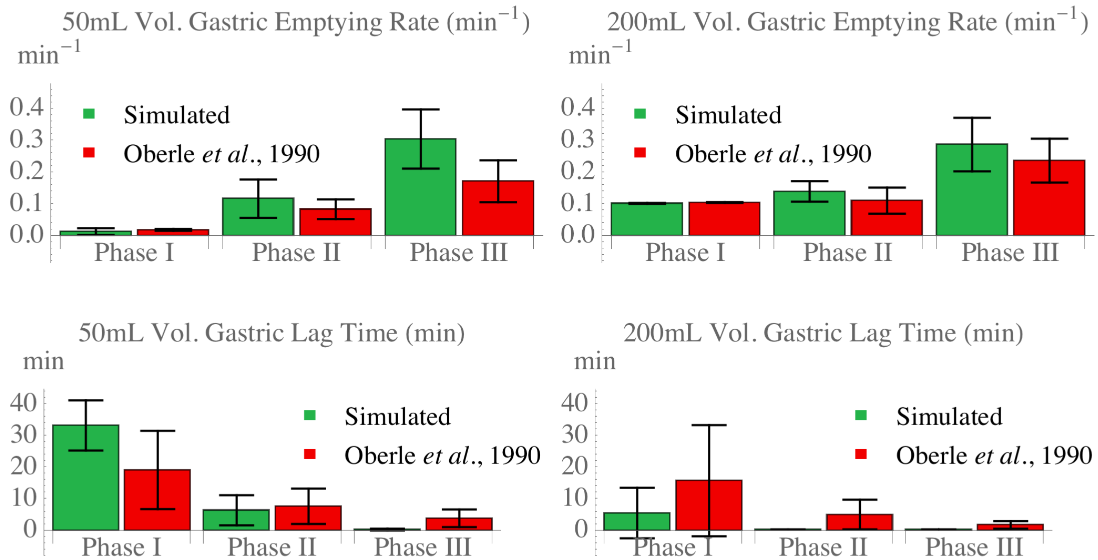


Figure 1.6: A comparison of simulated (green) and experimentally determined[20] (red) gastric emptying rates and lag times for 50 mL (left) and 200 mL (right) volumes.

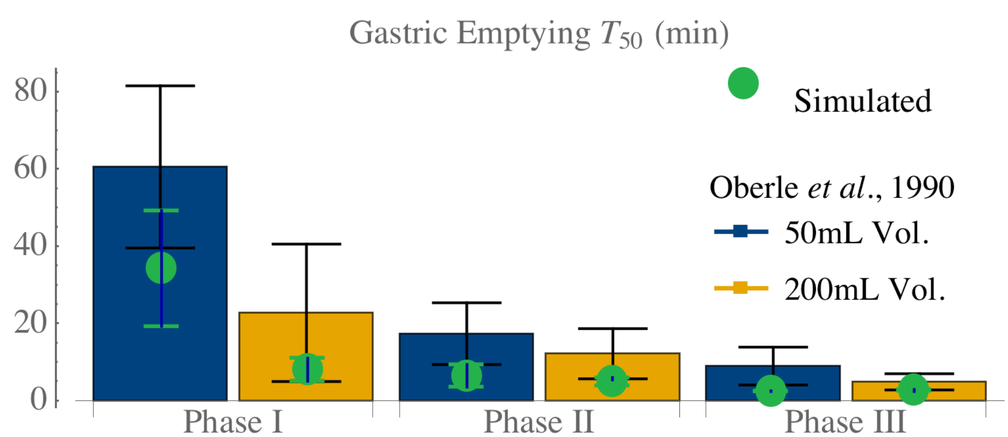


Figure 1.7: Simulated emptying half-times t_{50} (green dots) compared to measured citepOberle1990 t_{50} of 50 mL (blue) and 200 mL (yellow) volumes.

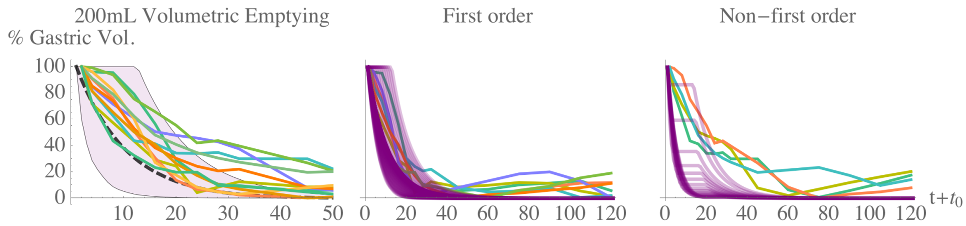


Figure 1.8: Left, the predicted gastric emptying results for a 120-minute range of dose time t_0 (green shaded region, mean prediction as dashed green line) with superimposed volumetric emptying data from subjects measured by MRI[25]. Emptying patterns show both first-order (center, 75%) and non-first order (right, 25%) corresponding to the distribution of simulated emptying patterns.

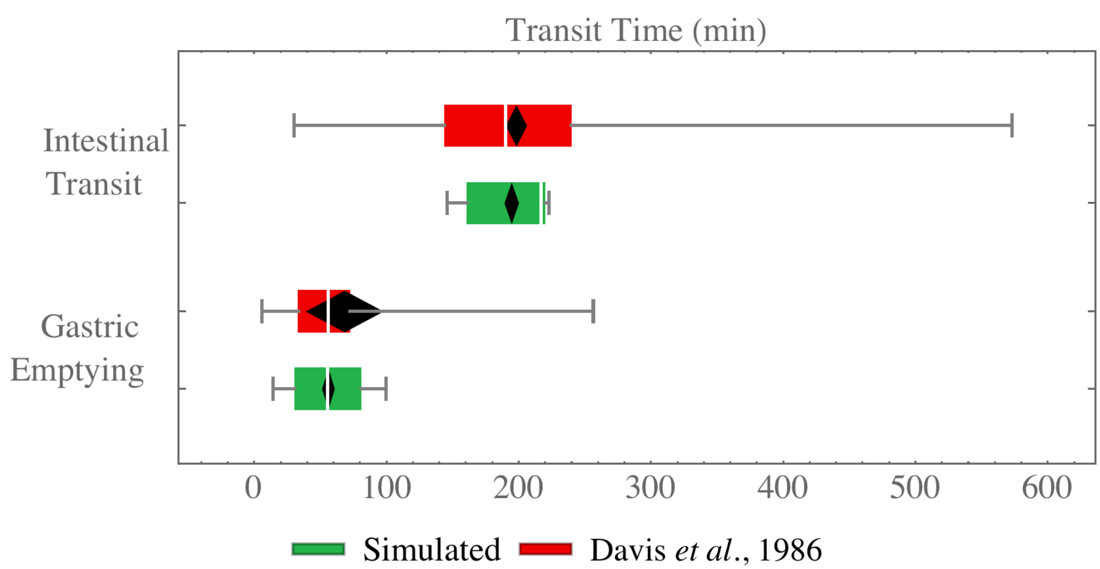


Figure 1.9: Accordance of experimental[36] and simulated gastric emptying and intestinal transit time ranges. The black diamonds represent the 95% CI about the means, the white bars within the colored regions are the medians; the colored regions represent the 25-75 percentiles; and the whiskers span the entire range. The difference between the means is not statistically significant (Mann-Whitney $p = 0.79$ and $p = 0.62$ for gastric emptying and intestinal transit times, respectively).

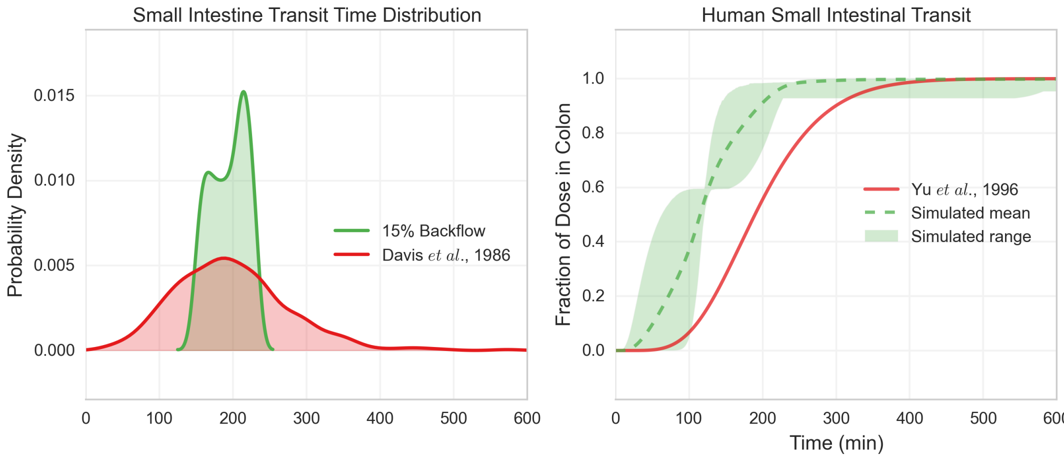
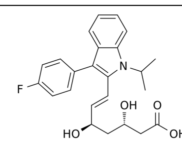
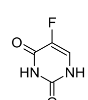
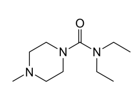
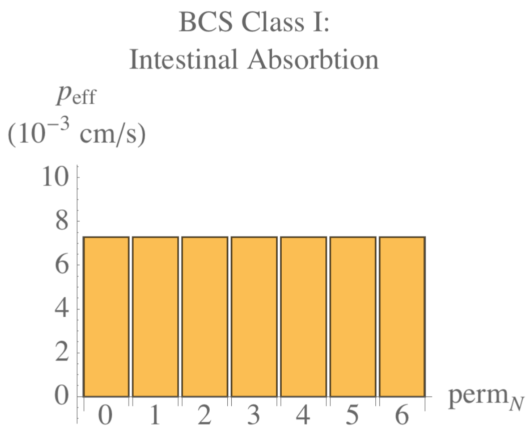


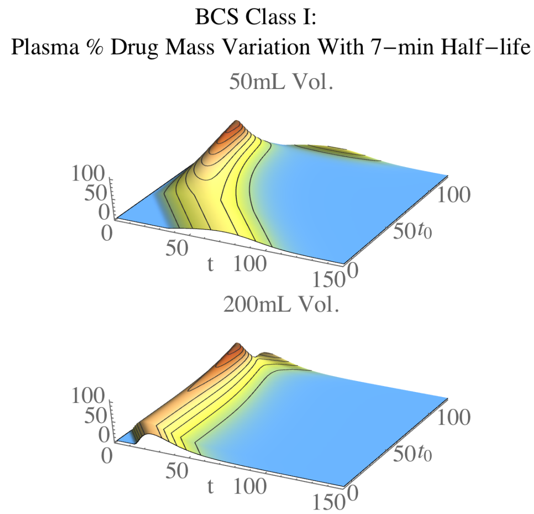
Figure 1.10: Left: probability density of simulated (green) and experimental[36] intestinal transit times for solutions and dosage forms. Right: small intestinal transit flow using previously reported compartmental model[35] (red) and the current model (mean as dashed line; predicted range as green shaded region).

Table 1.2: Properties of example compounds with short plasma elimination half-lives used for simulation validations.

Name	Solubility	cLogP	Structure
Fluvastatin	33 mg/ml	4.00	
Fluorouracil	11,000 mg/ml	-0.66	
Diethylcarbamazine	750 mg/ml	1.62	

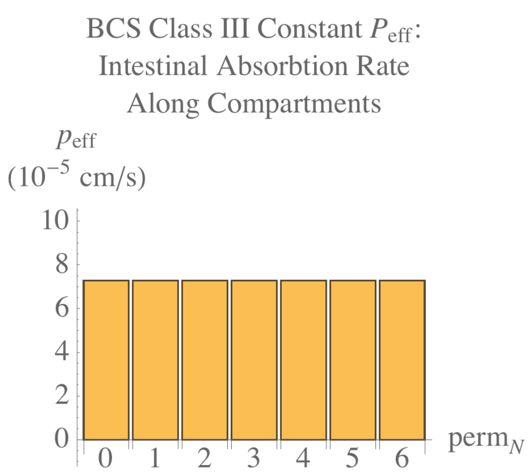


(a) Continuous high effective permeability across the intestinal compartments.

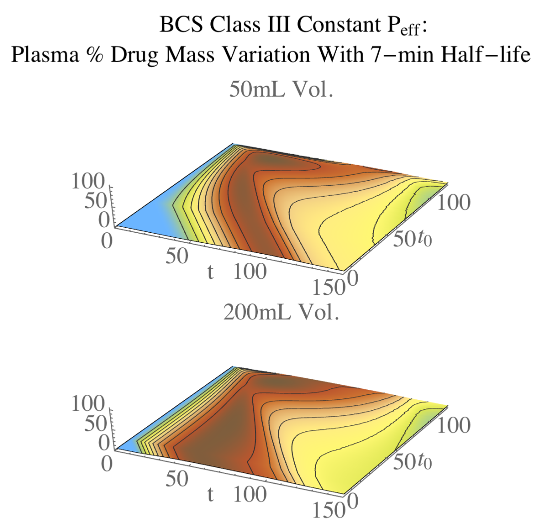


(b) Volumetric effect on plasma level variation. The 50 mL volume (above) is much more susceptible to different phases while it is less pronounced an effect in the 200 mL volume (below).

Figure 1.11: BCS Class I with high p_{eff} showing volumetric effect on plasma levels.

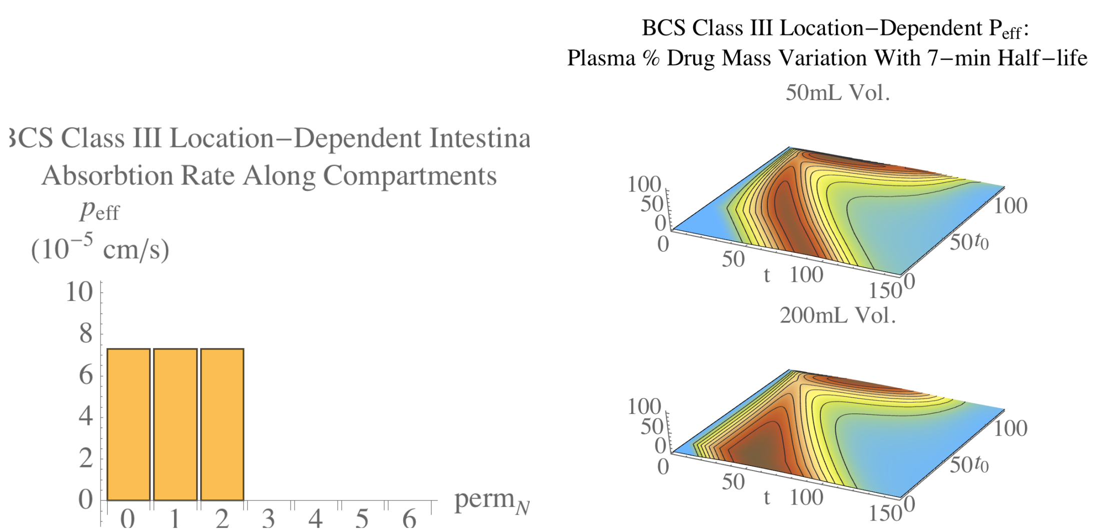


(a) Constant low effective permeability along the intestinal compartments.



(b) Plasma level variation is mitigated when there is continuous but slow permeation across the intestinal compartments.

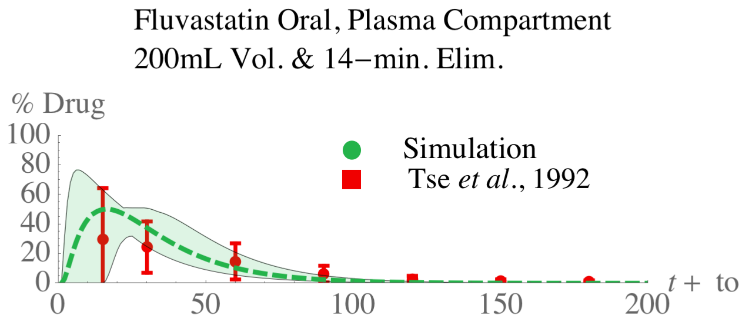
Figure 1.12: BCS Class III constant p_{eff} and volumetric effect on plasma levels.



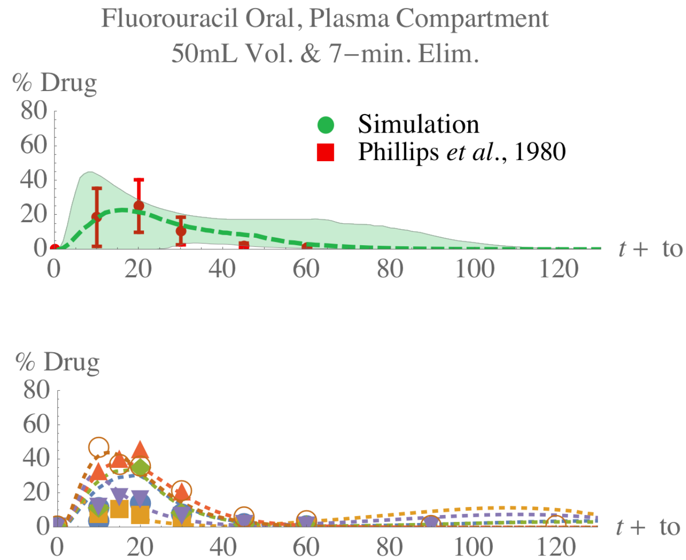
(a) Low effective permeability in the early compartments followed by no absorption in the latter regions.

(b) Plasma level variation is recaptured with location-dependent permeation for both 50 mL (above) and 200 mL (below) volumes.

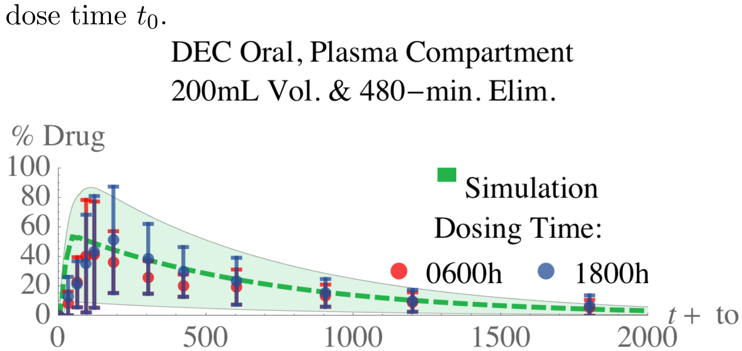
Figure 1.13: BCS Class III location-dependent p_{eff} and volumetric effect on plasma levels.



(a) Experimental[52] fluvastatin plasma levels (red) overlayed on top of mean prediction (dashed green line) and envelope function (green shaded region) showing upper and lower bounds of predicted range.



(b) Plasma level variations of oral fluorouracil (dose- and weight-adjusted)[55]. Above, the shaded regions represent the bounds of plasma level predictions. Below, individual plasma level predictions can be reproduced as a function of dose time t_0.



(c) DEC plasma levels[58] dosed in the morning (red) and evening (blue) overlayed on top of the mean prediction (dashed green line) and the envelope function (green shaded region) showing upper and lower bounds of predicted range.

Figure 1.14: Examples of predicting plasma level variation for short plasma elimination half-life compounds.

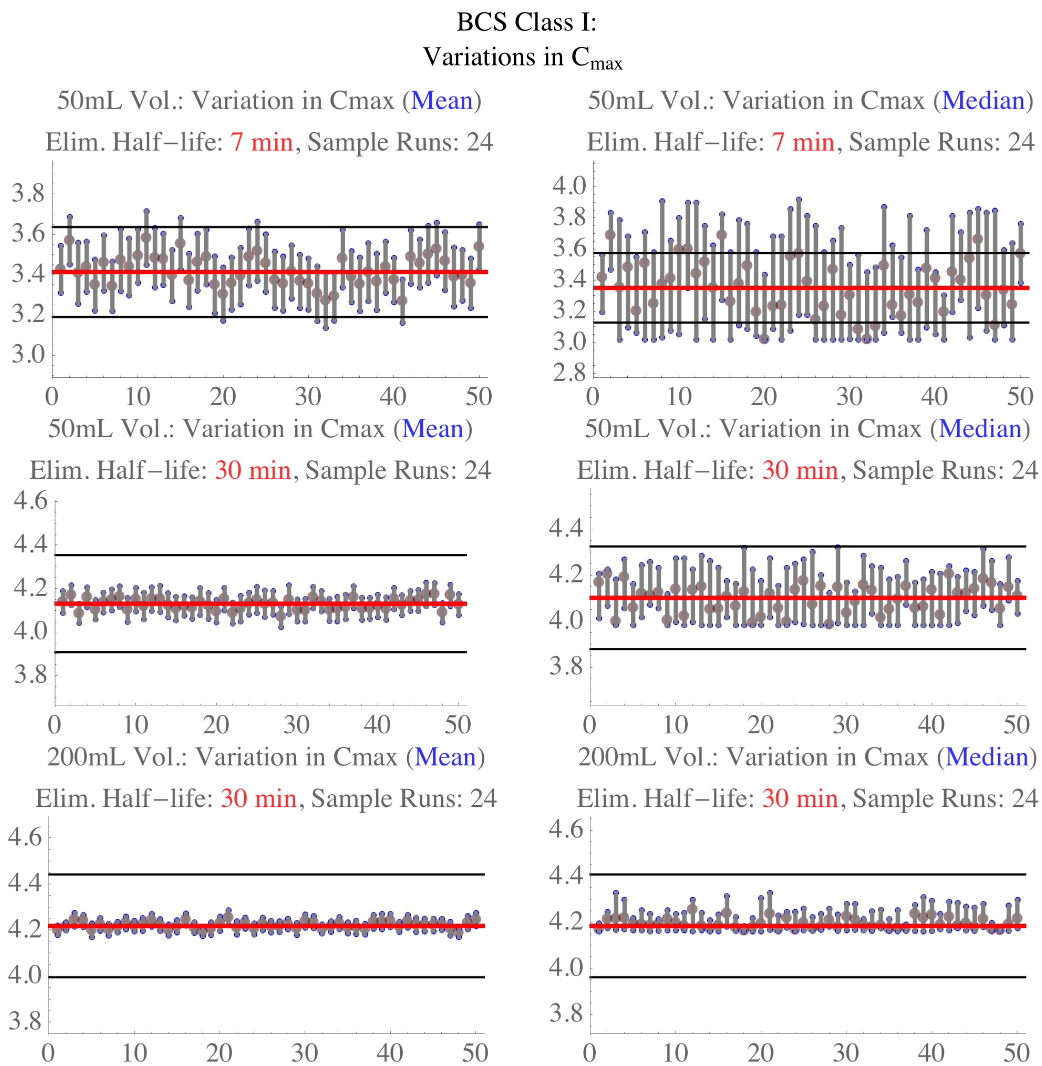


Figure 1.15: Simulated BCS Class I BE trials. The black horizontal bars represent the reference 80-125% range. The vertical bars are individual BE simulations with 24 virtual subjects each, indicating the C_{max} mean 90% CI. In the left column, the mean C_{max} is used while the median is considered in the right column.

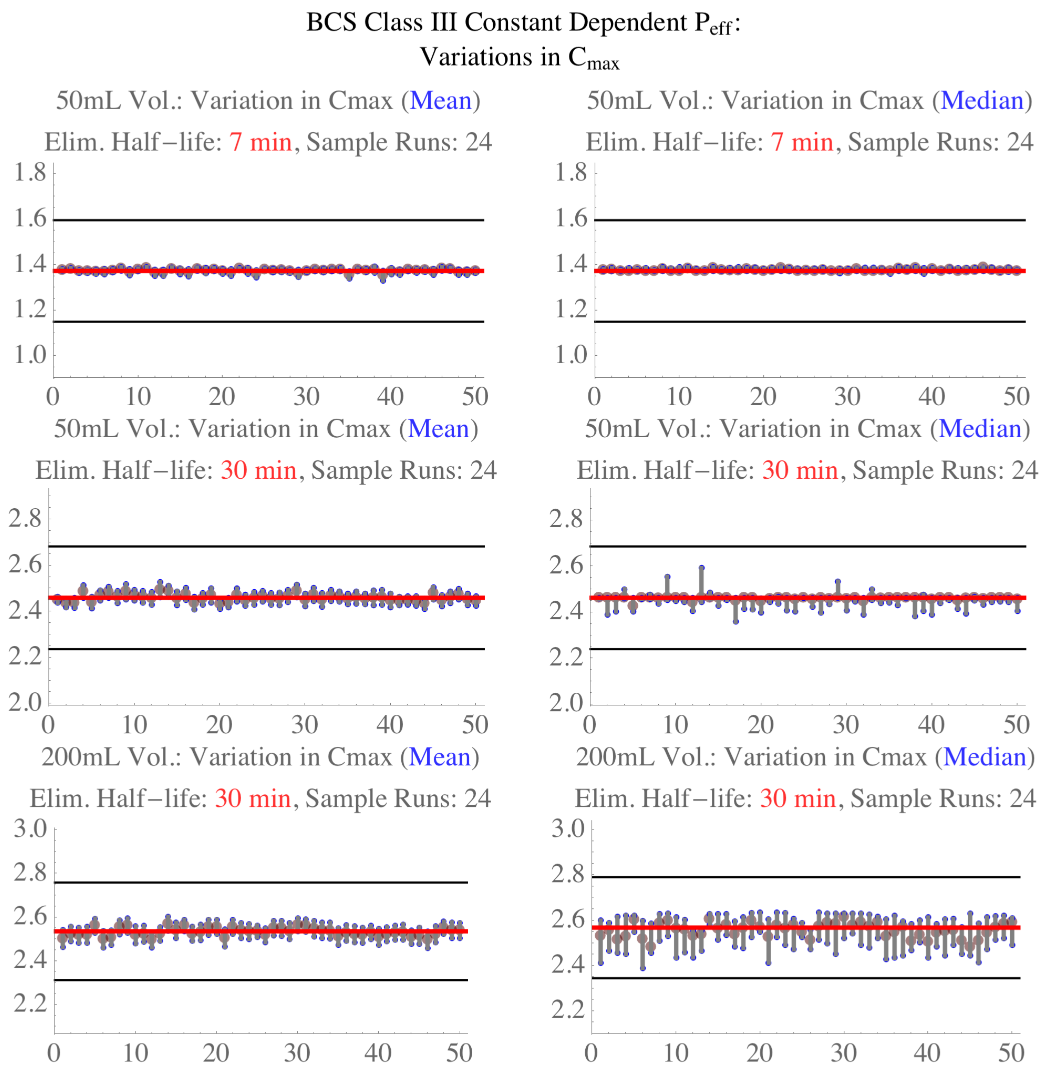


Figure 1.16: Simulated BCS Class III BE trials using constant permeation. The vertical bars are individual BE simulations with 24 virtual subjects each, indicating the C_{max} mean 90% CI. In the left column, the mean C_{max} is used while the median is considered in the right column.

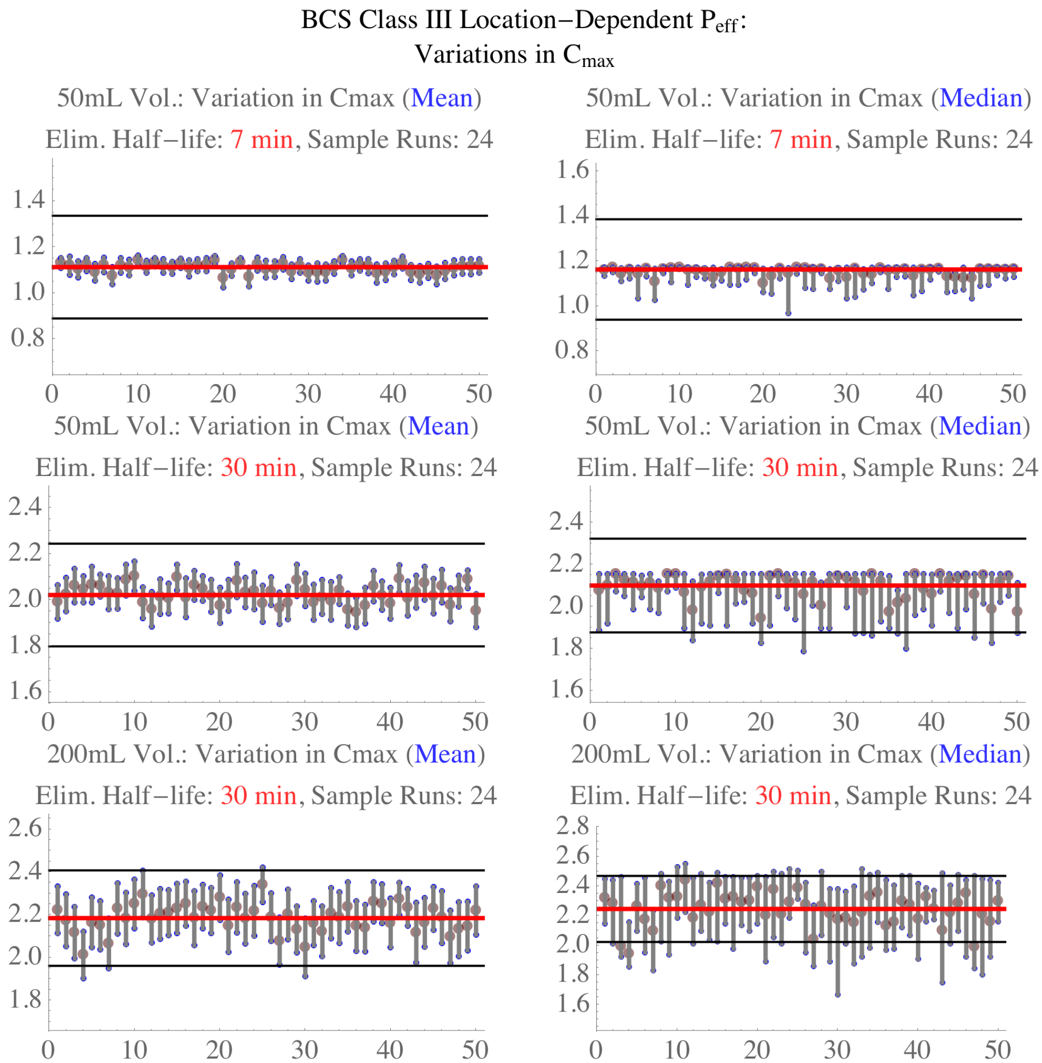
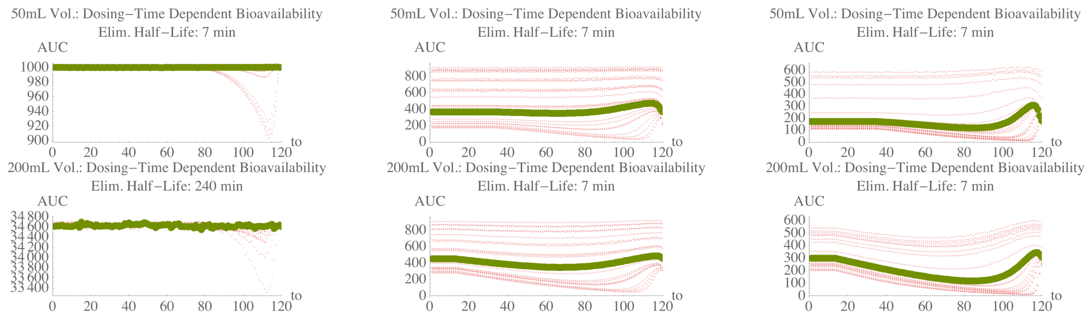


Figure 1.17: Simulated BCS Class III BE trials using location-dependent permeation. The vertical bars are individual BE simulations with 24 virtual subjects each, indicating the C_{max} mean 90% CI. In the left column, the mean C_{max} is used while the median is considered in the right column.



(a) (b) (c)

Figure 1.18: Variations in bioavailability for BCS Class I (a), BCS Class III with constant permeation (b), and BCS Class III with locational permeation (b). Bioavailability as a function of dose time t_0 (green) based on mean emptying rates and lag times compared to a population with varied emptying rates and lag times (dotted red) for each of the three simulations.

Table 1.3: Variation in BCS Class I simulations

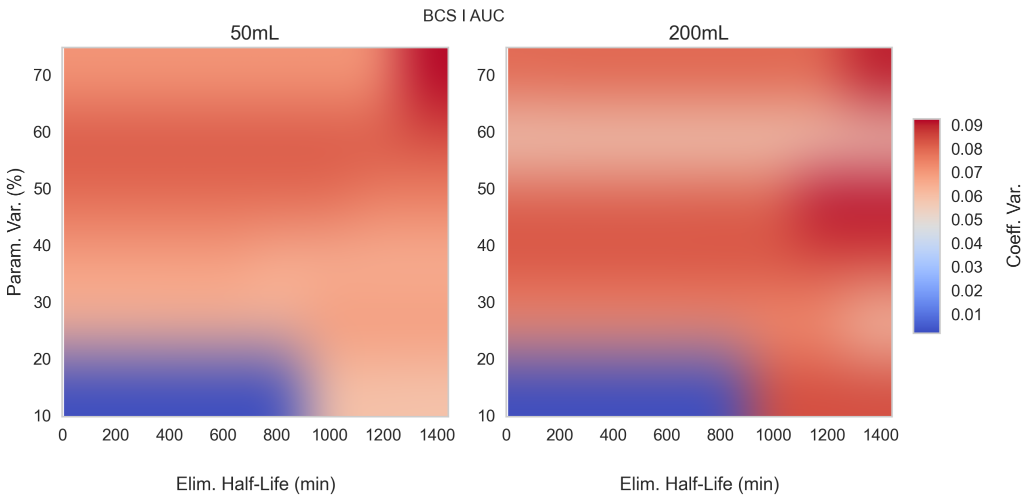
50 mL Elim.	% Fail (mean)	% Fail (median)	% 80-125 (mean)	% 80-125 (median)
7 min	24	98	59.41	134.11
14 min	0	66	39.65	85.02
30 min	0	0	23.32	49.07
60 min	0	0	14.29	27.22
240 min	0	0	5.54	8.03
200 mL Elim.	% Fail (mean)	% Fail (median)	% 80-125 (mean)	% 80-125 (median)
7 min	0	10	22.37	32.00
14 min	0	2	16.45	24.49
30 min	0	0	11.51	19.16
60 min	0	0	7.76	12.18
240 min	0	0	3.07	5.47

Table 1.4: Variation in BCS Class III (cont. abs.) simulations

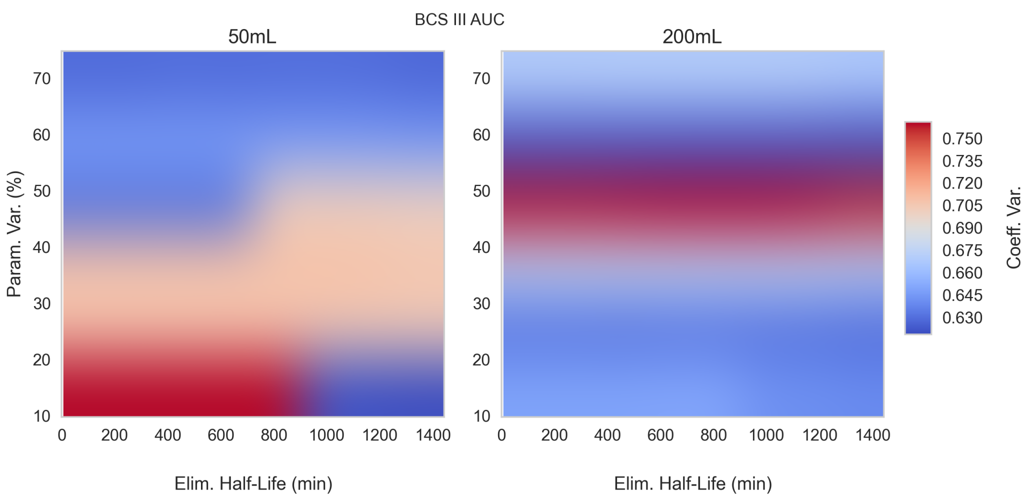
50 mL Elim.	% Fail (mean)	% Fail (median)	% 80-125 (mean)	% 80-125 (median)
7 min	0	0	4.20	2.89
14 min	0	0	5.85	6.01
30 min	0	0	11.21	9.53
60 min	0	2	15.81	12.57
240 min	0	0	12.82	11.40
200 mL Elim.	% Fail (mean)	% Fail (median)	% 80-125 (mean)	% 80-125 (median)
7 min	0	0	3.51	2.73
14 min	0	0	8.91	12.60
30 min	8	0	15.50	28.95
60 min	0	0	18.83	38.85
240 min	0	0	14.39	25.64

Table 1.5: Variation in BCS Class III (loc. abs.) simulations

50 mL Elim.	% Fail (mean)	% Fail (median)	% 80-125 (mean)	% 80-125 (median)
7 min	0	0	13.82	15.97
14 min	0	2	22.45	28.55
30 min	0	22	30.79	44.66
60 min	0	30	35.98	48.62
240 min	4	32	37.63	47.44
200 mL Elim.	% Fail (mean)	% Fail (median)	% 80-125 (mean)	% 80-125 (median)
7 min	0	10	27.24	41.99
14 min	0	30	39.13	65.72
30 min	8	82	50.30	95.98
60 min	10	88	55.14	104.78
240 min	14	98	56.91	113.23

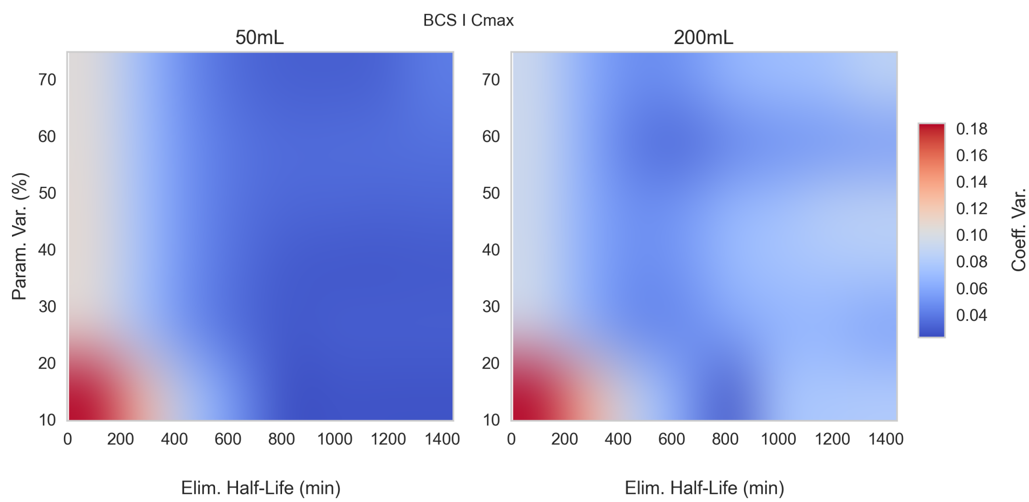


(a)

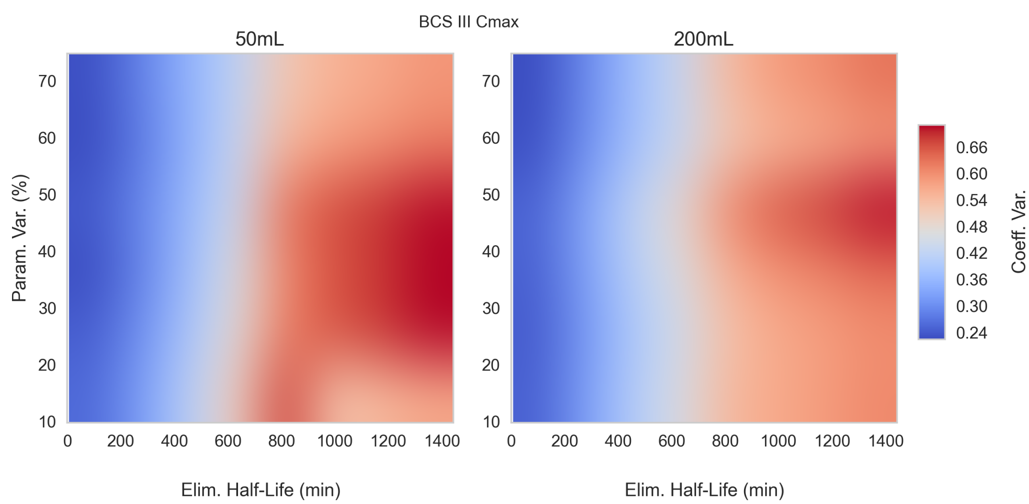


(b)

Figure 1.19: Bioavailability CV for BCS Class I (a) and BCS Class III with locational permeation (a) simulations with respect to plasma elimination half-life and parameter variation (as % of mean).

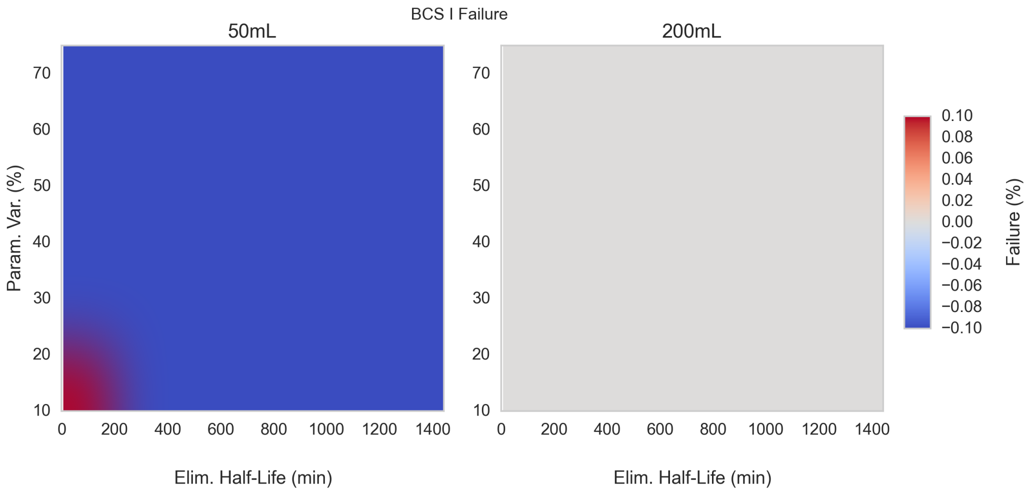


(a)

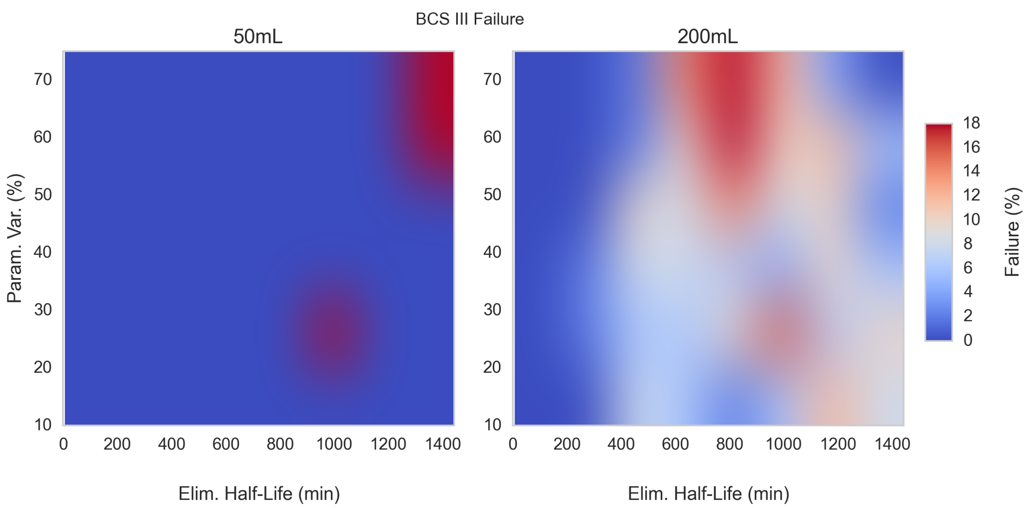


(b)

Figure 1.20: C_{max} CV for BCS Class I (a) and BCS Class III with locational permeation (b) simulations with respect to plasma elimination half-life and parameter variation (as % of mean).



(a)



(b)

Figure 1.21: Expected failure rate for BCS Class I (a) and BCS Class III with locational permeation (b) simulations with respect to plasma elimination half-life and parameter variation (as % of mean).

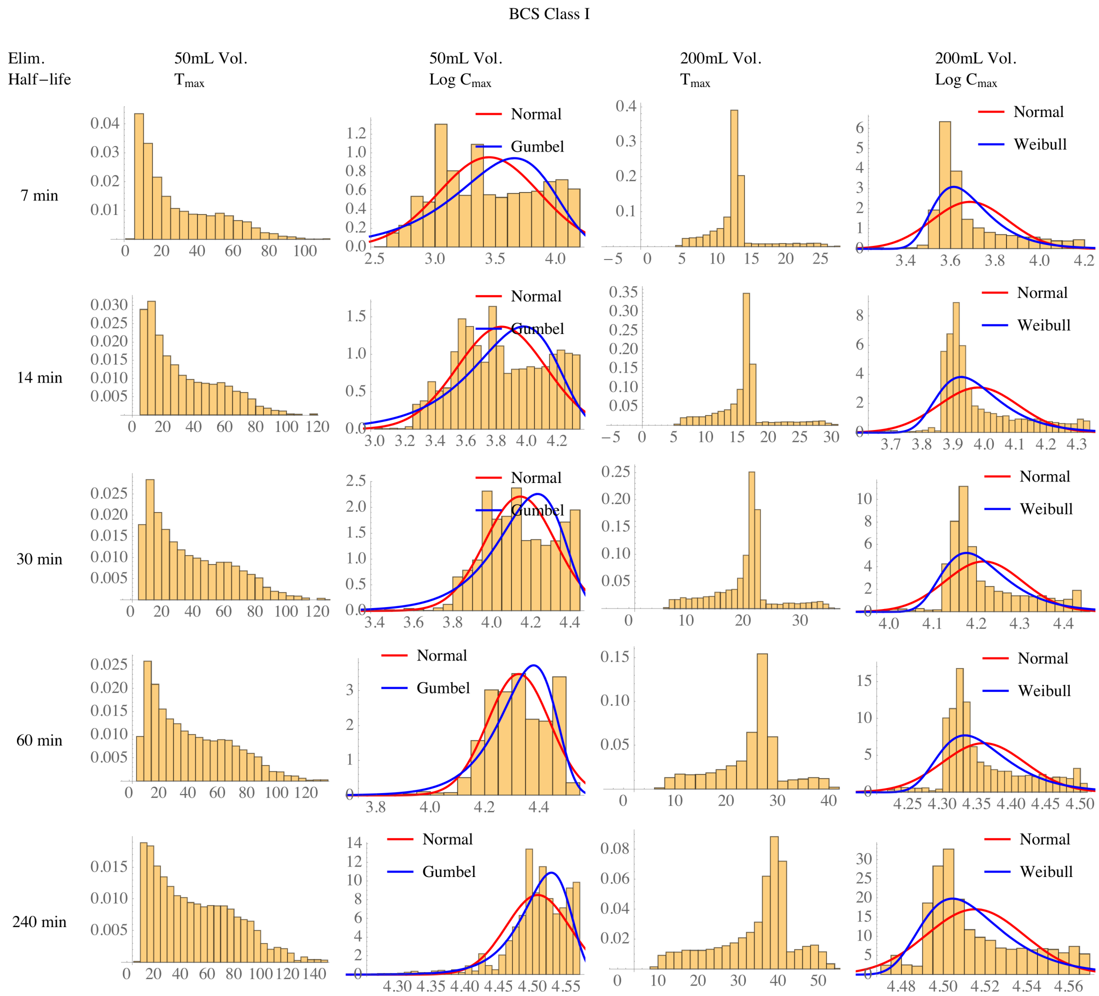


Figure 1.22: Simulated BCS Class I half-life dependence of T_{max} and C_{max} distributions for 50 mL and 200 mL volumes. Overlayed on the C_{max} plots are various distributions, highlighting the lack of normality following a log transformation.

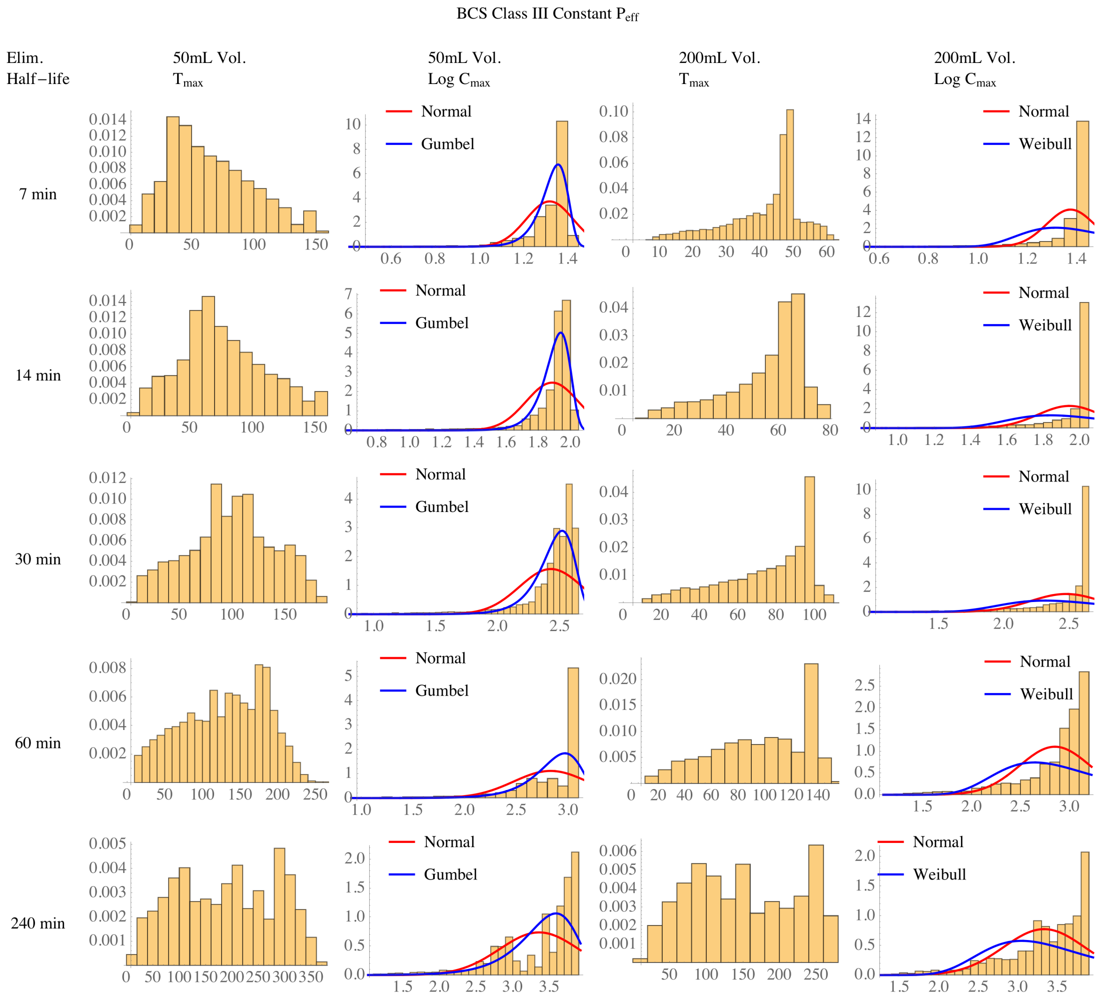


Figure 1.23: Simulated BCS Class III half-life dependence of T_{max} and C_{max} distributions for 50 mL and 200 mL volumes using constant permeation. Overlayed on the C_{max} plots are various distributions, highlighting the lack of normality following a log transformation.

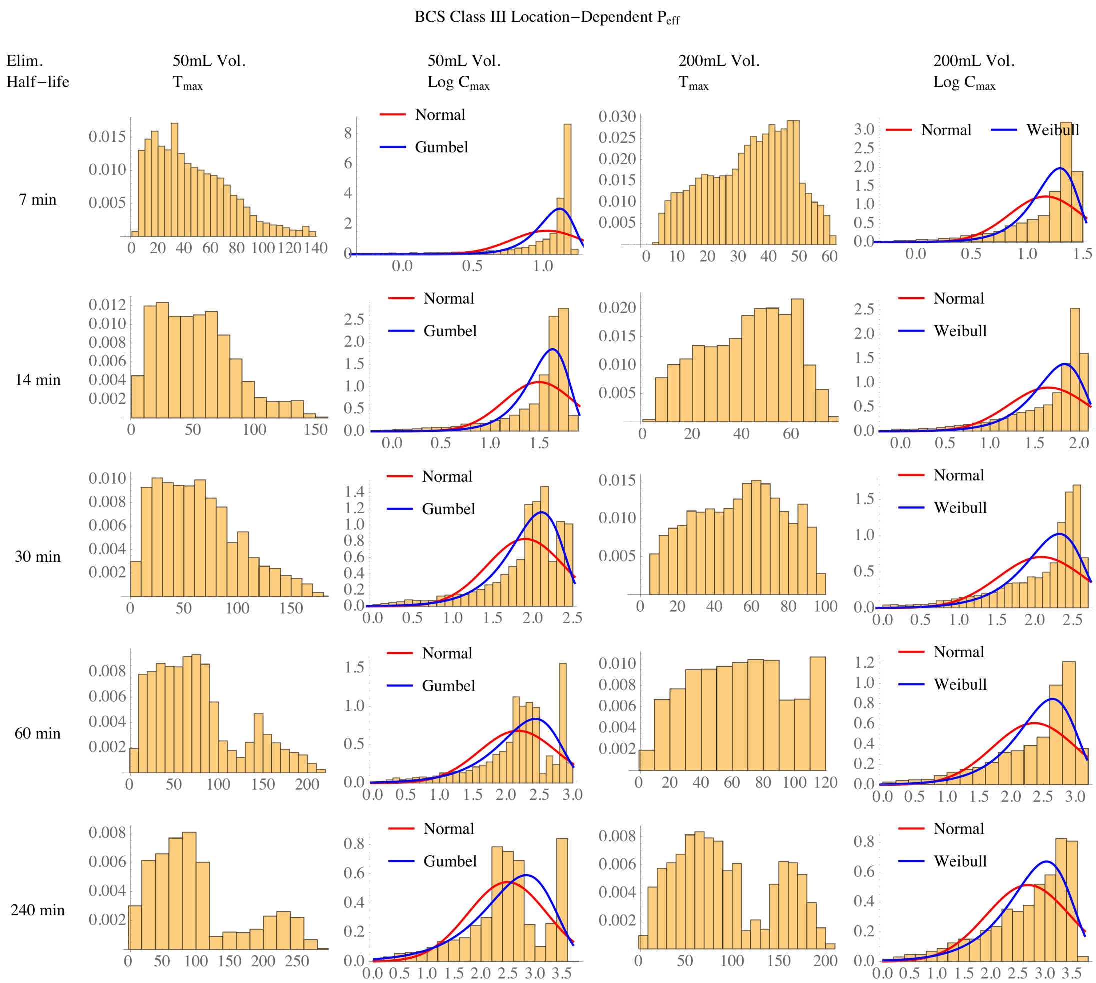


Figure 1.24: Simulated half-life dependence of T_{max} and C_{max} distributions for 50 mL and 200 mL volumes using location-dependent permeation. Overlayed on the C_{max} plots are various distributions, highlighting the lack of normality following a log transformation.

1.6 References

[1] Alan H Maurer. Advancing gastric emptying studies: standardization and new parameters to assess gastric motility and function. *Seminars in nuclear medicine*, 42(2):101–12, March 2012. ISSN 1558-4623. doi: 10.1053/j.semnuclmed.2011.10.001. URL `http://www.ncbi.nlm.nih.gov/pubmed/22293165`.

[2] N W Read and L A Houghton. Physiology of gastric emptying and pathophysiology of gastroparesis. *Gastroenterology clinics of North America*, 18(2):359–373, 1989. ISSN 08898553.

[3] G. Gondolesi, D. Ramisch, J. Padin, H. Almau, M. Sandi, P. B. Schelotto, A. Fernandez, C. Rumbo, and H. Solar. What is the normal small bowel length in humans? First donor-based cohort analysis. In *American Journal of Transplantation*, volume 12, 2012. doi: 10.1111/j.1600-6143.2012.04148.x.

[4] Herbert F Helander and Lars Fändriks. Surface area of the digestive tract - revisited. *Scandinavian journal of gastroenterology*, 49(6):681–9, 2014. ISSN 1502-7708. doi: 10.3109/00365521.2014.898326. URL `http://www.ncbi.nlm.nih.gov/pubmed/24694282`.

[5] W B Cannon. Peristalsis, segmentation, and the myenteric reflex. *American Journal of Physiology*, 30:114–128, 1912.

[6] Dee Unglaub Silverthorn. *Human Physiology: An Integrated Approach*. Benjamin Cummings, San Francisco, 4th edition, 2006.

[7] J. N. Hunt. Gastric emptying in relation to drug absorption. *The American Journal of Digestive Diseases*, 8(11):885–894, November 1963. ISSN 0002-9211. doi: 10.1007/BF02232082. URL `http://link.springer.com/10.1007/BF02232082`.

[8] ML Grivel and Y Ruckebusch. The propagation of segmental contractions along the small intestine. *The Journal of physiology*, 227(2):611–25, December 1972. ISSN 0022-3751. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1331213&tool=pmcentrez&rendertype=abstracthttp://jp.physoc.org/content/227/2/611.short`.

[9] P Kerlin, A Zinsmeister, and S Phillips. Relationship of motility to flow of contents in the human small intestine. *Gastroenterology*, 82(4):701–706, 1982. ISSN 00165085. doi: S0016508582000699[pii].

[10] Peter Fleckenstein. Migrating electrical spike activity in the fasting human small intestine. *The American Journal of Digestive Diseases*, 23(9):769–775, September 1978. ISSN 0002-9211. doi: 10.1007/BF01079784. URL `http://link.springer.com/10.1007/BF01079784`.

[11] S K Sarna and M F Otterson. Small intestinal physiology and pathophysiology. *Gastroenterology clinics of North America*, 18(2):375–404, 1989. ISSN 08898553.

[12] K Higaki, S Yamashita, and G L Amidon. Time-dependent oral absorption models. *Journal of pharmacokinetics and pharmacodynamics*, 28(2):109–28, April 2001. ISSN 1567-567X. URL `http://www.ncbi.nlm.nih.gov/pubmed/11381566`.

[13] Kazutaka Higaki, Sally Y Choe, Raimar Löbenberg, Lynda S Welage, and Gordon L Amidon. Mechanistic understanding of time-dependent oral absorption based on gastric motor activity in humans. *European journal of pharmaceutics and biopharmaceutics : official journal of Arbeitsgemeinschaft für Pharmazeutische Verfahrenstechnik e.V*, 70(1):313–25, September 2008. ISSN 0939-6411. doi: 10.1016/j.ejpb.2008.02.022. URL `http://www.ncbi.nlm.nih.gov/pubmed/18434110`.

[14] R L Oberle and G L Amidon. The influence of variable gastric emptying and intestinal transit rates on the plasma level curve of cimetidine; an explanation for the double peak phenomenon. *Journal of pharmacokinetics and biopharmaceutics*, 15(5):529–44, October 1987. ISSN 0090-466X. URL `http://www.ncbi.nlm.nih.gov/pubmed/3694496`.

[15] Kayode Ogungbenro, Henry Pertinez, and Leon Aarons. Empirical and semi-mechanistic modelling of double-peaked pharmacokinetic profile phenomenon due to gastric emptying. *The AAPS journal*, 17(1):227–36, 2015. ISSN 1550-7416. doi: 10.1208/s12248-014-9693-5. URL `http://www.ncbi.nlm.nih.gov/pubmed/25413723`.

[16] N Ramsbottom, M T Knox, and J N Hunt. Gastric emptying of barium sulphate suspension compared with that of water. *Gut*, 18(7):541–542, 1977. ISSN 0017-5749. doi: 10.1136/gut.18.7.541.

[17] Andreas Steingoetter, Mark Fox, Reto Treier, Dominik Weishaupt, Borut Marincek, Peter Boesiger, Michael Fried, and Werner Schwizer. Effects of posture on the physiology of gastric emptying: a magnetic resonance imaging study. *Scandinavian journal of gastroenterology*, 41(10):1155–1164, 2006. ISSN 0036-5521. doi: 10.1080/00365520600610451.

[18] T. Umenai, N. Arai, and E. Chihara. Effect of the preliminary hydration on gastric emptying time for water in healthy volunteers. *Acta Anaesthesiologica Scandinavica*, 53(2):223–226, 2009. ISSN 00015172. doi: 10.1111/j.1399-6576.2008.01832.x.

[19] J. Wright, V. Adams, J. Hykin, P. Gowland, B. Issa, P. Boulby, P. Tokarczuk, D. Evans, R. Spiller, and P. Mansfield. The measurement of gastric motor function and transit in man by echo planar magnetic resonance imaging. *Magma: Magnetic Resonance Materials in Physics, Biology, and Medicine*, 2(3):467–469, 1994. ISSN 1352-8661. doi: 10.1007/BF01705299. URL http://link.springer.com/10.1007/BF01705299.

[20] R L Oberle, T S Chen, C Lloyd, J L Barnett, C Owyang, J Meyer, and G L Amidon. The influence of the interdigestive migrating myoelectric complex on the gastric emptying of liquids. *Gastroenterology*, 99(5):1275–1282, 1990.

[21] J B Dressman, D Fleisher, and G L Amidon. Physicochemical model for dose-dependent drug absorption. *Journal of pharmaceutical sciences*, 73(9):1274–1279, 1984. ISSN 0022-3549.

[22] R Hintz and K Johnson. The effect of particle size distribution on dissolution rate and oral absorption, 1989. ISSN 03785173.

[23] P E Luner and G L Amidon. Description and simulation of a multiple mixing tank model to predict the effect of bile sequestrants on bile salt excretion. *Journal of pharmaceutical sciences*, 82(3):311–318, 1993.

[24] B Agoram, W S Woltosz, and M B Bolger. Predicting the impact of physiological and biochemical processes on oral drug bioavailability. *Advanced drug delivery reviews*, 50 Suppl 1:S41–S67, 2001. ISSN 0169-409X.

[25] Deanna Mudie, Kathryn Murray, Caroline L Hoad, Susan E Pritchard, Martin Garnett, Gordon L Amidon, Penny A Gowland, Robin C Spiller, Gregory E Amidon, and Luca Marciani. Quantification of gastrointestinal liquid volumes and distribution following a 240 mL dose of water in the fasted state. *Mol Pharm*, 11(9):3039–47, September 2014. ISSN 15438392. doi: 10.1021/mp500210c. URL http://www.ncbi.nlm.nih.gov/pubmed/25115349.

[26] J. M. Andrews, D. G. O'Donovan, G. S. Hebbard, C. H. Malbert, S. M. Doran, and J. Dent. Human duodenal phase III migrating motor complex activity is predominantly antegrade, as revealed by high-resolution manometry and colour pressure plots. *Neurogastroenterology and Motility*, 14(4):331–338, 2002. ISSN 13501925. doi: 10.1046/j.1365-2982.2002.00337.x.

[27] M Castedal and H Abrahamsson. High-resolution analysis of the duodenal interdigestive phase III in humans. *Neurogastroenterology and motility : the official journal of the European Gastrointestinal Motility Society*, 13(5):473–81, October 2001. ISSN 1350-1925. URL http://onlinelibrary.wiley.com/doi/10.1046/j.1365-2982.2001.00281.x/fullhttp://www.ncbi.nlm.nih.gov/pubmed/11696109.

[28] L S Costanzo. *BRS Physiology*. Lippincott Williams & Wilkins, Philadelphia, 5th edition, 2010. ISBN 0781798760.

[29] L. Marciani, J. Wright, S. Foley, C. L. Hoad, J. J. Totman, D. Bush, C. Hartley, A. Armstrong, P. Manby, E. Blackshaw, A. C. Perkins, P. A. Gowland, and R. C. Spiller. Effects of a 5-HT3 antagonist, ondansetron, on fasting and postprandial small bowel water content assessed by magnetic resonance imaging. *Alimentary Pharmacology and Therapeutics*, 32(5):655–663, 2010. ISSN 02692813. doi: 10.1111/j.1365-2036.2010.04395.x.

[30] Luca Marciani, Eleanor F. Cox, Caroline L. Hoad, Susan Pritchard, John J. Totman, Steve Foley, Amisha Mistry, Steven Evans, Penny A. Gowland, and Robin C. Spiller. Postprandial Changes in Small Bowel Water Content in Healthy Subjects and Patients With Irritable Bowel Syndrome. *Gastroenterology*, 138(2), 2010. ISSN 00165085. doi: 10.1053/j.gastro.2009.10.055.

[31] C Schiller, C-P Fröhlich, T Giessmann, W Siegmund, H Mönnikes, N Hosten, and W Weitschies. Intestinal fluid volumes and transit of dosage forms as assessed by magnetic resonance imaging. *Alimentary pharmacology & therapeutics*, 22(10):971–9, November 2005. ISSN 0269-2813. doi: 10.1111/j.1365-2036.2005.02683.x. URL http://www.ncbi.nlm.nih.gov/pubmed/16268972.

[32] T Shingaki, T Takashima, Y Wada, M Tanaka, M Kataoka, A Ishii, Y Shigihara, Y Sugiyama, S Yamashita, and Y Watanabe. Imaging of gastrointestinal absorption and biodistribution of an orally administered probe using positron emission tomography in humans. *Clinical pharmacology and therapeutics*, 91(4):653–9, 2012. ISSN 1532-6535. doi: 10.1038/clpt.2011.267. URL http://www.ncbi.nlm.nih.gov/pubmed/22378159.

[33] S.S. Davis, J.G. Hardy, M.J. Taylor, D.R. Whalley, and C.G. Wilson. The effect of food on the gastrointestinal transit of pellets and an osmotic device (Osmet), 1984. ISSN 03785173.

[34] P. Langguth, K. M. Lee, H. Spahn-Langguth, and G. L. Amidon. Variable gastric emptying and discontinuities in drug absorption profiles: Dependence of rates and extent of cimetidine absorption on motility phase and ph. *Biopharmaceutics and Drug Disposition*, 15(9):719–746, 1994. ISSN 01422782. doi: 10.1002/bdd.2510150902.

[35] Lawrence X. Yu, John R. Crison, and Gordon L. Amidon. Compartmental transit and dispersion model analysis of small intestinal transit flow in humans. *International Journal of Pharmaceutics*, 140(1):111–118, August 1996. ISSN 03785173. doi: 10.1016/0378-5173(96)04592-9. URL http://linkinghub.elsevier.com/retrieve/pii/0378517396045929.

[36] S S Davis, J G Hardy, and J W Fara. Transit of pharmaceutical dosage forms through the small intestine. *Gut*, 27(8):886–92, August 1986. ISSN 0017-5749. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1433369&tool=pmcentrez&rendertype=abstract.

[37] LX Yu and GL Amidon. A compartmental absorption and transit model for estimating oral drug absorption. *International journal of pharmaceutics*, 186(2):119–25, September 1999. ISSN 0378-5173. URL http://www.ncbi.nlm.nih.gov/pubmed/10486429http://www.sciencedirect.com/science/article/pii/S0378517399001477.

[38] Werner Weitschies, Henning Blume, and Hubert Mönnikes. Magnetic Marker Monitoring: High resolution real-time tracking of oral solid dosage forms in the gastrointestinal tract, 2010. ISSN 09396411.

[39] F N Christensen, S S Davis, J G Hardy, M J Taylor, D R Whalley, and C G Wilson. The use of gamma scintigraphy to follow the gastrointestinal transit of pharmaceutical formulations. *The Journal of pharmacy and pharmacology*, 37(2):91–95, 1985. ISSN 0022-3573. doi: 10.1111/j.2042-7158.1985.tb05013.x.

[40] A. J. Coupe, S. S. Davis, and I. R. Wilding. Variation in gastrointestinal transit of pharmaceutical dosage forms in healthy subjects. *Pharmaceutical Research*, 8(3):360–364, 1991. ISSN 07248741. doi: 10.1023/A:1015849700421.

[41] S S Davis, F Norring-Christensen, R Khosla, and L C Feely. Gastric emptying of large single unit dosage forms. *The Journal of pharmacy and pharmacology*, 40(3):205–207, 1988. ISSN 0022-3573.

[42] S.S. Davis, J.G. Hardy, M.J. Taylor, D.R. Whalley, and C.G. Wilson. A comparative study of the gastrointestinal transit of a pellet and tablet formulation, 1984. ISSN 03785173.

[43] S.S. Davis, R. Khosia, C.G. Wilson, and N. Washington. Gastrointestinal transit of a controlled-release pellet formulation of tiaprofenic acid and the effect of food, 1987. ISSN 03785173.

[44] R. Khosla, L.C. Feely, and S.S. Davis. Gastrointestinal transit of non-disintegrating tablets in fed subjects, 1989. ISSN 03785173.

[45] J L Madsen. Effects of gender, age, and body mass index on gastrointestinal transit times, 1992. ISSN 0163-2116.

[46] J L Madsen and M Jensen. Gastrointestinal transit of technetium-99m-labeled cellulose fiber and indium-111-labeled plastic particles. *Journal of nuclear medicine : official publication, Society of Nuclear Medicine*, 30(3):402–406, 1989. ISSN 0161-5505.

[47] I. R. Wilding, S. S. Davis, M. Bakhshaee, H. N E Stevens, R. A. Sparrow, and J. Brennan. Gastrointestinal transit and systemic absorption of captopril from a pulsed-release formulation. *Pharmaceutical Research*, 9(5):654–657, 1992. ISSN 07248741. doi: 10.1023/A:1015806211556.

[48] L C Kaus, W R Gillespie, A S Hussain, and G L Amidon. The effect of in vivo dissolution, gastric emptying rate, and intestinal transit time on the peak concentration and area-under-the-curve of drugs with different gastrointestinal permeabilities. *Pharmaceutical research*, 16(2):272–280, 1999. ISSN 0724-8741. doi: 10.1023/A:1018836727001.

[49] Michael Davidson, Peter P. Toth, and Kevin C. Mak, editors. *Therapeutic Lipidology*. Springer Science & Business Media, Totowa, 2007.

[50] J P Deslypere. Clinical implications of the biopharmaceutical properties of fluvastatin. Technical Report 14, 1994.

[51] T. Kantola, J. T. Backman, M. Niemi, K. T. Kivistö, and P. J. Neuvonen. Effect of fluconazole on plasma fluvastatin and pravastatin concentrations. *European Journal of Clinical Pharmacology*, 56(3):225–229, 2000. ISSN 00316970. doi: 10.1007/s002280000127.

[52] F L Tse, J M Jaffe, and A Troendle. Pharmacokinetics of fluvastatin after single and multiple doses in normal volunteers. *Journal of clinical pharmacology*, 32(7):630–8, July 1992. ISSN 0091-2700. URL http://www.ncbi.nlm.nih.gov/pubmed/1640002.

[53] K. Kavitha, A. Srinivasa Rao, and C. N. Nalini. An investigation on enhancement of solubility of 5 fluorouracil by applying complexation technique-characterization, dissolution and molecular-modeling studies. *Journal of Applied Pharmaceutical Science*, 3(3):162–166, 2013. ISSN 22313354. doi: 10.7324/JAPS.2013.30330.

[54] Daniel B Longley, D Paul Harkin, and Patrick G Johnston. 5-fluorouracil: mechanisms of action and clinical strategies. *Nature reviews. Cancer*, 3(5):330–338, 2003. ISSN 1474175X. doi: 10.1038/nrc1074.

[55] T Phillips, A Howell, R J Grieve, and P G Welling. Pharmacokinetics of oral and intravenous fluorouracil in humans. *Journal of pharmaceutical sciences*, 69(12):1428–31, December 1980. ISSN 0022-3549. URL http://www.ncbi.nlm.nih.gov/pubmed/7463330.

[56] R L Schilsky. Biochemical and clinical pharmacology of 5-fluorouracil. *Oncology (Williston Park, N.Y.)*, 12(10 Suppl 7):13–18, 1998. ISSN 0890-9091.

[57] Peter M. Wigmore, Sarah Mustafa, Maha El-Beltagy, Laura Lyons, Jariya Umka, and Geoff Bennett. Effects of 5-FU. *Advances in Experimental Medicine and Biology*, 678:157–164, 2010. ISSN 00652598. doi: 10.1007/978-1-4419-6306-2_20.

[58] Sambamoorthy Bolla, Ramesh R Boinpally, Srinivasu Poondru, Rambhau Devaraj, and Bhaskara R Jasti. Pharmacokinetics of diethylcarbamazine after single oral dose at two different times of day in human subjects. *Journal of clinical pharmacology*, 42(3):327–31, March 2002. ISSN 0091-2700. URL http://www.ncbi.nlm.nih.gov/pubmed/11865970.

[59] J S Dixon, J M Borg-Costanzi, S J Langley, L F Lacey, and S Toon. The effect of renal function on the pharmacokinetics of ranitidine. *European journal of clinical pharmacology*, 46(2):167–171, 1994. ISSN 0031-6970. doi: 10.1007/BF00199883.

[60] G Edwards, A M Breckenridge, K K Adjepon-Yamoah, M L Orme, and S A Ward. The effect of variations in urinary pH on the pharmacokinetics of diethylcarbamazine. *British journal of clinical pharmacology*, 12(6):807–812, 1981. ISSN 03065251. doi: 10.1111/j.1365-2125.1981.tb01311.x.

[61] Zulfequar Ahamad Khan, Rahul Tripathi, and Brahmeshwar Mishra. Floating Elementary Osmotic Pump Tablet (FEOPT) for Controlled Delivery of Diethylcarbamazine Citrate: a Water-Soluble Drug, 2011. ISSN 1530-9932.

[62] P Layer, M V Singer, and V E Eysselein. Effect of circadian rhythm on gastrointestinal motility. *Zeitschrift fur Gastroenterologie*, 25 Suppl 3:69–73, 1987.

[63] J Keller, G Gröger, L Cherian, B Günther, and P Layer. Circadian coupling between pancreatic secretion and intestinal motility in humans. *American journal of physiology. Gastrointestinal and liver physiology*, 280(2):G273–G278, 2001. ISSN 01931857.

[64] Yamaguchi M, Kotani K, Tsuzaki K, Takagi A, Motokubota N, Komai N, Sakane N, Moritani T, and Nagai N. Circadian Rhythm Genes CLOCK and PER3 Polymorphisms and Morning Gastric Motility in Humans. page e0120009, 2015. ISSN 1932-6203.

[65] S Anderson and W W Hauck. A new procedure for testing equivalence in comparative bioavailability trials. *Communication in Statistics, Theory and Methods*, 12:2663–2692, 1983. ISSN 0361-0926. doi: 10.1080/03610928308828634.

[66] Sharon Anderson and Walter W. Hauck. Consideration of individual bioequivalence. *Journal of Pharmacokinetics and Biopharmaceutics*, 18(3):259–273, 1990. ISSN 0090166X. doi: 10.1007/BF01062202.

[67] Roger L. Berger, Jason C. Hsu, Walter W. Hauck, Sharon Anderson, Michael P. Meredith, Mark A. Heise, Jen-pei Liu, Shein-Chung Chow, Donald J. Schuirmann, J. T. Gene Hwang, Roger L. Berger, and Jason C. Hsu. Bioequivalence trials, intersection-union tests and equivalence confidence setsCommentCommentCommentCommentCommentRejoinder, 1996. ISSN 08834237.

[68] Walter W. Hauck and Sharon Anderson. A Comparison of Large-Sample Confidence Interval Methods for the Difference of Two Binomial Probabilities. *The American Statistician*, 40(4):318–322, 1986. ISSN 0003-1305. doi: 10.1080/00031305.1986.10475426. URL http://www.tandfonline.com/doi/abs/10.1080/00031305.1986.10475426.

[69] J P Liu and C S Weng. Estimation of direct formulation effect under log-normal distribution in bioavailability/bioequivalence studies. *Statistics in medicine*, 11(7):881–896, 1992. ISSN 02776715.

[70] K F Phillips. A log-normal model for individual bioequivalence. *Journal of biopharmaceutical statistics*, 3(2):185–201, 1993. ISSN 1054-3406. doi: 10.1080/10543409308835059.

[71] D J Schuirmann. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of pharmacokinetics and biopharmaceutics*, 15(6):657–680, 1987. ISSN 0090-466X. doi: 10.1007/BF01068419.

[72] JN Thompson, OJ Reichman, and PJ Morin. Frontiers of ecology. *BioScience*, 51(1):15–24, 2001. ISSN 16182642. doi: 10.1641/0006-3568. URL http://www.jstor.org/stable/10.1641/0006-3568(2001)051[0015:FOE]2.0.CO;2.

[73] Siyan Xu, Steven Y. Hua, Ronald Menton, Kerry Barker, Sandeep Menon, and Ralph B. D'Agostino. Inference of bioequivalence for log-normal distributed data with unspecified variances. *Statistics in Medicine*, 33(17):2924–2938, 2014. ISSN 02776715. doi: 10.1002/sim.6081. URL http://doi.wiley.com/10.1002/sim.6081.

CHAPTER II

Fasted State Dissolution and Transit of Ibuprofen

2.1 Introduction

Solid oral dosage forms must dissolve in the GI fluids prior to being absorbed and reaching systemic circulation. The rate and extent of this process depends not only on the physiochemical properties of the dosage form–including lipophilicity, chemical or enzymatic stability, solubility, particle size, density, diffusivity, pKa, and crystal form–but also the physiological environment[1]. Buffer species, bile salts, gastric emptying, intestinal motility and hydrodynamics, and pH all play significant roles in this process [2].

Several mass balance approaches have been previously described[3–7]. Ozturk et al. presented dissolution kinetics of ionizable drugs assuming the process to be diffusion-limited with instantaneous and reversible reactions[8]. However, many of these analysis do not take into account the dynamics of the GI tract, for example locational permeability profiles, environmental changes altering dissolution such as pH fluctuations; pancreatic and biliary secretions; large inter-individual differences in gastric emptying onset & rate; dynamic intestinal transit; and the effects these have on dissolution kinetics[9–14]. Indeed this is due to the complexity of underlying mechanisms as well as the difficulty in estimating adsorption from systemic availability[9].

2.1.1 Physiological properties of the gastrointestinal tract

Much of the dynamism in along the GI tract during the fasted state is due to cyclical fluctuations referred to as the migrating motor complex (MMC) which includes three distinct phases: the quiescent phase of little to no activity (phase I); a period of more frequent and gradually increasing amplitudinal contractions that allow for much back and forth but little net forward movement (phase II); and a short-lasting but high-frequency period of activity that propels content aborally (phase III)[15]. However, many other factors contribute to the mutable activeness of the GI tract. A summary of literature ranges for GI fluid composition, fluid volumes, and geometries can be found in Table 2.1.

2.1.1.1 Fluid composition and volumes

GI fluid is composed of many ingredients that change over time, including saliva, secretory fluids, and refluxed duodenal fluid. Among these, bicarbonate is one of the major intestinal buffer species maintaining a pH gradient along the GI tract and protecting the gastric mucosal layer against acid and proteolytic digestion by pepsin [16–18]. Bicarbonate concentration in the human small intestine ranges from 6-20 mM implying a buffer capacity of 2.5-2.8 per pH[19–23]. Crucial to drug product dissolution, the pH is highly variable both locally and along the entire GI tract. During the fasted state, the gastric pH can vary between 1 and 8 with the mean generally in the range of 1 to 2[21,24–26], while small bowel pH has been shown to vary between 4 and 8 with a typical mean around 6.5 in the proximal regions[21,22,27–31] and 6.5-8 in the distal regions[32,33]. Closely linked to gastric acid secretion is the activity of pepsin which can affect protein and peptide stability. Once activated at low pH, it aids in proteolytic breakdown of dietary proteins, and it is deactivated at pH above 4[34–36]. Lipase is pertinent especially for lipid-based formulations and crucial in the absorption of dietary fats and may not contribute significantly in the fasted state due to its low concentrations[37–39]. In addition, mucus production and secretion help prevent physical injury; oxyntic and pyloric glands of the stomach produce gastric juices to further aid in digestion; bile produced from the liver solubilizes fats; pancreatic juice released into the duodenum neutralizes acidic contents entering from the stomach; and the endogenous bacterial populations release enzymes promoting metabolism[40].

Liquid volumes along the GI have an unequivocal influence on drug dissolution. Indeed, the extent to which this occurs and thus the capacity for absorption and systemic availability is largely affected by both inter- and intra-patient variability[37]. The fasted stomach resting volume has been measured between 18-54 mL[41,42]. Following ingestion of a 240 mL liquid volume and subsequent gastric emptying, Mudie et al. showed, using magnetic resonance imaging, that subjects returned to approximately their initial resting volumes[43], driven presumably by the contribution of increased gastric secretion to compensate for the augmented emptying. The reported range of liquid volume in the fasted small intestine spans an order of magnitude, from of 30-420 mL, while several studies showed it was around 100 mL on average[44,45]. Perhaps most importantly for drug dissolution, Schiller et al. and Mudie et al. reported that that fluid in the GI is not evenly distributed but rather in discrete packets of varying volumes[43,45]. It is not hard to imagine how this might impact a transiting dosage form: variable gastric emptying and random fluid pocket formations can lead to radically different local environments in terms of volume, hydrogen ion concentration, and fluid composition.

2.1.1.2 Geometry

Absorption is in large part controlled by the GI surface area to which it is exposed, with a larger surface area potentiating greater absorption. The extent of drug absorption in the stomach is generally negligible as compared to the small intestine, presumably due to the much larger surface area and longer residence times[46]. However, the stomach and its associated hydrodynamics is important in the disintegration and dissolution process for many oral dosage formulations. Using ultrasonography to map the stomach in 3D, Liao et al. measured the luminal volume and inner surface area for the whole stomach as 277.6 cm^3 and 196.3 cm^2, respectively, and 181.6 cm^3 and 125.3 cm^2, respectively, for the gastric antrum[47]. In contrast to the stomach, the radius of the human small intestine is 4 cm at its junction with the stomach whence it narrows progressively–thereby also altering the surface-to-area ratio–to 1-2 cm at the ileocecal valve[48–50]. The lengths of the duodenum, jejunum, and ileum are 21, 105, and 156 cm, respectively, totaling a living physiological length of 282 cm. The complex geometry in the small intestine, with villi extending from the surface each containing their own smaller microvilli extensions, expands the surface area between 10^4 - 2 x 10^6 cm^2 as compared to the smooth surface of a flat tube with the same length and diameter[50]. This 3800-fold increase in small intestinal surface area relative to that of the stomach elucidates preferential absorption in the small intestine[40], however the unstirred water layer of the mucosal surface still presents resistance even for highly permeable drugs[51] (although this might be overestimated and resistance may rather be membrane controlled irrespective of transport mechanism[52]).

2.2 Methods

Herein is presented a dissolution and disintegration mass balance analysis. Incorporating this with the GI transit analysis from Chapter I, *in vivo* distribution of drug concentrations along the GI tract is predicted based on motility phase and drug product physiochemical properties. Ibuprofen as well as the tracer dye phenol red are used as example cases.

Chemical properties of drug products dictate their dissolution and disintegration performances under varying physiological conditions. Figure 2.1 illustrates the a schematic for synthesizing disintegration, dissolution, and motility. A drug is dosed orally with a volume of liquid (here the FDA-required 240 mL) and begins disintegrating and, depending on its physiochemical properties, dissolving in the stomach. Horizontal arrows to the right represent disintegration and dissolution. The dissolving particles and fluid transit along the GI tract, represented here by the vertical arrows pointing downward. Several assumptions are made:

- A disintegrating formulation separates into identical, monodispersed particles with starting radius r_0.
- Gastric secretion is driven solely by the resting volume, current stomach volume, and emptying rate.
- The pH, buffering capacity, and effective permeation can vary between compartments, however they remain constant (i.e. unaffected by dissolving drug).
- The system is well-buffered and stirred but with non-sink conditions (non-negligible bulk concentration of dissolved drug).
- Larger particles are emptied more slowly from the stomach compartment[53,54], illustrated in Figure 2.1 as the light green arrow showing size-dependent transit of the solid drug into the intestinal compartments.

2.2.1 Fluid and particle transit

The cyclical, time-dependent transit model described in Equations 1.1 and 1.2 are used for the flow of fluid and particles suspension between compartments. Briefly, a Fourier series approximation is used to represent the cyclical gastric emptying rate function and lag time is described by a sigmoidal decay function, with 7 intestinal compartments, each a continuously-stirred reactor tank where influx and outflux are balanced. Drug and fluid enter the stomach at dose time t_0 and empty according to the cyclical, time-dependent emptying rate function, into sequential intestinal compartments with two outflows each–an effective permeation rate into the plasma compartment and an intestinal transit rate (equal to a phase shifting of the gastric emptying rate, i.e. $k_{int_N}(t) = k_{ge}(t + \tau)$ for some time offset τ in intestinal compartment N). This is equivalent to a contractile wave of the MMC propagating along the GI tract.

2.2.1.1 Gastric secretion

To account for resting volumes in the stomach and small intestines, gastric fluid secretion occurs relative to the current gastric volume and gastric emptying rate:

$$k_{gs} = \frac{2v_r k_{ge}}{v_g + v_r} \tag{2.1}$$

When the current volume in the stomach is greater than the resting volume ($v_g > v_r$), the gastric secretion rate is less than the gastric emptying rate ($k_{gs} < k_{ge}$) resulting in net outflow of fluid. Conversely, when the current volume is less than the resting volume ($v_g < v_r$) the gastric fluid is replenished with greater gastric secretion than emptying ($k_{gs} > k_{ge}$). The relationship between gastric secretion, emptying, and volume is illustrated in Figure 2.2.

2.2.1.2 Particle size effect on gastric emptying

Experimental results suggest particles larger than 2 mm in diameter tend to be retained longer in the stomach, allowing for peristaltic back-and-forth mixing to further break them down[53]. Almost all content, however, transits out of the stomach during the powerful contractions of phase III. A dampening of the gastric emptying rate function is used to reflect the size dependence, described by a logistic function(Figure 2.3).

2.2.1.3 Effective boundary layer

The effective boundary layer for dissolving particles is determined by a given critical radius r_c, described by Wang and Flanagan (Equation 2.2)[55]. Previously, Hintz and Johnson suggested a critical radius of 30 μm based on rotating disk experiments, and below this they treated the effective boundary layer linearly with respect to particle radius. Here, the effective boundary layer is equal to its radius for small particles, however for larger particles where $r \geq r_c$, the effective boundary layer remains constant. This is described by a continuous function:

$$\frac{1}{h_{eff}} = \frac{1}{1/r + 1/r_c} \tag{2.2}$$

2.2.2 Disintegration and dissolution mass balance

Flux from the particle surface depends on the product solubility S_T, particle radius r, diffusion coefficient in water D, bulk concentration C_b and pH, as described by the Nernst and Brunner equation[57]:

$$J = \frac{D}{h_{eff}}(S_{T,pH} - C_b) \tag{2.3}$$

The mass release rate across a surface is thus related by the flux and the surface area A, $dM/dx = A \cdot J$. While the pH and thus predicted solubility in a compartment remain constant, the bulk concentration C_b determines the extent of dissolution and indeed potential precipitation.

2.2.3 Ibuprofen and phenol red

Physiochemical properties of ibuprofen and phenol red are summarized in Tables 2.2 & 2.3, respectively. Phenol red is a non-absorbable dye that can be used to monitor the net concentration change along the GI tract, revealing information about volumetric flow. Ibuprofen is a well-characterized, weakly acidic BCS Class II non-steroidal anti-inflammatory drug that is poorly soluble and readily absorbable[58]. It is here used as an example case for the analysis. Ibuprofen dissolution is determined based on intrinsic solubility, its pKa, and the environmental pH. Potthast et al., Levis et al., and Shaw et al. reported the pH-dependent solubility of ibuprofen (Figure 2.6), and while three experimental points deviate toward a plateau (potentially due to the salt effect where dissolution is driven by the solubility product K_{sp}[61]) , this study bases solubility prediction on the experimental results of Shaw et al.[58–60].

The stomach resting volume is fixed at 30 mL, while the small intestine has a total of 70 mL divided between the 7 compartments. This reflects the low reported intestinal volumes[43]. Disintegration for ibuprofen, treated as a first order process, is extremely rapid (90% within 3 minutes), and the initial particle radius once disintegration has commenced is 400 μm. Permeation is constant at 8.0 x 10^{-4} cm^2/s along the entire small intestine (all compartments have same p_{eff}). The plasma elimination half-life is 2.25 hours. The initial dose is either 200, 400, or 800 mg for ibuprofen and 65 mg for phenol red.

2.3 Results and discussion

The phenol red is introduced at dose time t_0 (relative to the motility state) into the stomach compartment as a fully dissolved solution (65 mg in 240 mL) and thus there is no disintegration or dissolution taking place. The solution transits inter-compartmentally based on the contractile activity influencing the rates $k_{ge}(t)$ and $k_{int_N}(t)$. Figure 2.7 illustrates the effect of motility phase on the distribution of the phenol red marker along the GI tract. During phase I, there is a transit-related lag from when the solution is introduced in the stomach until it appears in the first intestinal compartment, as well as a lag until it first arrives in the final intestinal compartment. Progressing through phase II and III, these lags are shortened as the gastric emptying rate increases and phenol red solution appears almost immediately in the first intestinal compartment and transits quickly until the end. However, in late phase III, there is an initial quick release from the stomach but the emptying rate is soon reduced as the contractile pattern cycles back to phase I, thereby re-introducing a lag. This latency results in even longer total transit time and it is not until nearly an hour and half post dose that the solution appears in the distal intestinal compartments.

Applying the GI transit analysis and dissolution mass balance to the specific case of ibuprofen, the plasma profiles show clear dependence on motility phase for 200, 400, and 800 mg doses (Figure 2.8). Later dose times with respect to motility result in faster transit and thus earlier absorption, shifting the T_{max} earlier. However, a late phase III dose time causes a delay of up to 50 minutes in the T_{max}. Ibuprofen transit along the intestinal GI tract shows detectable concentrations transiting into the last compartments and even undissolved particles in the distal region (Figure 2.9). Initially, simulations were done using larger intestinal volumes and particle dissolution occurred almost entirely in the first intestinal compartments, resulting in much earlier T_{max} predictions. However, reflecting findings that showed much lower intestinal volumes[43], dissolution is predicted to be a rate-limiting step in the systemic appearance of ibuprofen. Similar to the phenol red transit, early dosing in the cycle has a longer lag time before appearance in the intestines while later dosing results in shortening of the lag time. However, dosing in late phase III results in longer gastric residence time and slower transit through the intestinal compartments. This allows for a greater extent of dissolution, and so fewer particles are seen distally. The rate of absorption is also faster here than transit, and so the distal ibuprofen concentration is predicted to be less than when dosed earlier in the cycle.

As further valuation, the distribution of predicted bioequivalence metrics is considered for ibuprofen with 200, 400, and 800 mg doses. Using the dissolution and transit model, each simulation is carried out with the given dose administered with a 240 mL fluid volume in accordance with FDA guidelines. The dose time is varied to reflect the random nature of the underlying MMC. Resultant maximum plasma concentrations C_{max}, peak plasma concentration times T_{max}, and bioavailability AUC are predicted in accordance with reported values (Figure 2.10)[62]. Literature values encompass a very broad range of subjects who differ greatly in weight, age, and inherent physiological parameters. This accounts for greater variation, in particular for bioavailability, in the experimental results whereas the variation in predicted results is due solely to simulating over the possible range of dose times relating to motility phase and assuming a 70 kg subject with a fixed volume of distribution.

2.4 Conclusion

Over the course of several decades, researchers have worked toward developing mechanistic absorption models for oral drug products. Such prediction relies on the ability to determine a drug's fate in the GI tract. The coupling of a pH-dependent, non-sink condition dissolution model with the cyclical, motility-driven gastrointestinal transit model is an important step toward a more accurate physiological analysis of oral drug products. Using the non-absorbable phenol red dye, transit based on fluid flow is simulated showing a decidedly motility-dependent lag in distal intestinal appearance of fluid. Ibuprofen is used as an example case, showing strong accord with reported pharmacokinetics values. There remains a testable hypothesis of prolonged particulate matter transiting through much of the small intestine due to the small reported fluid volumes.

Applying these methods to current and future pharmaceutical products should lead to successful prediction of *in vivo* pharmacokinetic profiles thereby simplifying tests and regulatory burdens‘ as well as allow industry to incorporate product changes in a timelier manner. While there remains much to refine based on forthcoming experimental work, the compartmental transit model has been at the foundation of *in vivo* predictive dissolution and absorption of oral drug products and will continue to be at the forefront of mechanistic drug product development and evaluation of bioequivalence standards.

2.5 Tables and figures

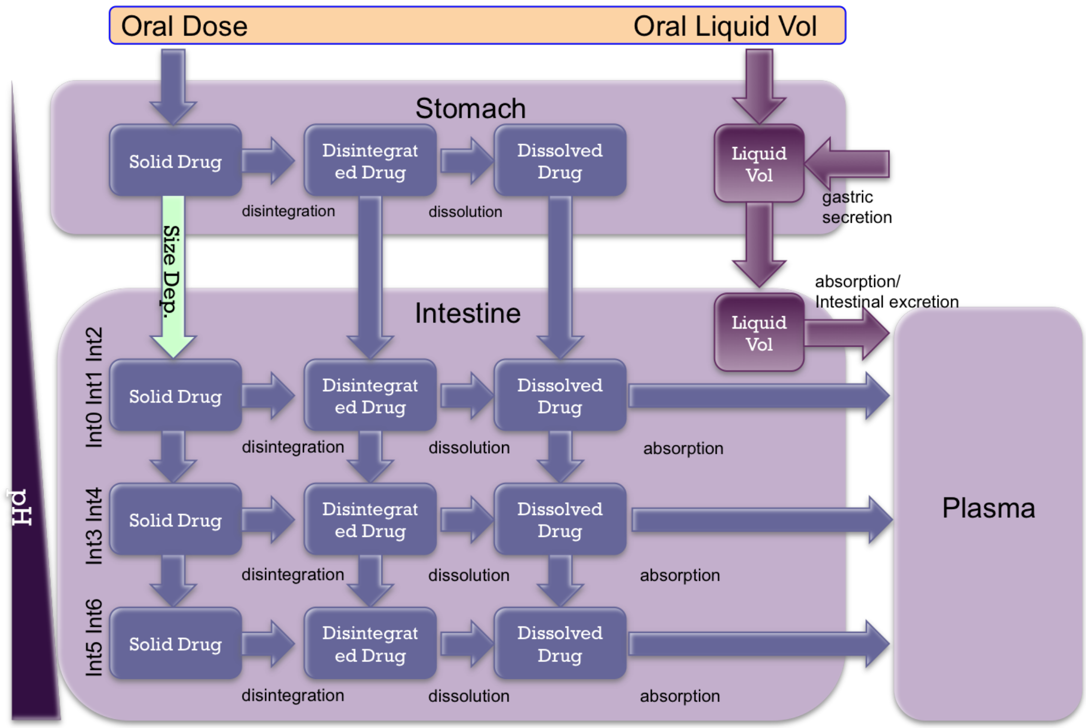


Figure 2.1: Incorporating dissolution mass balance into the GI transit system. A solid dose is administered with a liquid volume. Gastric emptying drives fluid and solid out of the stomach. Fluid is replenished via gastric secretion. Disintegration and dissolution take place in all compartments based on the physiological environment and drug physiochemical properties.

Table 2.1: GI fluid component concentrations & properties, liquid volumes, and geometry during the fasted state[37]

	Stomach	Duodenum	Jejunum	Ileum
Bicarbonate (mEq L^{-1})	7-20[63,64]	2.7-15[65–67]	2-20[68–70]	40-75[66,71–74]
Bile salts (mM)	0.08-0.275[30,75–77]	0.3-9.6[2,21,22,30,76,78,79]	0-17[2,22,23,26]	2-10[80]
Calcium (mM)	0.6±0.2[26]		0.5±0.3[26]	
Chloride (mM)	102±28[26]		126-135[26,71]	125±12[71]
Lipase (mg/mL)	0.1[38]			
Lipids (mg/mL)	0.56[78]	0-1.8[30,78]	0.1±0.001[23]	
Pepsin (mg/mL)	0.11-1.27[21,81,82]			
Phospholipids (mM)		0.8-2.4[30]	3±0.3[23]	
Potassium (mM)	13.4±3.0[26]		4.8-5.4[26,71]	4.9±1.5[71]
Sodium (mM)	68±29[26]		142±13[26,71]	140±6[71]
Buffer capacity (mmol L^{-1} pH^{-1})	7-18[21]	4-13[21,22]	2.4-3.23[23,83]	6.4[83]
Osmolality (mOsm kg^{-1})	29-276[21,26,68,84,85]	124-266[21,22,30,84]	200-278[22,26,85]	
pH	1-7.5[21,24–26]	4.0-7.0[21,22,27–31]	4.4-8.1[22–24,26,86]	6.5-6.8[32,33]
Surface tension (mN m^{-1})	41.9-45.7[21]	32.3-46.0[21,30]	28-33.7[23,85]	
Diameter (cm)	4-37[87,88]	3.5-6[87,88]	2.5-5[87,88]	2-5[87,88]
Length (cm)		18-30[88]	105-1128[88]	156-395[88]
Surface area (cm^2)	525.58-1100[87]	$9.0 \cdot 10^3 - 1.2 \cdot 10^4$[87]	$6 \cdot 10^5$[87]	$6 \cdot 10^6$[87]
Volume (mL)	18-54[41,42]	34-319[44,45]		

Table 2.2: Ibuprofen physiochemical properties

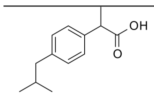
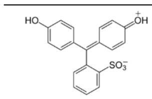
	
Molecular mass	206.3 g/mol
Diffusion coefficient in water	7.5E-6 cm^2/s
Intrinsic Solubility (37°C)	0.066 mg/mL
pKa (37°C)	4.4
LogP	3.84
Density	1.1 g/cm^3
Plasma elimination half-life	1.3-3 hrs
Disintegration	90% in 2 min

Table 2.3: Phenol red physiochemical properties

	
Molecular mass	354.38 g/mol
Diffusion coefficient in water	7.5E-6 cm^2/s
Solubility	0.77 mg/mL
pKa (37°C)	8.0
LogP	4.11
Density	1.5 g/cm^3

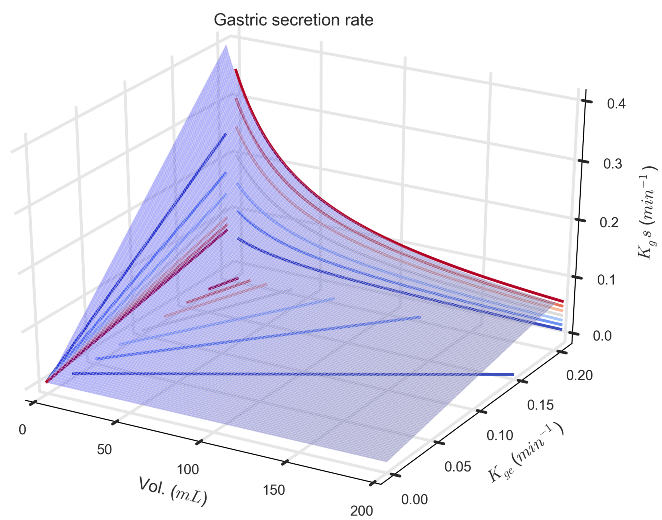


Figure 2.2: Visualizing gastric secretion rate relative to stomach volume and gastric emptying rate.

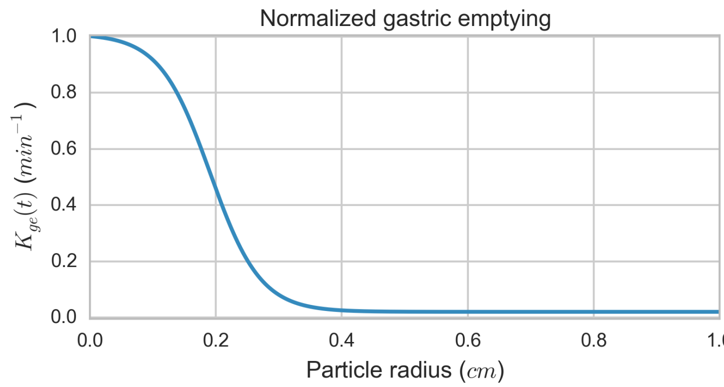


Figure 2.3: The effect of particle size on emptying rate.

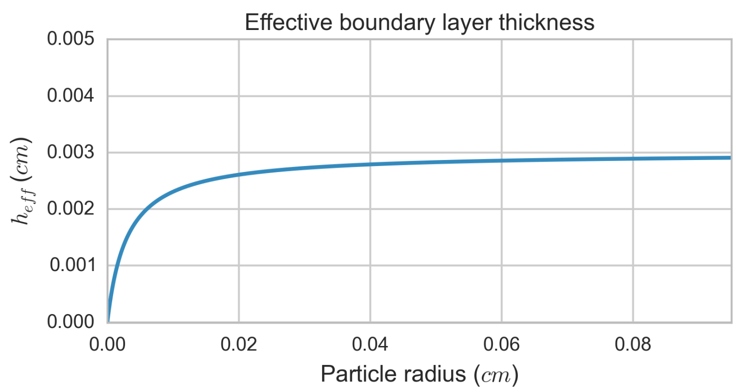


Figure 2.4: Effective boundary layer thickness h_{eff} (Equation 2.2).

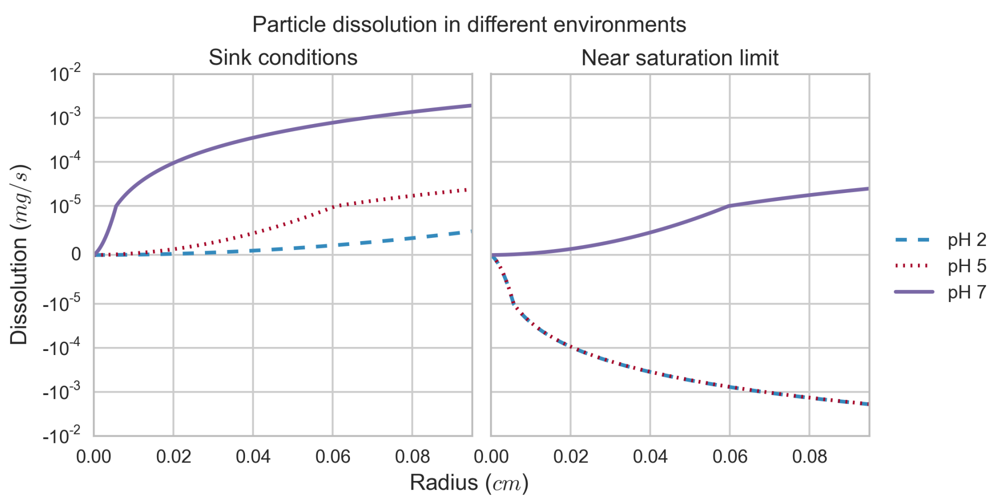


Figure 2.5: Ibuprofen dissolution under different conditions.

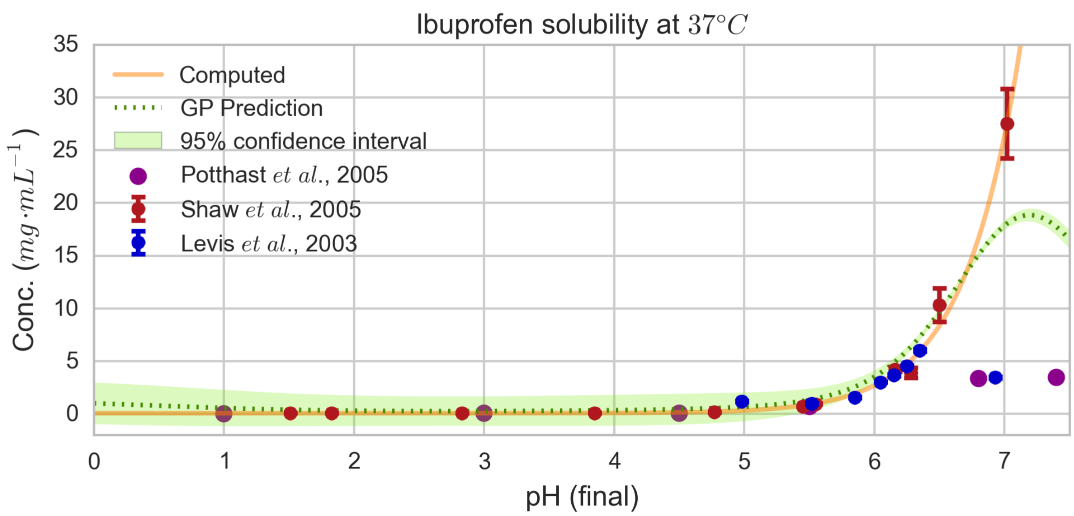
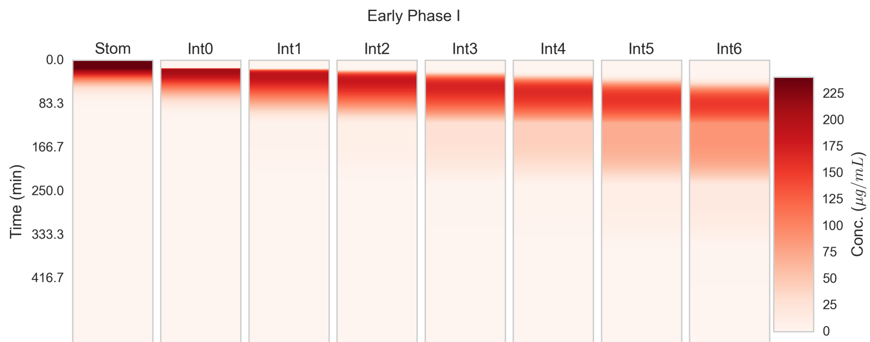
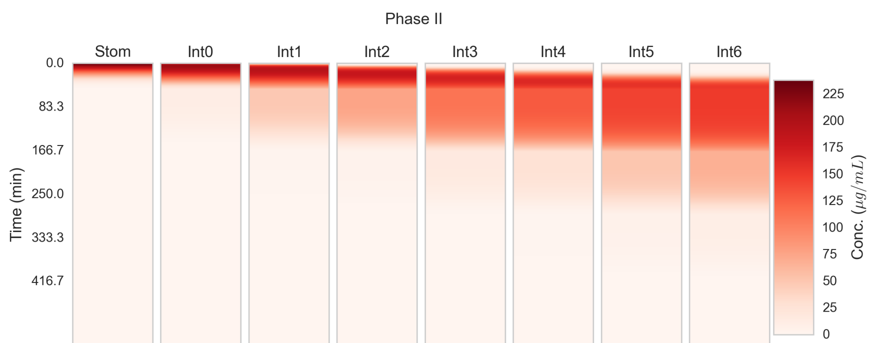


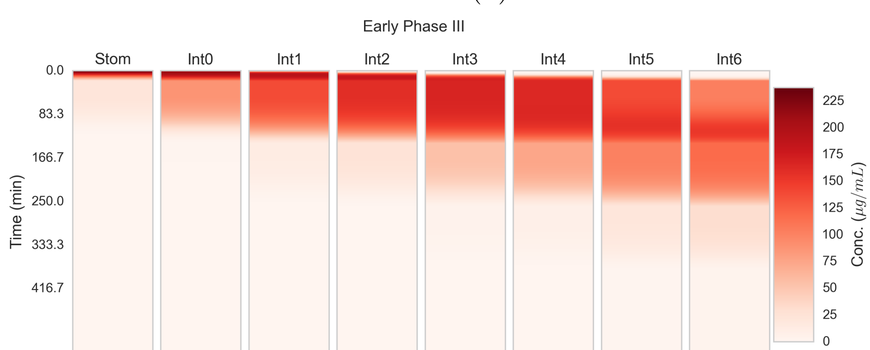
Figure 2.6: Reported[58–60] and calculated ibuprofen solubility.



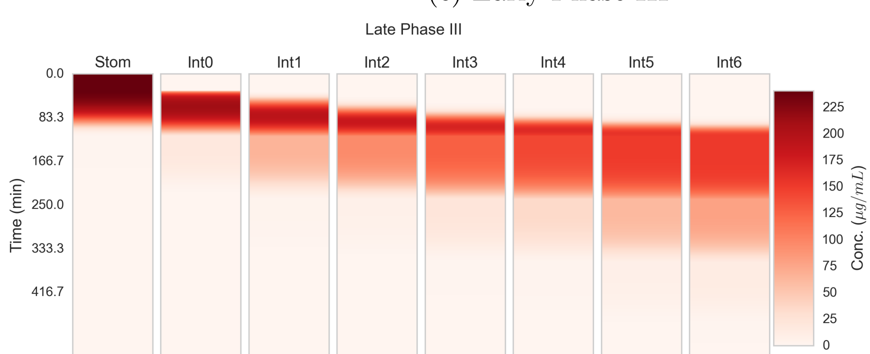
(a) Early Phase I



(b) Phase II



(c) Early Phase III



(d) Late Phase III

Figure 2.7: Phenol red solution transit through GI tract when dosed during (a) early phase I; (b) phase II; (c) early phase III; and (d) late phase III.

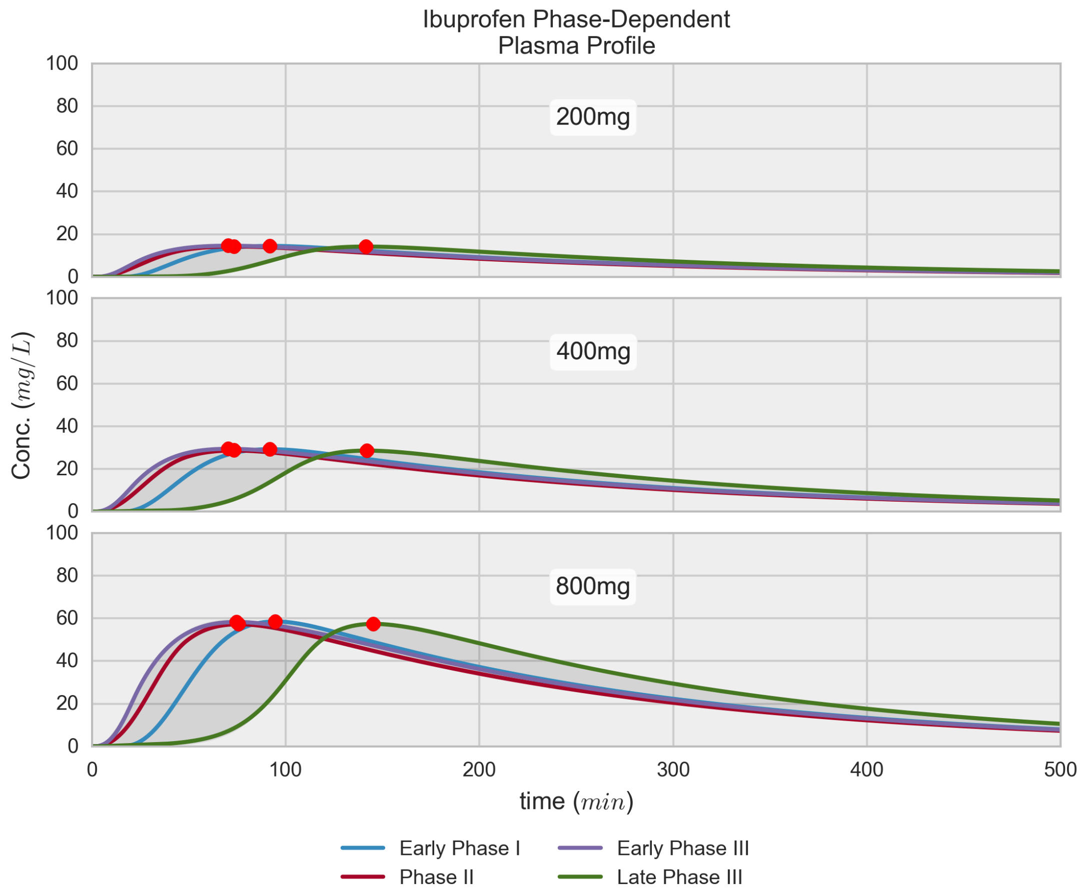
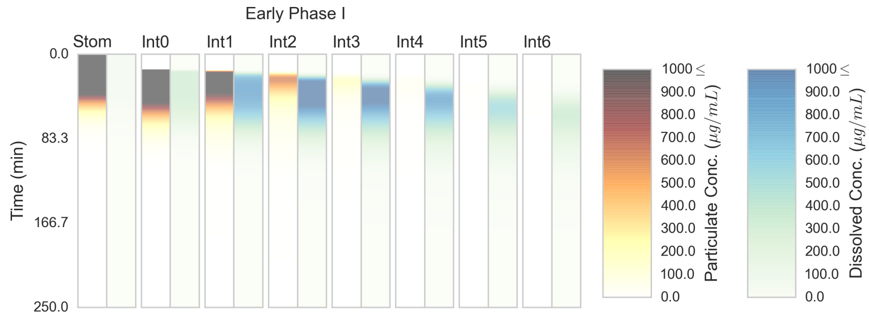
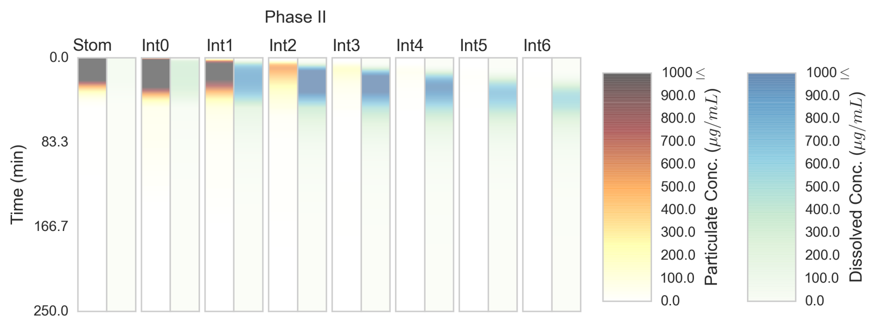


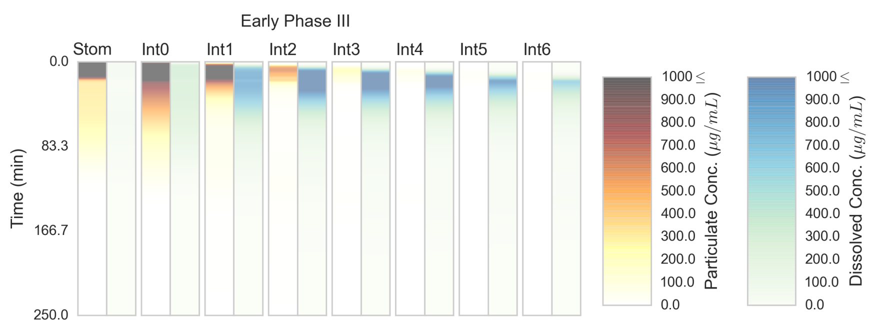
Figure 2.8: Ibuprofen plasma profiles for early phase I (blue), phase II (red), early phase III (purple), and late phase III (green) showing the variation in C_{max} and especially T_{max}.



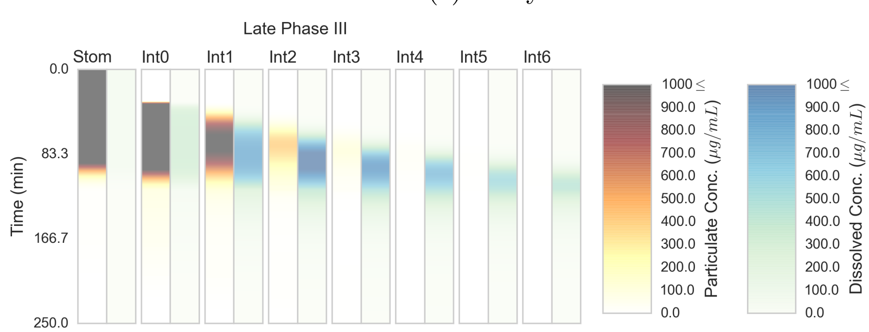
(a) Early Phase I



(b) Phase II



(c) Early Phase III



(d) Late Phase III

Figure 2.9: Ibuprofen particulate and solution transit through GI tract when dosed during (a) early phase I; (b) phase II; (c) early phase III; and (d) late phase III.

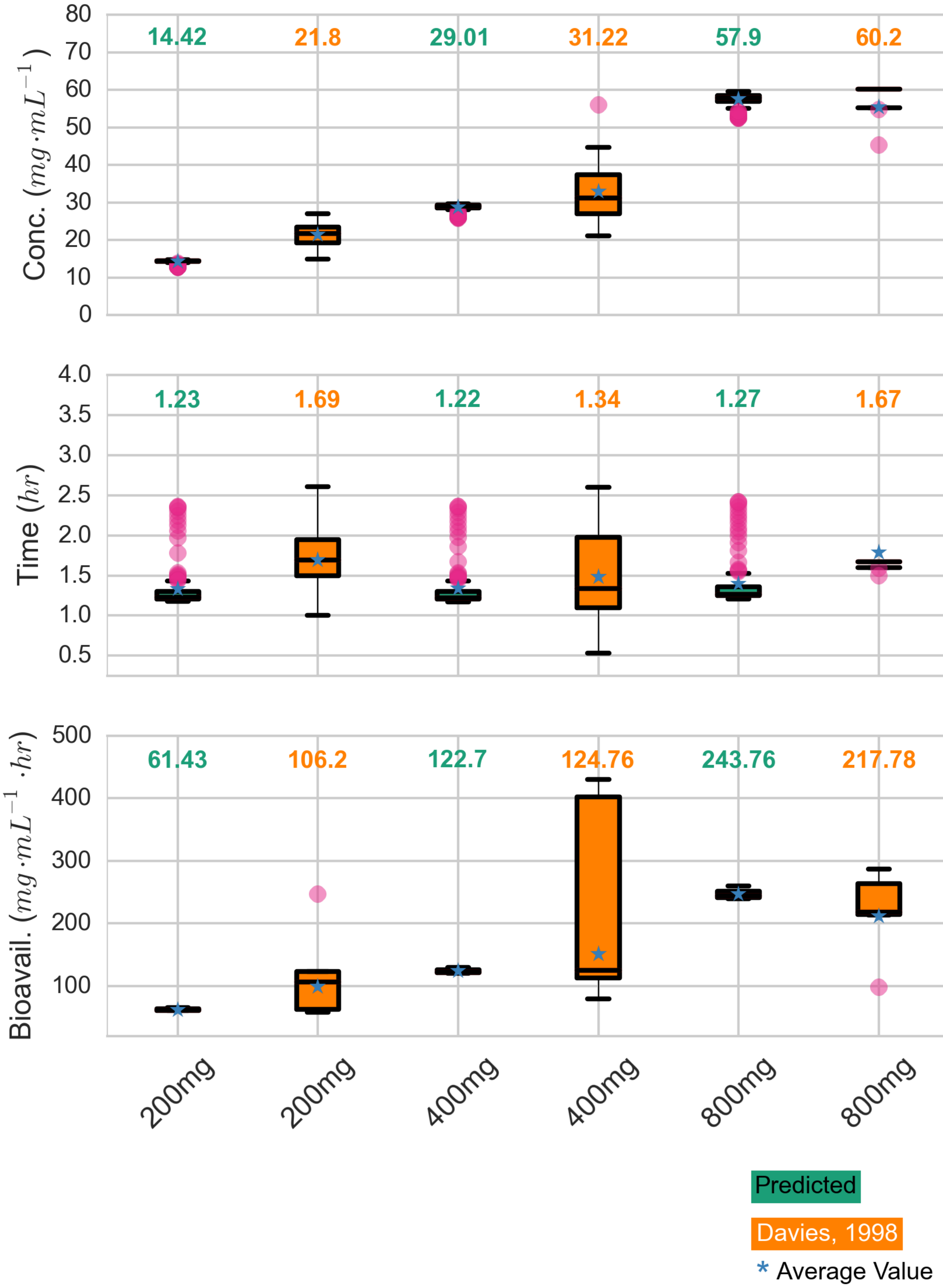


Figure 2.10: Predicted (green) and experimental (orange) values for C_{max} (top), T_{max} (middle), and AUC (bottom) for 200, 400, and 800mg doses of ibuprofen. The colored boxes represent the 25-75 percentiles, the whiskers span the entire range, and the fuchsia circles are outliers.

2.6 References

[1] D. M. Oh, R. L. Curl, and G. L. Amidon. Estimating the fraction dose absorbed from suspensions of poorly soluble compounds in humans: A mathematical model. *Pharmaceutical Research*, 10(2):264–270, 1993. ISSN 07248741. doi: 10.1023/A:1018947113238.

[2] Jennifer B. Dressman, Gordon L. Amidon, Christos Reppas, and Vinod P. Shah. Dissolution testing as a prognostic tool for oral drug absorption: Immediate release dosage forms, 1998. ISSN 07248741.

[3] J B Dressman, D Fleisher, and G L Amidon. Physicochemical model for dose-dependent drug absorption. *Journal of pharmaceutical sciences*, 73(9):1274–1279, 1984. ISSN 0022-3549.

[4] J B Dressman, G L Amidon, and D Fleisher. Absorption potential: estimating the fraction absorbed for orally administered compounds. *Journal of pharmaceutical sciences*, 74(5):588–589, 1985. ISSN 00223549. doi: 10.1002/jps.2600740523.

[5] B C Goodacre and P J Murray. A mathematical model of drug absorption. *Journal of clinical and hospital pharmacy*, 6(2):117–133, 1981. ISSN 01433180.

[6] Norman F.H. Ho, Hans P. Merkle, and William I. Higuchi. Quantitative, mechanistic and physiologically realistic approach to the biopharmaceutical design of oral drug delivery systems. *Drug Development and Industrial Pharmacy*, 9(7):1111–1184, 1983. ISSN 0363-9045. doi: 10.3109/03639048309046315. URL `http://www.informapharmascience.com/doi/abs/10.3109/03639048309046315`.

[7] P J Sinko, G D Leesman, and G L Amidon. Predicting fraction dose absorbed in humans using a macroscopic mass balance approach. *Pharmaceutical research*, 8(8):979–988, 1991. ISSN 0724-8741. doi: 10.1023/a:1015892621261.

[8] S S Ozturk, B O Palsson, and J B Dressman. Dissolution of ionizable drugs in buffered and unbuffered solutions. *Pharmaceutical research*, 5(5):272–282, 1988. ISSN 0724-8741.

[9] G L Amidon, H Lennernäs, V P Shah, and J R Crison. A theoretical basis for a biopharmaceutic drug classification: the correlation of in vitro drug product dissolution and in vivo bioavailability. *Pharmaceutical research*, 12(3):413–420, 1995. ISSN 0724-8741. doi: 10.1023/A:1016212804288.

[10] J A Clements, R C Heading, W S Nimmo, and L F Prescott. Kinetics of acetaminophen absorption and gastric emptying in man. *Clinical pharmacology and therapeutics*, 24(4):420–431, 1978. ISSN 0009-9236.

[11] Shunji Haruta, Norio Iwasaki, Ken Ichi Ogawara, Kazutaka Higaki, and Toshikiro Kimura. Absorption behavior of orally administered drugs in rats treated with propantheline. *Journal of Pharmaceutical Sciences*, 87(9):1081–1085, 1998. ISSN 00223549. doi: 10.1021/js980117+.

[12] K Higaki, S Yamashita, and G L Amidon. Time-dependent oral absorption models. *Journal of pharmacokinetics and pharmacodynamics*, 28(2):109–28, April 2001. ISSN 1567-567X. URL `http://www.ncbi.nlm.nih.gov/pubmed/11381566`.

[13] Kazutaka Higaki, Sally Y Choe, Raimar Löbenberg, Lynda S Welage, and Gordon L Amidon. Mechanistic understanding of time-dependent oral absorption based on gastric motor activity in humans. *European journal of pharmaceutics and biopharmaceutics : official journal of Arbeitsgemeinschaft für Pharmazeutische Verfahrenstechnik e.V*, 70(1):313–25, September 2008. ISSN 0939-6411. doi: 10.1016/j.ejpb.2008.02.022. URL `http://www.ncbi.nlm.nih.gov/pubmed/18434110`.

[14] V S Patel and W G Kramer. Allopurinol absorption from different sites of the rat gastrointestinal tract. *Journal of pharmaceutical sciences*, 75(3):275–277, 1986. ISSN 00223549.

[15] P Kerlin, A Zinsmeister, and S Phillips. Relationship of motility to flow of contents in the human small intestine. *Gastroenterology*, 82(4):701–706, 1982. ISSN 00165085. doi: S0016508582000699[pii].

[16] Adrian Allen and Gunnar Flemström. Gastroduodenal mucus bicarbonate barrier: protection against acid and pepsin. *American journal of physiology. Cell physiology*, 288(1):C1–C19, 2005. ISSN 0363-6143. doi: 10.1152/ajpcell.00102.2004.

[17] Brian J. Krieg, Seyed Mohammad Taghavi, Gordon L. Amidon, and Gregory E. Amidon. In Viv o Predictive Dissolution: Transport Analysis of the CO 2 , Bicarbonate In Vivo Buffer System. *Journal of Pharmaceutical Sciences*, 103(11):3473–3490, 2014. ISSN 00223549. doi: 10.1002/jps.24108. URL `http://doi.wiley.com/10.1002/jps.24108`.

[18] Jennifer J. Sheng, Daniel P. McNamara, and Gordon L. Amidon. Toward an In Vivo dissolution methodology: A comparison of phosphate and bicarbonate buffers. *Molecular Pharmaceutics*, 6(1):29–39, 2009. ISSN 15438384. doi: 10.1021/mp800148u.

[19] Christel A S Bergström, René Holm, Søren Astrup Jø rgensen, Sara B E Andersson, Per Artursson, Stefania Beato, Anders Borde, Karl Box, Marcus Brewster, Jennifer Dressman, Kung I. Feng, Gavin Halbert, Edmund Kostewicz, Mark McAllister, Uwe Muenster, Julian Thinnes, Robert Taylor, and Anette Mullertz. Early pharmaceutical profiling to predict oral drug absorption: Current status and unmet needs. *European Journal of Pharmaceutical Sciences*, 57 (1):173–199, 2014. ISSN 18790720. doi: 10.1016/j.ejps.2013.10.015.

[20] Alexander Fuchs and Jennifer B. Dressman. Composition and Physicochemical Properties of Fasted-State Human Duodenal and Jejunal Fluid: A Critical Evaluation of the Available Data. *Journal of Pharmaceutical Sciences*, 103 (11):3398–3411, 2014. ISSN 00223549. doi: 10.1002/jps.24183. URL `http://doi.wiley.com/10.1002/jps.24183`.

[21] Lida Kalantzi, Konstantinos Goumas, Vasilios Kalioras, Bertil Abrahamsson, Jennifer B. Dressman, and Christos Reppas. Characterization of the human upper gastrointestinal contents under conditions simulating bioavailability/bioequivalence studies. *Pharmaceutical Research*, 23(1):165–176, 2006. ISSN 07248741. doi: 10.1007/s11095-005-8476-1.

[22] Mariangeles Perez de la Cruz Moreno, Marianne Oth, Sven Deferme, Frank Lammert, Jan Tack, Jennifer Dressman, and Patrick Augustijns. Characterization of fasted-state human intestinal fluids collected from duodenum and jejunum. *The Journal of pharmacy and pharmacology*, 58(8):1079–1089, 2006. ISSN 0022-3573. doi: 10.1211/jpp.58.8.0009.

[23] Eva M Persson, Ann-Sofie Gustafsson, Anders S Carlsson, Ralf G Nilsson, Lars Knutson, Patrick Forsell, Gunilla Hanisch, Hans Lennernäs, and Bertil Abrahamsson. The effects of food on the dissolution of poorly soluble drugs in human and in model small intestinal fluids. *Pharmaceutical research*, 22(12):2141–2151, 2005. ISSN 0724-8741. doi: 10.1007/s11095-005-8192-x.

[24] J. B. Dressman, R. R. Berardi, L. C. Dermentzoglou, T. L. Russell, S. P. Schmaltz, J. L. Barnett, and K. M. Jarvenpaa. Upper gastrointestinal (GI) pH in young, healthy men and women. *Pharmaceutical Research*, 7(7):756–761, 1990. ISSN 07248741. doi: 10.1023/A:1015827908309.

[25] D F Evans, G Pye, R Bramley, A G Clark, T J Dyson, and J D Hardcastle. Measurement of gastrointestinal pH profiles in normal ambulant human subjects. *Gut*, 29(8):1035–1041, 1988. ISSN 0017-5749. doi: 10.1136/gut.29.8.1035.

[26] Anders Lindahl, Anna Lena Ungell, Lars Knutson, and Hans Lennernäs. Characterization of fluids from the stomach and proximal jejunum in men and women. *Pharmaceutical Research*, 14(4):497–502, 1997. ISSN 07248741. doi: 10.1023/A:1012107801889.

[27] Pieter Annaert, Joachim Brouwers, Ann Bijnens, Frank Lammert, Jan Tack, and Patrick Augustijns. Ex vivo permeability experiments in excised rat intestinal tissue and in vitro solubility measurements in aspirated human intestinal fluids support age-dependent oral drug absorption. *European Journal of Pharmaceutical Sciences*, 39(1-3):15–22, 2010. ISSN 09280987. doi: 10.1016/j.ejps.2009.10.005.

[28] A Benn and W T Cooke. Intraluminal pH of duodenum and jejunum in fasting subjects with normal and abnormal gastric or pancreatic function. *Scandinavian journal of gastroenterology*, 6(4):313–317, 1971. ISSN 0036-5521. doi: 10.3109/00365527109181126.

[29] Jason Bratten and Michael P Jones. Prolonged recording of duodenal acid exposure in patients with functional dyspepsia and controls using a radiotelemetry pH monitoring system. Technical Report 6, 2009.

[30] S. Clarysse, J. Tack, F. Lammert, G. Duchateau, C. Reppas, and P. Augustijns. Postprandial evolution in composition and characteristics of human duodenal fluids in different nutritional states. *Journal of Pharmaceutical Sciences*, 98 (3):1177–1192, 2009. ISSN 00223549. doi: 10.1002/jps.21502.

[31] L Ovesen, F Bendtsen, U Tage-Jensen, N T Pedersen, B R Gram, and S J Rune. Intraluminal pH in the stomach, duodenum, and proximal jejunum in normal subjects and patients with exocrine pancreatic insufficiency. *Gastroenterology*, 90(4):958–962, 1986. ISSN 00165085. doi: S0016508586001257[pii].

[32] B W Watson, S J Meldrum, H C Riddle, R L Brown, and G E Sladen. pH profile of gut as measured by radiotelemetry capsule. *British medical journal*, 2(5805):104–106, 1972. ISSN 0959-8138. doi: 10.1136/bmj.2.5805.104.

[33] Carole A. Youngberg, Rosemary R. Berardi, William F. Howatt, Martha L. Hyneck, Gordon L. Amidon, James H. Meyer, and Jennifer B. Dressman. Comparison of gastrointestinal pH in cystic fibrosis and healthy subjects. *Digestive Diseases and Sciences*, 32(5):472–480, 1987. ISSN 01632116. doi: 10.1007/BF01296029.

[34] I M Samloff. Pepsins, peptic activity, and peptic inhibitors. *Journal of clinical gastroenterology*, 3(Suppl 2):91–94, 1981. ISSN 01920790.

[35] I M Samloff. Peptic ulcer: the many proteinases of aggression. *Gastroenterology*, 96(2 Pt 2 Suppl):586–595, 1989. ISSN 0016-5085.

[36] M. L. Schubert and G. M. Makhlouf. Neural, hormonal, and paracrine regulation of gastrin and acid secretion. In *Yale Journal of Biology and Medicine*, volume 65, pages 553–560, 1992.

[37] Deanna M. Mudie, Gordon L. Amidon, and Gregory E. Amidon. Physiological parameters for oral delivery and in vitro testing, 2010. ISSN 15438384.

[38] Maria Vertzoni, Jennifer Dressman, James Butler, John Hempenstall, and Christos Reppas. Simulation of fasting gastric conditions and its importance for the in vivo dissolution of lipophilic compounds. *European Journal of Pharmaceutics and Biopharmaceutics*, 60(3):413–417, 2005. ISSN 09396411. doi: 10.1016/j.ejpb.2005.03.002.

[39] F K Winkler, A D'Arcy, and W Hunziker. Structure of human pancreatic lipase. *Nature*, 343(6260):771–774, 1990. ISSN 0028-0836. doi: 10.1038/343771a0.

[40] J. M. DeSesso and C. F. Jacobson. Anatomical and physiological parameters affecting gastrointestinal absorption in humans and rats, 2001. ISSN 02786915.

[41] Dileep N. Lobo, Paul O. Hendry, Gabriel Rodrigues, Luca Marciani, John J. Totman, Jeff W. Wright, Tom Preston, Penny Gowland, Robin C. Spiller, and Kenneth C H Fearon. Gastric emptying of three liquid oral preoperative metabolic preconditioning regimens measured by magnetic resonance imaging in healthy adult volunteers: A randomised double-blind, crossover study. *Clinical Nutrition*, 28(6):636–641, 2009. ISSN 02615614. doi: 10.1016/j.clnu.2009.05.002.

[42] Andreas Steingoetter, Mark Fox, Reto Treier, Dominik Weishaupt, Borut Marincek, Peter Boesiger, Michael Fried, and Werner Schwizer. Effects of posture on the physiology of gastric emptying: a magnetic resonance imaging study. *Scandinavian journal of gastroenterology*, 41(10):1155–1164, 2006. ISSN 0036-5521. doi: 10.1080/00365520600610451.

[43] Deanna Mudie, Kathryn Murray, Caroline L Hoad, Susan E Pritchard, Martin Garnett, Gordon L Amidon, Penny A Gowland, Robin C Spiller, Gregory E Amidon, and Luca Marciani. Quantification of gastrointestinal liquid volumes and distribution following a 240 mL dose of water in the fasted state. *Mol Pharm*, 11(9):3039–47, September 2014. ISSN 15438392. doi: 10.1021/mp500210c. URL `http://www.ncbi.nlm.nih.gov/pubmed/25115349`.

[44] Luca Marciani, Eleanor F. Cox, Caroline L. Hoad, Susan Pritchard, John J. Totman, Steve Foley, Amisha Mistry, Steven Evans, Penny A. Gowland, and Robin C. Spiller. Postprandial Changes in Small Bowel Water Content in Healthy Subjects and Patients With Irritable Bowel Syndrome. *Gastroenterology*, 138(2), 2010. ISSN 00165085. doi: 10.1053/j.gastro.2009.10.055.

[45] C Schiller, C-P Fröhlich, T Giessmann, W Siegmund, H Mönnikes, N Hosten, and W Weitschies. Intestinal fluid volumes and transit of dosage forms as assessed by magnetic resonance imaging. *Alimentary pharmacology & therapeutics*, 22(10):971–9, November 2005. ISSN 0269-2813. doi: 10.1111/j.1365-2036.2005.02683.x. URL `http://www.ncbi.nlm.nih.gov/pubmed/16268972`.

[46] L. F. Prescott. Gastric emptying and drug absorption. *British Journal of Clinical Pharmacology*, 1(3):189–190, 1974.

[47] D Liao, H Gregersen, T Hausken, O H Gilja, M Mundt, and G Kassab. Analysis of surface geometry of the human stomach using real-time 3-D ultrasonography in vivo. *Neurogastroenterology and motility : the official journal of the European Gastrointestinal Motility Society*, 16(3):315–324, 2004. ISSN 1350-1925. doi: 10.1111/j.1365-2982.2004.00522.x.

[48] H. Frederic Martini, Judi L. Nath, and Edwin F. Bartholomew. *Fundamentals of Anatomy & Physiology*. Benjamin Cummings, San Francisco, 7th edition, 2006.

[49] Margaret E. Smith and Dion G. Morton. *The Digestive System*. Churchill Livingstone, Edinburgh, 2001.

[50] Kiyohiko Sugano. *Biopharmaceutics Modeling and Simulations: Theory, Practice, Methods, and Applications*. John Wiley & Sons Ltd, Hoboken, 2012.

[51] Thorsteinn Loftsson and Marcus E. Brewster. Physicochemical properties of water and its effect on drug delivery. A commentary. *International Journal of Pharmaceutics*, 354(1-2):248–254, 2008. ISSN 03785173. doi: 10.1016/j.ijpharm.2007.08.049.

[52] Hans Lennernäs. Human intestinal permeability, 1998. ISSN 00223549.

[53] S S Davis, J G Hardy, and J W Fara. Transit of pharmaceutical dosage forms through the small intestine. *Gut*, 27(8):886–92, August 1986. ISSN 0017-5749. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1433369&tool=pmcentrez&rendertype=abstract`.

[54] Hans Gregersen. *Biomechanics of the gastrointestinal tract: new perspectives in motility research and diagnostics*. Springer-Verlag, London, 2003. ISBN 1852335203. doi: 10.1007/978-1-4471-3742-9.

[55] Jianzhuo Wang and Douglas R. Flanagan. General solution for diffusion-controlled dissolution of spherical particles. 1. Theory. *Journal of Pharmaceutical Sciences*, 88(7):731–738, 1999. ISSN 00223549. doi: 10.1021/js980236p.

[56] R Hintz and K Johnson. The effect of particle size distribution on dissolution rate and oral absorption, 1989. ISSN 03785173.

[57] Aristides Dokoumetzidis and Panos Macheras. A century of dissolution research: From Noyes and Whitney to the Biopharmaceutics Classification System, 2006. ISSN 03785173.

[58] H. Potthast, J. B. Dressman, H. E. Junginger, K. K. Midha, H. Oeser, V. P. Shah, H. Vogelpoel, and D. M. Barends. Biowaiver monographs for immediate release solid oral dosage forms: Ibuprofen, 2005. ISSN 00223549.

[59] Karl A. Levis, Majella E. Lane, and Owen I. Corrigan. Effect of buffer media composition on the solubility and effective permeability coefficient of ibuprofen. *International Journal of Pharmaceutics*, 253(1-2):49–59, 2003. ISSN 03785173. doi: 10.1016/S0378-5173(02)00645-2.

[60] Lance R Shaw, William J Irwin, Tim J Grattan, and Barbara R Conway. The effect of selected water-soluble excipients on the dissolution of paracetamol and Ibuprofen. *Drug development and industrial pharmacy*, 31(6):515–525, 2005. ISSN 0363-9045. doi: 10.1080/03639040500215784.

[61] Shobha N. Bhattachar, Laura A. Deschenes, and James A. Wesley. Solubility: it's not just for physical chemists, 2006. ISSN 13596446.

[62] Neal M. Davies. Clinical pharmacokinetics of ibuprofen. The first 30 years. *Clinical Pharmacokinetics*, 34(2):101–154, 1998. ISSN 03125963. doi: 10.2165/00003088-199834020-00002.

[63] M Kristensen. Titration curves for gastric secretion. A study on duodenal ulcer and gastric ulcer with particular reference to the effect of glycopyrronium. *Scandinavian journal of gastroenterology. Supplement*, 32:11–144, 1975. ISSN 00855928.

[64] W D Rees, D Botham, and L A Turnberg. A demonstration of bicarbonate production by the normal human stomach in vivo. *Digestive diseases and sciences*, 27(11):961–966, 1982. ISSN 01632116.

[65] Gladys R Bucher, Joseph C Flynn, and C S Robinson. The Action of hte Human Small Intestine in Altering the Composition of hte Physiological Saline. *Journal of Biological Chemistry*, 155(1):305–313, 1944. URL `http://www.jbc.org/content/155/1/305.short`.

[66] Joel Griffith Hardman, Alfred Goodman Gilman, Lee Limbird, and Theodore W. Rall, editors. *Goodman and Gilman's The Pharmacological Basis of Therapeutics, Twelfth Edition*. McGraw-Hill, New York, 10th edition, 2001.

[67] M Repishti, D L Hogan, V Pratha, L Davydova, M Donowitz, C M Tse, and J I Isenberg. Human duodenal mucosal brush border Na(+)/H(+) exchangers NHE2 and NHE3 alter net bicarbonate movement. *American journal of physiology. Gastrointestinal and liver physiology*, 281(1):G159–G163, 2001. ISSN 0193-1857.

[68] Horace W. Davenport, editor. *Physiology of the Digestive Tract*. Year Book Medical, Chicago, 1982.

[69] Lemuel C McGee and A Baird Hastings. The Carbon Dioxide Tension and Acid-Base Balance of Jejunal Secretions in Man. *Journal of Biological Chemistry*, 142(2):893–904, February 1942. URL `http://www.jbc.org/content/142/2/893.short`.

[70] S J Rune and F W Henriksen. Carbon dioxide tensions in the proximal part of the canine gastrointestinal tract. *Gastroenterology*, 56(4):758–762, 1969. ISSN 00165085.

[71] J. G. Banwell, S. L. Gorbach, N. F. Pierce, R. Mitra, and A. Mondal. Acute undifferentiated human diarrhea in the tropics. II. Alterations in intestinal fluid and electrolyte movements. *Journal of Clinical Investigation*, 50(4):890–900, 1971. ISSN 00219738. doi: 10.1172/JCI106561.

[72] Leonard R. Johnson. *Gastrointestinal Physiology*. Mosby, St. Louis, 6th edition, 2001.

[73] J.B. West, editor. *Best & Taylor's Physiological Basis of Medical Practice*. Williams & Wilkins, Baltimore, 1985.

[74] Abraham White, Philip Handler, and Emil L. Smith. *Principles of Biochemistry*. McGraw-Hill, New York, 4th edition, 1968.

[75] M Efentakis and J B Dressman. Gastric juice as a dissolution medium: surface tension and pH. *European journal of drug metabolism and pharmacokinetics*, 23(2):97–102, 1998. ISSN 0378-7966. doi: 10.1007/BF03189322.

[76] J Rhodes, D E Barnardo, S F Phillips, R A Rovelstad, and A F Hofmann. Increased reflux of bile into the stomach in patients with gastric ulcer. *Gastroenterology*, 57(3):241–252, 1969. ISSN 00165085.

[77] Maria Vertzoni, Helen Archontaki, and Christos Reppas. Determination of intralumenal individual bile acids by HPLC with charged aerosol detection. *Journal of lipid research*, 49(12):2690–2695, 2008. ISSN 0022-2275. doi: 10.1194/jlr.D800039-JLR200.

[78] M Armand, P Borel, B Pasquier, C Dubois, M Senft, M Andre, J Peyrot, J Salducci, and D Lairon. Physicochemical characteristics of emulsions during fat digestion in human stomach and duodenum. *The American journal of physiology*, 271(1 Pt 1):G172–G183, 1996. ISSN 0002-9513.

[79] Joachim Brouwers, Jan Tack, Frank Lammert, and Patrick Augustijns. Intraluminal drug and formulation behavior and integration in in vitro permeability estimation: A case study with amprenavir. *Journal of Pharmaceutical Sciences*, 95(2):372–383, 2006. ISSN 00223549. doi: 10.1002/jps.20553.

[80] T C Northfield and I McColl. Postprandial concentrations of free and conjugated bile acids down the length of the normal human small intestine. *Gut*, 14(7):513–518, 1973. ISSN 0017-5749. doi: 10.1136/gut.14.7.513.

[81] R Lambert, F Martin, and M Vagne. Relationship between hydrogen ion and pepsin concentration in human gastric secretion. *Digestion*, 1(2):65–77, 1968. ISSN 0012-2823. doi: 10.1159/000196835.

[82] H A Schmidt, G Fritzlar, W Dölle, and H Goebell. Comparative studies on the histamine and insulin stimulated acid pepsin secretion in patients suffering from ulcus duodeni and control persons. *Deutsche medizinische Wochenschrift (1946)*, 95(40):2011–2016, 1970.

[83] H. M. Fadda, T. Sousa, A. S. Carlsson, B. Abrahamsson, J. G. Williams, D. Kumar, and A. W. Basit. Drug solubility in luminal fluids from different regions of the small and large intestine of humans. *Molecular Pharmaceutics*, 7(5): 1527–1532, 2010. ISSN 15438384. doi: 10.1021/mp100198q.

[84] C V Gisolfi, R W Summers, G P Lambert, and T Xia. Effect of beverage osmolality on intestinal fluid absorption during exercise. Technical Report 5, 1998.

[85] B. L. Pedersen, A. Mullertz, H. Brondsted, and H. G. Kristensen. A comparison of the solubility of danazol in human and simulated gastroiatestinal fluids. *Pharmaceutical Research*, 17(7):891–894, 2000. ISSN 07248741. doi: 10.1023/A:1007576713216.

[86] J D Maxwell, A Ferguson, and W C Watson. The effect of gastric secretory status on jejunal pH measured by radiotelemetering. *Digestion*, 4(6):345–352, 1971. ISSN 0012-2823. doi: 10.1159/000197139.

[87] Mario Grassi, Gabriele Grassi, Romano Lapasin, and Italo Colombo. *Understanding Drug Release and Absorption Mechanisms: A Physical and Mathematical Approach.* CRC Press, Boca Raton, 2006.

[88] W. S. Snyder, M. J. Cook, E. S. Nasset, L. R. Karhausen, G. P. Howells, and I. H. Tipton. Report of the Task Group on Reference Man : a report. Technical report, International Commission on Radiological Protection, New York, 1975.

CHAPTER III

Methodology for Statistical Analysis of Antroduodenal Manometry Signals

> "On no subject in physiology do we meet with so many discrepancies of fact and opinion as in that of the physiology of the intestinal movements. Among factors contributing to such divergences must doubtless be included the varying behavior of the gut in different animals, the varying conditions of the animal with regard to feeding or conditions of experiment, such as exposure and cooling of the intestines." [1]

3.1 Introduction

The migrating motor or myoelectric complex (MMC) occurs in most mammals to ensure proper mixing, transport, digestion and absorption of luminal content, as well as facilitate blood flow and clear accumulated secretory fluids and overpopulating bacteria[2,3]. Typically, a cycle of the MMC–seemingly controlled by enteric neural mechanisms that are modulated by the central nervous system and circulating endogenous substances–includes a quiescent state (phase I) followed by progressively increasing contractional frequency (phase II) and finally the most active state of high amplitude contractions (phase III)[3]. During the fasted state, the MMC is characterized by cyclical transitions between three states: phase I, lasting 20-90 minutes, is a period of little activity where few contractions over 10 mmHg are seen; phase II is characterized by an increase in frequency for a period of 35-135 minutes during which irregular contractions over 10 mmHg occur but below a frequency of 10 contractions per minute; and phase III, or the active phase, shows regular contractions ($\geq$ 10 per minute) for 2 to 15 minutes[4–8] and the coefficient of variation of contraction forces is much greater in this phase[9].

Through quantitative analysis of radiological and manometric studies, Vantrappen et al. demonstrated the similarities of the MMC in canines and humans[10]. Using fasted dogs, Sarna et al.[12] sought to correlate contractile activity with plasma levels of motilin–thought to be one of the most important factors in controlling the MMC [13]. Morphine is known to act on opioid receptors and initiate phase III activities[11], and they found that both spontaneous (i.e. natural) and morphine-induced phase III patterns correlated with high levels of motilin in the plasma which subsequently declined during phase I. Furthermore, it was shown that, in fasted dogs, bands of action potential activity migrated caudally with new bands starting in the stomach or duodenum as earlier ones reached the ileum[14–17]. Indeed, the propagation of the MMC seemed to decline aborally, with the steepest decrease occurring in the distal jejunum, while contraction frequencies of phase III remained constant in the upper small bowel[18].

It is therefore unsurprising that characterization of normal gastrointestinal (GI) functions of many regions of the gut is still an ongoing endeavor that has led to advances in many techniques. GI motility in particular has been the subject of intense and growing research for some time. Gastric contractions have been evaluated by Fourier analysis of condensed images of gastric scintigraphy studies[19]. Other technologies include: high resolution manometry to study motor patterns with detailed spatiotemporal mapping; magnetic resonance imaging (MRI) and ultrasound measuring dynamic flow and morphological change without the need for radiation; wireless capsules that transit along the entire gut yielding concurrent measures of contractions and transit times along with other physiological data such as changes in pH; and slow wave mapping and magnetometry reporting GI electrical activity [20–28]. Veritably, a recent study regarding colonic motility described the necessity for high resolution manometry by demonstrating the high incidence of misinterpretation of both frequency and polarity of propagating sequences when sensors were too far apart[29]. Further complexification arises from the large variability in site of origin, cycle length, and migration velocity of the MMC in humans, both healthy and diseased[8]. Nevertheless, there remains a need for an objective and automated means of classifying motility signals and determining the statistics of phase durations, cycle lengths, and MMC wave propagation.

3.2 Methods

3.2.1 Data acquisition

As of this writing, a current study funded by the FDA is under way to analyze manometric data based on triplicate sampling in the stomach, duodenum, jejunum, and, if possible, the ileum (Figure 3.1). The three manometry ports are spaced 5cm apart at each site, sufficient for proper identification of MMC polarity[29]. For the development of a statistical-based classification scheme here, Medical Measurement Systems (MMS) provided multi-channel antroduodenal manometry data sampled at a frequency of 10Hz for approximately 250 minutes (Figure 3.2).

It is reasonable to assume, given the periodic nature of MMCs, that a first approach would involve a Fourier transform analysis of the signal to detect cycle frequency. Due to noise and lack of absolute regularity, however, this proves a much more complicated task. Figure 3.3 illustrates the signal, a histogram of pressure occurrences heavily skewed toward low baseline values, and a Fourier transformation of the signal wherein no apparent frequency stands out. Previous classification attempts included the wavelet transformation of signal data followed by a principal component analysis to determine if wavelet coefficients sufficed in differentiating regions of activity from inactivity; and the optimization a Poisson process to best represent the stochastic evolution of signal data. Neither of these methods proved able classifiers due to the underlying noise as well as the pseudo-deterministic cycle (i.e. the necessarily ordered transition from phase I to II to III back to I). Herein, two orthogonal methods are discussed which complement each other's results when applied to the test data set.

3.2.2 Wavelet transformations

Manometry signals are generally characterized by sharp peaks entrenched in noisy data, thus requiring a careful approach when determining what constitutes pressure activity of interest. Due to the semi-periodic nature of GI motility, Fourier transforms cannot be relied upon to extract information on cyclicality of phase transitions. Rather, the analogous continuous wavelet transform (CWT) approach is more robust due to the time-frequency representation of a signal offering frequency localization. CWT-based peak detection methods have been used to characterize various types of spectra with low signal-to-noise ratios including x-ray diffraction, mass spectrometry, atmospheric patterns of convection, and complex geological processes[30–33]. A CWT is a function $\psi \in L^2(\mathbb{R})$ such that $\int_{\mathbb{R}} \psi(t)dt = 0$, normalized so $||\psi||_2 = 1$. This function is called the *mother wavelet*, from which a family of time-scale waveforms can be obtained for scale $a > 0$ and translational value b (Equation 3.1). Given a time-dependent signal $f(t) \in L^2(\mathbb{R})$, the CWT is defined as its projection on that the wavelet basis (Equation 3.2).

$$\psi_{a,b}(t) = \frac{1}{|a|^{1/2}} \psi\Big(\frac{t-b}{a}\Big) \quad a, b \in \mathbb{R} \tag{3.1}$$

$$F_w(a,b) = \int_{\mathbb{R}} f(t)\psi_{a,b}(t)dt \tag{3.2}$$

The signal is first de-trended (using *scipy.signal.detrend*) and thresholded such that pressures below 50 mmHg are discounted. Du et al. developed a peak detection method based on the CWT of a one-dimensional signal. The algorithm is as follows: a series of widths are chosen corresponding to the expected widths of the signal peaks; the signal is convolved with a wavelet of each defined width; the maxima that are maintained across all widths–forming "ridges" across each row that corresponds to the convolution with that particular width–are thus defined as peaks; there can be allowances for the maximum distance between ridge connections (in this case, a temporal shift in either direction); a gap threshold defining potential discontinuities across rows, the minimum length a ridge must be; the noise floor which is the percentile of data examined below which to consider noise; and the signal-to-noise ratio (the signal is the value of the CWT matrix at the shortest length scale). Abramovich et al., Antoniadis, and Silverman provide comprehensive reviews of the statistical applications of wavelet analysis[34–36].

3.2.3 Kernel density estimators

GI manometry can be treated as a spike train sampled from a stochastic process. Shimazaki and Shinomoto developed a kernel smoother for estimating the instantaneous rate of spike occurrences[37]. The array of time points associated with the CWT-detected peaks, x_t, is convolved with the kernel, $k(s)$, to obtain the kernel density estimate (KDE) such that $\int k(s)ds = 1$ (Equation 3.3). The most frequently-used kernel is the Gaussian density function (Equation 3.4), where $w^2 = \int s^2 k(s)ds < \infty$ is a finite bandwidth.

$$\hat{\lambda}_t = \int_{-\infty}^{\infty} x_{t-s}\, k(s)ds \tag{3.3}$$

$$k_w(s) \;=\; \frac{1}{\sqrt{2\pi}w} exp\Big(-\frac{s^2}{2w^2}\Big) \tag{3.4}$$

Assuming the spike train to be an inhomogeneous Poisson point process, the estimate $\hat{\lambda}_t$ is optimized to best reflect the unknown underlying rate λ_t using the mean integrated square error (MISE) as a goodness-of-fit of the estimate (Equation 3.5) where E is the expectation with respect to rate λ_t.

$$MISE = \int_a^b E(\hat{\lambda}_t - \lambda_t)^2 dt \tag{3.5}$$

Shimazaki and Shinomoto further expand on determination of a fixed bandwidth that best evinces the underlying rate λ_t (equations not reproduced here). Briefly, for an interval of interest $[a, b]$, the cost function $C_n(w)$ of a kernel k_w is expressed as a function of the associated bandwidth w (Equation 3.6). Thus there is a bandwidth w^* that minimizes the cost function $C_n(w)$. Assuming the underlying data of size n being estimated is Gaussian with standard deviation $\hat{\sigma}$, an approximation for the optimal bandwidth w is given by *Silverman's rule of thumb*[38] (Equation 3.7).

$$C_n(w) = \frac{1}{n^2} \sum_{i,j} \int_a^b k_w(t-t_i)\; k_w(t-t_j)dt \;-\; \frac{2}{n^2} \sum_{i \neq j} k_w(t_i - t_j) \tag{3.6}$$

$$w = \Big(\frac{4\hat{\sigma}^5}{3n}\Big)^{\frac{1}{5}} \approx 1.06\hat{\sigma}n^{-1/5} \tag{3.7}$$

3.2.4 Gaussian processes

As functions are continuous over time, there are an infinite number of points over which they are defined. For a given range of time, an uncountably infinite number of possible functions that may exist. Therefore there exists a seemingly impossible task of computing such possibilities in finite time. However, Gaussian probability distributions can be extended from random variables to random functions with associated values for particular inputs. Despite the apparent naïvety of such an approach, it has been employed with tremendous success in many different applications.

Since this is simply an extension of multivariate distributions, there thus exist mean functions and covariances. Observations, in this case the manometry data, are related via the covariance function $k(x, x')$ for two inputs x and x'. A popular choice is the squared exponential:

$$k(x, x') = \sigma_f \ exp\left[\frac{-(x - x')^2}{2l^2}\right] \tag{3.8}$$

The maximum covariance is defined as σ_f, when $x \approx x'$. Conversely, if the inputs are far apart, then $k(x, x') \approx 0$. The separation of x and x' is reflected in the parameter l. Thus, given n observations of the signal $\boldsymbol{y}$, the objective is to predict the value y_* rather than the actual function f_* whose expectations are equivalent but differ in terms variance according to observational noise (see Rasmussen and Williams for further discussion[39]).

Using this approach, an expected pressure and whether that pressure is above or below certain thresholds (75th or 25th percentile, respectively) can be determined for a given time point. A continuous region above the threshold lasting 5 minutes or longer is defined as phase III. If, on the other hand, there is a region of extensive low-pressure activity below the threshold with peaks spaced more than 10 minutes apart, this is phase I. Phase II is the intermittent regions where expected pressure oscillates between high and low but is not sustained for a sufficient period–the 5 minute time span corresponding to phase III.

3.3 Results and discussion

3.3.1 Peak detection

The method of CWT-based peak detection is successful in determining a time-variant peak density distribution in the test data. Two duodenal and one antral port are analyzed with considerably differing patterns to test the potential extreme scenarios (dense regions of pressure peaks and sparsely distributed ones with large intermittent regions of inactivity). The detected peaks and their corresponding histogram plots are presented in Figure 3.4. It is important to note the dependence of histogram presentation on the arbitrary bin width when plotting, an issue that motivates non-parametric smoothing of the data.

3.3.2 Kernel smoothing

Detected peaks from the duodenal channel (Figure 3.2) are plotted showing the evolution of spike activity over time (Figure 3.5). There is an apparent cluster of high activity centered around ∼75 minutes and ∼225 minutes. Treating each peak (dark red bar) as the mean of a Gaussian distribution results in N basis functions where N is the number of detected peaks (Figure 3.6).

Two important factors in KDE are the bandwidth parameter and the kernel function. The effect of using different bandwidths is illustrated in Figure 3.7. Applying Silverman's rule (Equation 3.7) based on assumed normality, the bandwidth is 7.17. An alternative to this assumption is evaluation of the bandwidth hyperparameter using cross-validation as follows:

- The data is split into k sets
- The KDE is evaluated for a range of bandwidths using $k-1$ sets as training data
- The resultant model is validated on the remaining part where a performance measure is computed to produce the maximum likelihood

Employing this empirical method, the optimum bandwidth is 8.64. The effects of using different bandwidths can lead to both over-smoothing and under-smoothing, however both the estimated Silverman bandwidth and the computed empirical bandwidth produce very similar results (Figure 3.7). KDE can be accomplished with a variety of kernel functions, the most common being the Gaussian kernel. The kernel function does not have as great an impact on the resultant density as the bandwidth estimation does. Several other common kernel functions include the quartic/bi-weight, cosine, Epanechnikov, triangular, and tri-weight (Figure 3.8). Wand and Jones provide extensive discussion of common kernel functions[40]. Summing the Gaussian basis functions used with the estimated bandwidth, regions of high peak density are detected (Figure 3.9). These are potentially regions of phase III activity, and low peak densities thus corresponds to phase I.

3.3.3 Regression

Applying a Gaussian process regression, the large amplitudinal differences between baseline and peak pressures are minimized and smoothed resulting in identifiable regions of high, moderate, and low expected pressure activity (Figure 3.10). Therefore regions of high activity can be clustered into contiguous timespans and so too can regions of low activity. Oscillations between high and low predicted ranges are thus periods of moderate activity.

3.3.3.1 Determination of motility states

Figure 3.11 illustrates the classification of the phases based on this approach. The phase durations of the sample data correspond to physiological ranges that have previously been reported[4], and this method will be used to analyze the manometry signals acquired from the volunteer subjects in the forthcoming FDA study. The annotated regions are shown with the detected peaks (plotted as notches along the x-axis). Indeed the high peak densities occur contemporaneously with the regions of phase III activity.

3.4 Conclusion

The use of CWT-based peak detection allows for a robust method of detecting true pressure-related activity engulfed in noisy data. This allows treatment of manometric data as a stochastic process, using a kernel density estimator which has the benefit of being smooth and independent of the endpoints unlike histograms, and it has a tunable bandwidth parameter that can be analytically or empirically solved to best represent the underlying stochastic rate. Employing GPs, the time-dependent pressure values can be estimated and used to identify the different regions of the signal based on thresholding. The accordance of the two approaches suggests the high likelihood of an accurate, unsupervised signal classification scheme.

Beyond distinguishing different phases, a pattern of interest is seen in the signals. Phase III peaks appear to be preceded by smaller spikes of activity not quite reaching the same amplitudes. However, the regions of phases I & II do not display such pre-spikes. Figure 3.12 illustrates examples from each phase. Higher temporal resolution in future studies will aide to substantiate the integrity of this pattern and what implications it may have for alternative classification method based on signal morphology.

3.5 Figures

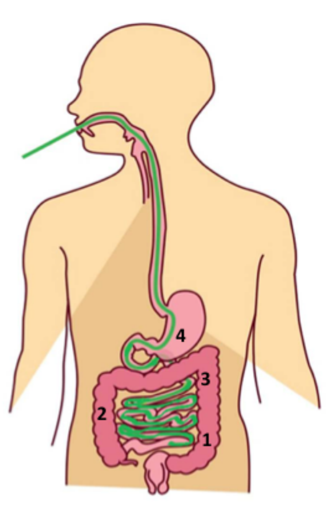


Figure 3.1: Schematic of tube placement with motility channels in the stomach (4), duodenum (3), proximal jejunum (2), and distal jejunum (1).

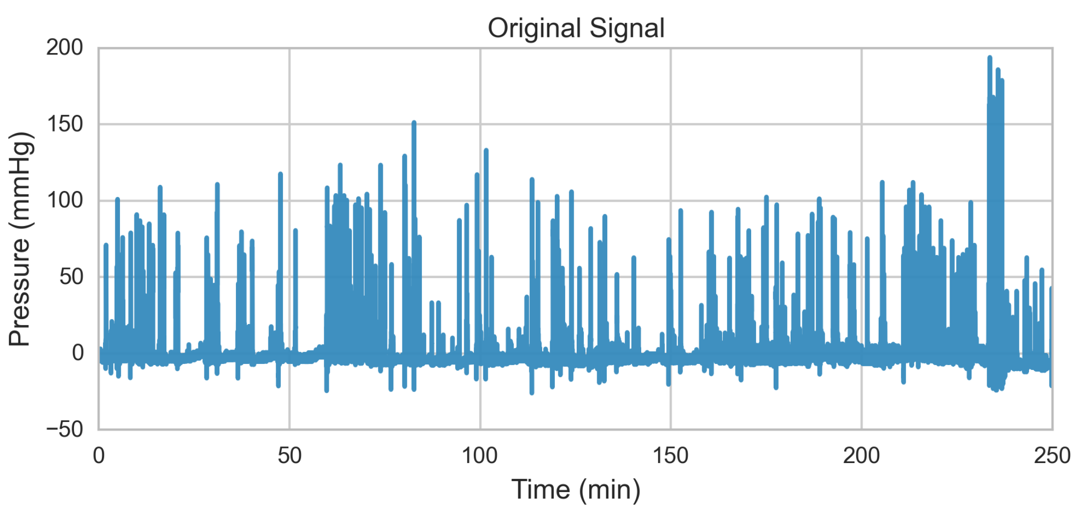


Figure 3.2: Manometry signal from a single duodenal channel.

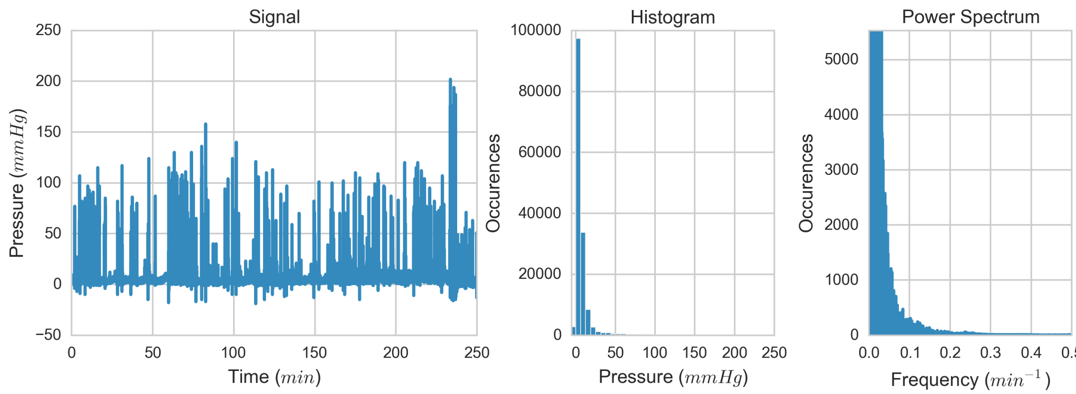


Figure 3.3: Left: original manometry tracing from Figure 3.2. Center: Histogram of pressure (mmHg) occurrences in the signal. Right: Fourier transform of the signal in Figure 3.2 showing lack of identifiable periodicity.

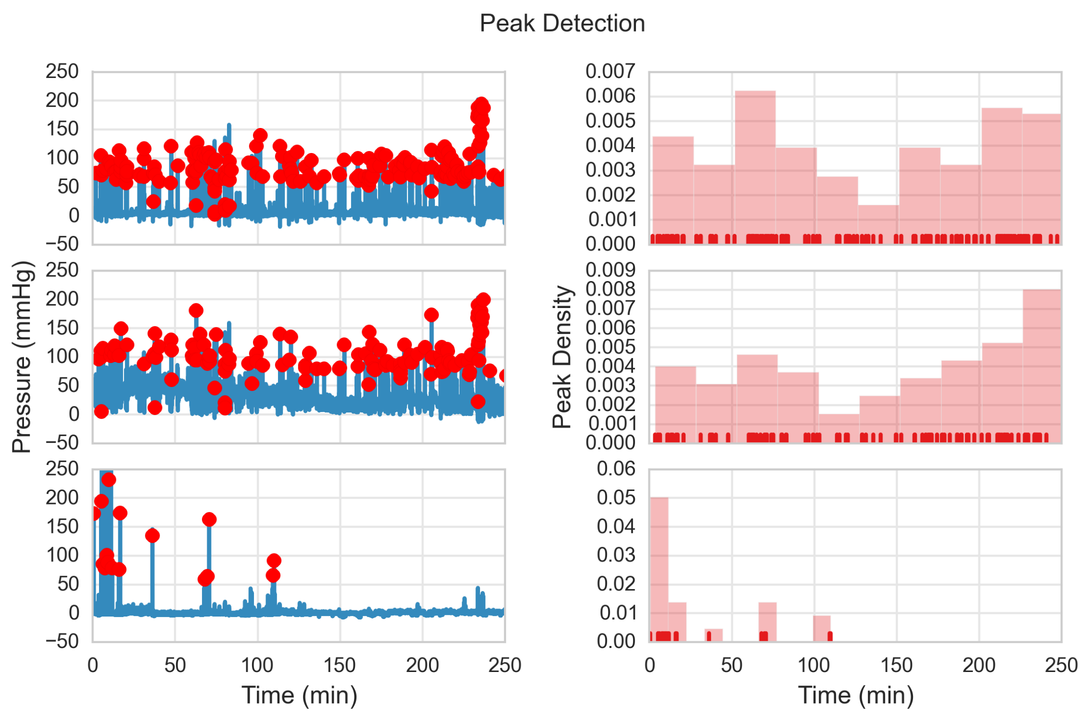


Figure 3.4: CWT-based peak detection results for three different antroduodenal manometry channels. The detected peaks are shown as red dots overplayed on top of the signals. Horizontally adjacent are histogram plots showing detected peak densities along the time axis.

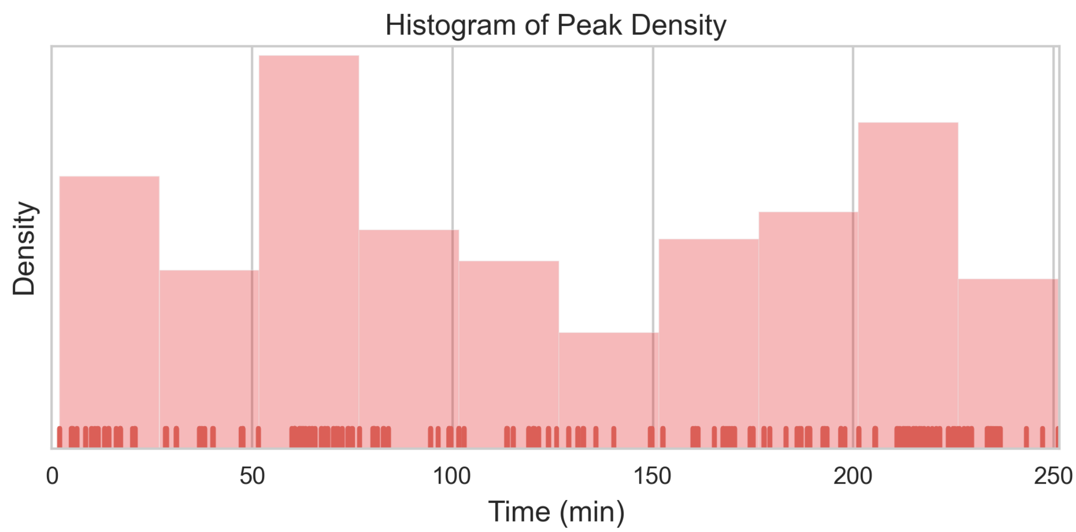


Figure 3.5: Temporal distribution of detected peaks (red ticks) represented as a histogram plot.

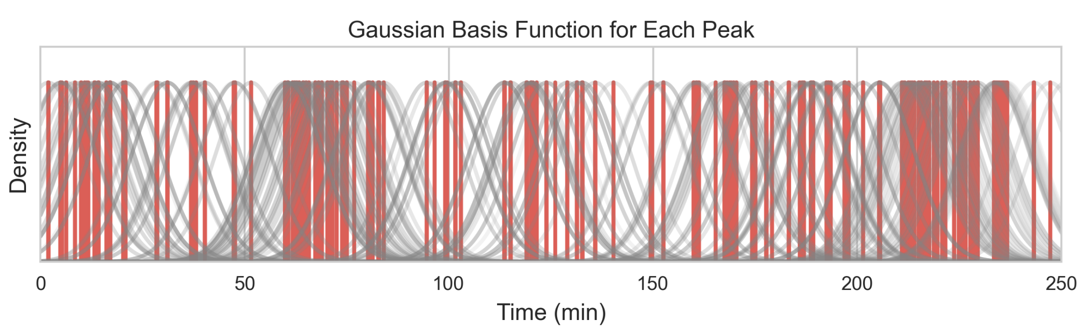


Figure 3.6: Each detected peak (red bar) is associated with a Gaussian basis function (gray curve).

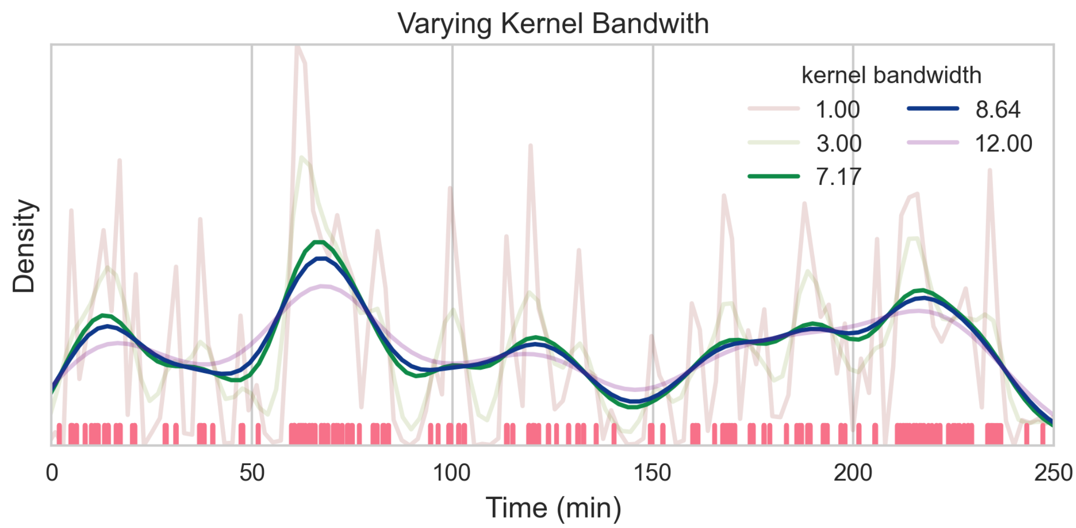


Figure 3.7: Kernel density estimations using different smoothing bandwidths.

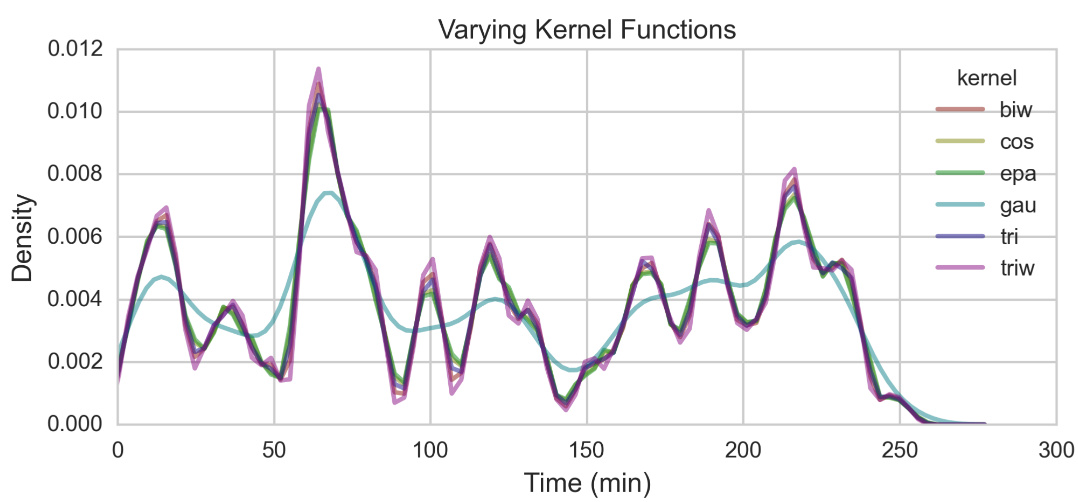


Figure 3.8: Using different kernel functions to estimate peak density: quartic/bi-weight, cosine, Epanechnikov, Gaussian, triangular, and tri-weight

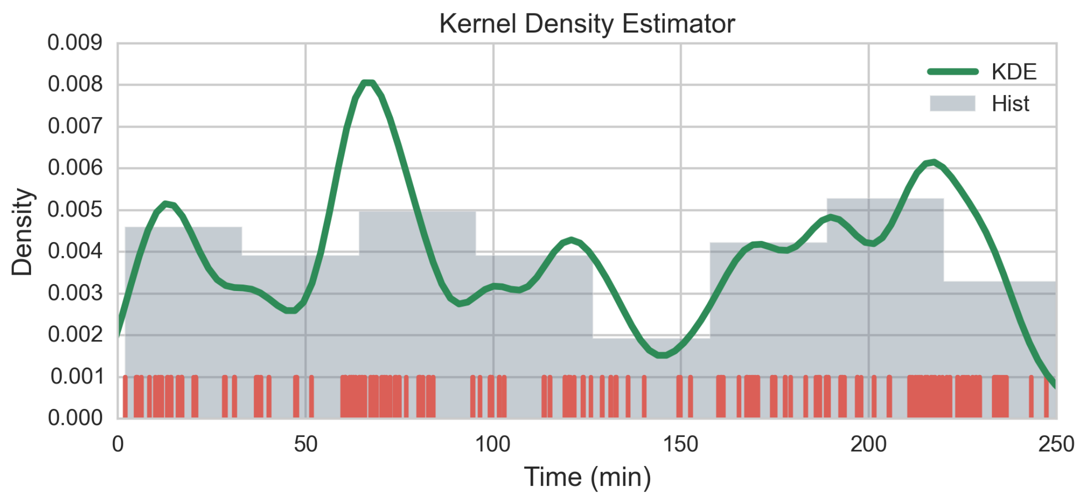


Figure 3.9: The distribution of detected peaks is estimated (green curve) by summing the basis functions (and normalizing so the functions integrates to 1 as a proper densities). The red tick marks are the time positions of peaks and the grey histogram shows their temporal distribution.

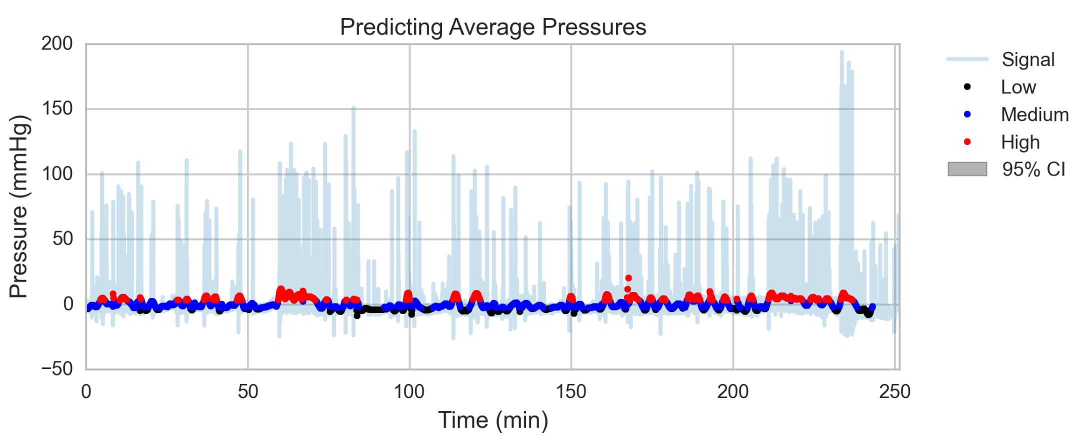


Figure 3.10: Prediction of average pressure function as determined by the Gaussian process regressions. Regions of low (black), medium (blue), and high (red) expected pressures are overplayed on the original signal (light blue).

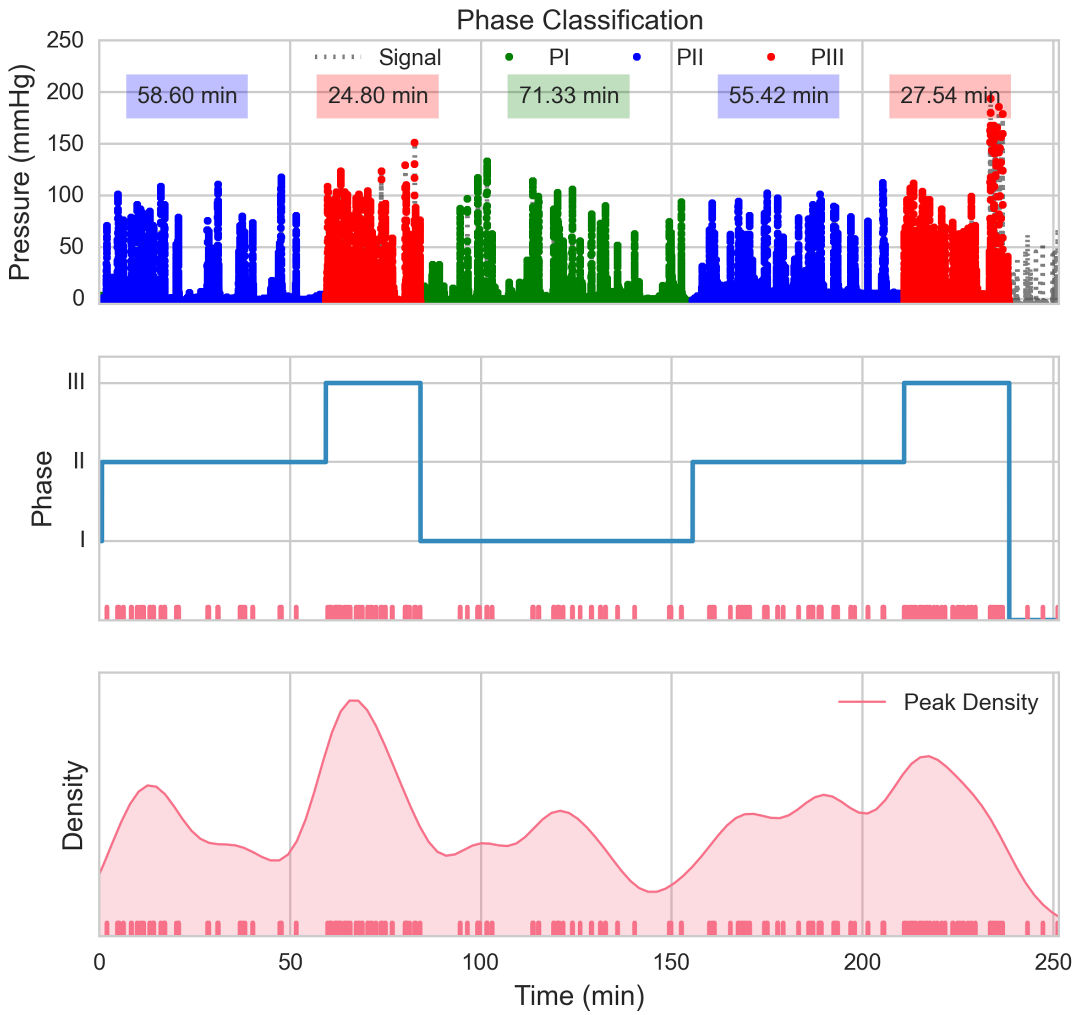


Figure 3.11: Top: overlay of the original signal with identified regions corresponding to the different phases (I in green, II in blue, and III in red). The durations are labeled above. Center: a transition plot, with the detected peaks plotted in red below. Bottom: the kernel density estimation (light red) of detected peak times (red tick marks).

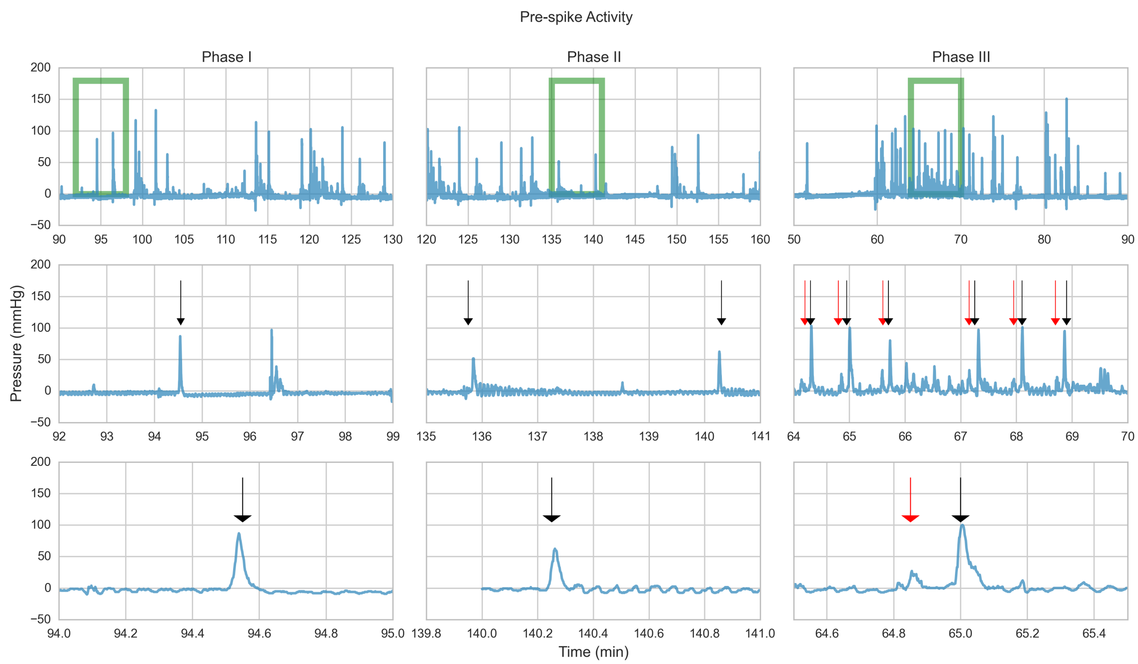


Figure 3.12: Phase III pressure peaks are preceded by short spikes that are not seen in either phases I or II. Top row contains 40-minute spans of signals in phases I-III. Middle row shows the boxed regions from top row in greater detail. The black arrows correspond to pressure peaks in the respective regions while the red arrows indicate the pre-spikes. Bottow row is further enlargement of a 1-minute period that shows pre-spikes occurring only in phase III.

3.6 References

[1] W M Bayliss and E H Starling. The movements and innervation of the small intestine. *The Journal of physiology*, 26(3-4):125–138, 1901. ISSN 0022-3751. doi: 10.1113/jphysiol.1899.sp000752.

[2] Krzysztof W. Romaski. Migrating motor complex in biological sciences: Characterization, animal models and disturbances, 2009. ISSN 00195189.

[3] S K Sarna. Cyclic motor activity; migrating motor complex: 1985. *Gastroenterology*, 89(4):894–913, 1985. ISSN 0016-5085. doi: S0016508585002888[pii].

[4] Peter Fleckenstein. Migrating electrical spike activity in the fasting human small intestine. *The American Journal of Digestive Diseases*, 23(9):769–775, September 1978. ISSN 0002-9211. doi: 10.1007/BF01079784. URL `http://link.springer.com/10.1007/BF01079784`.

[5] H Gregersen, S Rittig, L Vinter-Jensen, and K Kraglund. The relation between antral contractile activity and the duodenal component of the migrating motility complex. *Scandinavian journal of gastroenterology. Supplement*, 152: 36–41, 1988. ISSN 0036-5521. doi: 10.3109/00365528809095931.

[6] P Layer, T Schlesinger, G Gröger, and H Goebell. Modulation of human periodic interdigestive gastrointestinal motor and pancreatic function by the ileum. *Pancreas*, 8(4):426–432, 1993. ISSN 0885-3177. doi: 10.1097/00006676-199307000-00004.

[7] S. M. Scott, C. H. Knowles, D. Wang, E. Yazaki, L. Picon, D. L. Wingate, and G. Lindberg. The nocturnal jejunal migrating motor complex: Defining normal ranges by study of 51 healthy adult volunteers and meta-analysis. *Neurogastroenterology and Motility*, 18(10):927–935, 2006. ISSN 13501925. doi: 10.1111/j.1365-2982.2006.00824.x.

[8] Gordon L. Telford and Sushil K. Sarna. The migrating myoelectric complex of the small intestine. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 1(3):299–302, 1991. ISSN 10541500. doi: 10.1063/1.165843. URL `http://scholar.google.com/scholar?q=the+migrating+myoelectric+complex+of+the+small+intestine&btnG=&hl=en&as_sdt=0,15#`.

[9] M Schemann and H J Ehrlein. Mechanical characteristics of phase II and phase III of the interdigestive migrating motor complex in dogs. *Gastroenterology*, 91(1):117–123, 1986. ISSN 00165085.

[10] G. Vantrappen, J. Janssens, J. Hellemans, and Y. Ghoos. The interdigestive motor complex of normal subjects and patients with bacterial overgrowth of the small intestine. *Journal of Clinical Investigation*, 59(6):1158–1166, 1977. ISSN 00219738. doi: 10.1172/JCI108740.

[11] T D Lewis. Morphine and gastroduodenal motility. *Digestive diseases and sciences*, 44(11):2178–2186, 1999. ISSN 0163-2116.

[12] S Sarna, W Y Chey, R E Condon, W J Dodds, T Myers, and T M Chang. Cause-and-effect relationship between motilin and migrating myoelectric complexes. *The American journal of physiology*, 245(2):G277–G284, 1983. ISSN 00029513.

[13] Zen Itoh. Motilin and clinical application, 1997. ISSN 01969781.

[14] G M Carlson, B S Bedi, and C F Code. Mechanism of propagation of intestinal interdigestive myoelectric complex. *The American journal of physiology*, 222(4):1027–1030, 1972. ISSN 00029513.

[15] F. C. Code and J. A. Marlett. The interdigestive myo-electric complex of the stomach and small bowel of dogs. *Journal of Physiology*, 246(2):289–309, 1975.

[16] ML Grivel and Y Ruckebusch. The propagation of segmental contractions along the small intestine. *The Journal of physiology*, 227(2):611–25, December 1972. ISSN 0022-3751. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1331213&tool=pmcentrez&rendertype=abstracthttp://jp.physoc.org/content/227/2/611.short`.

[17] J H Szurszewski. A migrating electric complex of canine small intestine. *The American journal of physiology*, 217 (6):1757–1763, 1969. ISSN 0002-9513.

[18] M L Siegle, S Bühner, M Schemann, H R Schmid, and H J Ehrlein. Propagation velocities and frequencies of contractions along canine small intestine. *The American journal of physiology*, 258(5 Pt 1):G738–G744, 1990. ISSN 0002-9513.

[19] R. Linke, W. Muenzing, K. Hahn, and K. Tatsch. Evaluation of gastric motility by fourier analysis of condensed images. *European Journal of Nuclear Medicine*, 27(10):1531–1537, 2000. ISSN 03406997. doi: 10.1007/s002590000331.

[20] Amit C Ailiani, Thomas Neuberger, James G Brasseur, Gino Banco, Yanxing Wang, Nadine B Smith, and Andrew G Webb. Quantitative analysis of peristaltic and segmental motion in vivo in the rat small intestine using dynamic MRI. *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, 62(1):116–26, July 2009. ISSN 1522-2594. doi: 10.1002/mrm.21982. URL http://www.ncbi.nlm.nih.gov/pubmed/19353667.

[21] J. M. Andrews, D. G. O'Donovan, G. S. Hebbard, C. H. Malbert, S. M. Doran, and J. Dent. Human duodenal phase III migrating motor complex activity is predominantly antegrade, as revealed by high-resolution manometry and colour pressure plots. *Neurogastroenterology and Motility*, 14(4):331–338, 2002. ISSN 13501925. doi: 10.1046/j.1365-2982.2002.00337.x.

[22] J. Desipio, F. K. Friedenberg, A. Korimilli, J. E. Richter, H. P. Parkman, and R. S. Fisher. High-resolution solid-state manometry of the antropyloroduodenal region. *Neurogastroenterology and Motility*, 19(3):188–195, 2007. ISSN 13501925. doi: 10.1111/j.1365-2982.2006.00866.x.

[23] P. G. Dinning, J. W. Arkwright, H. Gregersen, G. O'Grady, and S. M. Scott. Technical advances in monitoring human motility patterns: Review article, 2010. ISSN 13501925.

[24] Andreas Gutzeit, Michael A. Patak, Constantin Von Weymarn, Nicole Graf, Aleksis Doert, Edwin Willemse, Christoph A. Binkert, and Johannes M. Froehlich. Feasibility of small bowel flow rate measurement with MRI. *Journal of Magnetic Resonance Imaging*, 32(2):345–351, 2010. ISSN 10531807. doi: 10.1002/jmri.22254.

[25] Luca Marciani, Paul Young, Jeff Wright, Rachel J. Moore, David F. Evans, Robin C. Spiller, and Penny A. Gowland. Echoplanar imaging in GI clinical practice: Assessment of gastric emptying and antral motility in four patients. *Journal of Magnetic Resonance Imaging*, 12(2):343–346, 2000. ISSN 10531807. doi: 10.1002/1522-2586(200008)12:2⟨343::AID-JMRI18⟩3.0.CO;2-M.

[26] Deanna Mudie, Kathryn Murray, Caroline L Hoad, Susan E Pritchard, Martin Garnett, Gordon L Amidon, Penny A Gowland, Robin C Spiller, Gregory E Amidon, and Luca Marciani. Quantification of gastrointestinal liquid volumes and distribution following a 240 mL dose of water in the fasted state. *Mol Pharm*, 11(9):3039–47, September 2014. ISSN 15438392. doi: 10.1021/mp500210c. URL http://www.ncbi.nlm.nih.gov/pubmed/25115349.

[27] Tanisa Patcharatrakul and Sutep Gonlachanvit. Technique of Functional and Motility Test: How to Perform Antroduodenal Manometry. *Journal of Neurogastroenterology and Motility*, 19(3):395–404, 2013. ISSN 2093-0879. doi: 10.5056/jnm.2013.19.3.395. URL http://www.jnmjournal.org/journal/view.html?doi=10.5056/jnm.2013.19.3.395.

[28] Satish S.C. Rao, Braden Kuo, Richard W. McCallum, William D. Chey, John K. DiBaise, William L. Hasler, Kenneth L. Koch, Jeffrey M. Lackner, Carrie Miller, Richard Saad, Jack R. Semler, Michael D. Sitrin, Gregory E. Wilding, and Henry P. Parkman. Investigation of Colonic and Whole-Gut Transit With Wireless Motility Capsule and Radiopaque Markers in Constipation. *Clinical Gastroenterology and Hepatology*, 7(5):537–544, May 2009. ISSN 15423565. doi: 10.1016/j.cgh.2009.01.017. URL http://linkinghub.elsevier.com/retrieve/pii/S1542356509000615.

[29] P. G. Dinning, L. Wiklendt, I. Gibbins, V. Patton, P. Bampton, D. Z. Lubowski, I. J. Cook, and J. W. Arkwright. Low-resolution colonic manometry leads to a gross misinterpretation of the frequency and polarity of propagating sequences: Initial results from fiber-optic high-resolution manometry studies. *Neurogastroenterology and Motility*, 25 (10), 2013. ISSN 13501925. doi: 10.1111/nmo.12170.

[30] Pan Du, Warren A. Kibbe, and Simon M. Lin. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics*, 22(17):2059–2065, 2006. ISSN 13674803. doi: 10.1093/bioinformatics/btl355.

[31] John M. Gregoire, Darren Dale, and R. Bruce Van Dover. A wavelet transform algorithm for peak detection and application to powder x-ray diffraction data. *Review of Scientific Instruments*, 82(1), 2011. ISSN 00346748. doi: 10.1063/1.3505103.

[32] C. Torrence and G.P. Compo. A Practical Guide to Wavelet Analysis. *Bulletin of the American Meteorological Society*, 79:61–78, 1998. ISSN 0003-0007. doi: 10.1.1.48.8367. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.8367.

[33] D. A. Yuen, G. Erlebacher, O. V. Vasilyev, D. E. Goldstein, and M. Fuentes. Role of wavelets in the physical and statistical modelling of complex geological processes. *Pure and Applied Geophysics*, 161(11-12):2231–2244, 2004. ISSN 00334553. doi: 10.1007/s00024-004-2560-z.

[34] Felix Abramovich, Trevor C Bailey, and Theofanis Sapatinas. Wavelet Analysis and Its Statistical Applications. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 49(1):1–29, 2000. ISSN 00390526. doi: 10.1111/1467-9884.00216. URL http://www.jstor.org/stable/2681252.

[35] Anestis Antoniadis. Wavelet methods in statistics: Some recent developments and their applications. *Statistics Surveys*, 1:16–55, 2007. ISSN 1935-7516. doi: 10.1214/07-SS014. URL `http://arxiv.org/abs/0712.0283`.

[36] B. W. Silverman. Wavelets in statistics: beyond the standard assumptions. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 357(1760):2459–2473, 1999. ISSN 1364-503X. doi: 10.1098/rsta.1999.0442.

[37] Hideaki Shimazaki and Shigeru Shinomoto. Kernel bandwidth optimization in spike rate estimation. *Journal of Computational Neuroscience*, 29(1-2):171–182, 2010. ISSN 09295313. doi: 10.1007/s10827-009-0180-4.

[38] B.W. Silverman and B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, London, 1986. ISBN 0-412-24620-1. doi: 10.2307/2347507.

[39] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, 2006. ISBN ISBN 026218253X. URL `http://www.gaussianprocess.org/gpml`.

[40] M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman & Hall, London, 1995.

CHAPTER IV

Conclusions

4.1 Summary

Motility of the gastrointestinal (GI) tract is without doubt a great source of variability for oral drug products both within individuals and across populations, yet its tremendous complexity has thus far encumbered many attempts at mechanistic analysis. Here, a continuous, time-dependent function is developed to model cyclical motility of the human GI tract during the fasted state and elucidate the effects of physiological variation on plasma levels. Results show that fast absorption and short plasma elimination half-lives profoundly affect bioequivalence testing. Plasma level variation is further compounded by a volumetric effect, calling into question patient compliance during self-administration. For oral drug products whose absorption profiles vary locationally, longer plasma elimination half-lives are still subject to great variation in systemic availability. The non log-normality of resultant C_{max} simulations also suggests the need for re-evaluating current metrics and the potential application of non-parametric methods in determining bioequivalence.

Predicting drug dissolution and transit along the GI tract relies on the ability to account for dynamic physiological conditions: locational pH changes, small fluid volumes, and the cyclical motility that drives contents, among others. A transit model is developed, using as examples the non-absorbable phenol red marker of net fluid change and ibuprofen, a weakly-acidic BCS Class II compound. Indeed, results suggest dissolution is not immediately completed and still occurs even in the mid-jejunum. Assuming non-sink conditions (i.e. measurable bulk concentrations of dissolving drug) leads to accurate predictions of reported C_{max}, T_{max}, and AUC. Validation and future refinement of this model will allow for successful *in vivo* predictions of drug product performances based on physiochemical properties.

Future refinement also relies on estimating the statistics pertaining to motility at the individual and population levels. It is then essential to have an objective quantification of motility, and treating manometric data as a stochastic process invites many applications of statistical computing to extract cycle and phase information. Kernel density estimation enables a non-parametric method of smoothing peak density along the time axis. A tunable bandwidth parameter is optimized thereby best representing the stochastic nature of peak occurrences and identifying regions of high-frequency pressure peaks. Orthogonal to kernel density estimation, a Gaussian process is used as a robust regression method based on thresholding. Time-dependent expected pressure values and their associated confidence intervals allow identification via relative differences in regions of the signal being analyzed. The accordance of the two methods therefore validates this unsupervised classification scheme. This can be extended to a multi-channel analysis where it becomes a two-dimensional problem, correlating signal patterns across space (location in the GI tract).

4.2 Future considerations

4.2.1 Mass balance analysis

A more robust model for diffusion-controlled dissolution can be employed by incorporating factors such as hydrodynamic influences, degree of confinement affecting bulk concentration, dissolving particle geometries, and [de]aggregation of particles due to inter-particle forces and shear stress[1]. The dissolution mass balance analysis presented in this dissertation has dealt with immediate release oral products. However, there is the potential to incorporate modified release and degradation. Indeed, the model can account for first-order degradation kinetics. Bulk degradation of polymers and release kinetics of modified formulations–taking into account erosion, drug dissolution, pore percolation, and matching multi-phasic release profiles–are certainly within reach[2,3], and this should be further explored for developing a more general and widely-applicable model of oral product performance along the GI tract.

4.2.2 Motility signal processing

Since a sub-pattern is seen in the signals during Phase III where small pressure spikes precede the larger peaks, alternative classification methods may be used in concert with those presented here to infer phase transitions and durations. Indeed, since the phase state is not directly observable but the dependent outputs (pressure signals) are, each state can be defined with a probability distribution over the output. The signal evolution then reveals information about the states. Since the sequence of the phases measured cannot be known directly (assuming sequential transitions–phase I to II to III back to I–but with random initial phase at the time of dosing), a Hidden Markov model can be employed to recover the initial state and thus the full sequence transition. A similar approach has previously been used, simultaneously recording intraluminal pressure and gut diameter in the isolated rabbit colon and relating changes in pressure to those in diameter along the length of the gut section [4].

Another approach involves exploring other generalized regression models. The Gaussian process approach is suitable because it reduces the noisy data to a smoothed signal and the associated posterior estimate can help identify regions of dramatic change (where a phase III begins). However, more a more robust approach would be, for example, a trained learning method employing Artificial Neural Networks with physician-annotated data sets that can estimate an unknown function best representing the manometry signal.

4.2.3 Physiological studies

High spatiotemporal resolution has provided insight in GI function from the perspective of the oral dosage form subject to the influences of the gut: magnetic marker monitoring has been used to quantitate *in vivo* drug release and transit based on magnetic dipole labeling of the solid dosage form[5]. MRI is also being explored as an alternative, non-invasive method with recent success in small bowel flow rate and volumetric emptying studies[6,7]. Direction of the migrating motor complex (MMC) propagation is also of great importance. High temporal resolution studies of both colonic and duodenal manometry revealed a large majority of pressure wave sequences were indeed retrograde[8,9]. This calls into question the assumptions underlying many transit models that account for net forward movement but not the shearing and turbulent effects of segmental back-mixing and aboral movement of GI contents. However, these are still examinations of GI motility's consequences rather than causes.

Slow wave propagation is potentially an underlying phenomenon driving motility but whose exact relationships with the MMC remain yet unknown. Magnetic measurements have been used to characterize what is thought to control smooth muscle activity along the small bowel[10]. Recently, gastric contractions in humans were correlated to gastric slow waves using high-resolution MRI[11]. Contrastingly, a study of tubular and sheet segments of feline duodenum showed the electrical activity of slow waves propagating as broad wave fronts, similar to electrical wave fronts recorded during peristaltic contractions, however at different velocities and with spontaneous interruptions in conduction during peristalsis, something not seen in slow wave propagations[12]. Furthermore, generation and directional propagation of the MMC in isolated mouse small intestines appeared independent of slow wave activity which is seemingly an intrinsic capability of the enteric nervous system[13]. However, tremendous advances in technologies allow for non-invasive and more sensitive detection methods of slow wave frequencies in the small bowel, including biomagnetic signatures assessment using magnetometer measurements in healthy humans[14,15].

Beyond quantification of the MMC and potential underlying electrical activity is the biochemistry controlling motility phases. Morphine is known to act on opioid receptors and initiate phase III activities[16]. It was shown in fasted dogs that as bands of action potential activity migrating caudally reached the ileum new bands began in the stomach or duodenum[17–20]. While contractile frequencies of phase III remained constant in the upper small intestines, the propagation of the MMC declined aborally and most steeply in the distal jejunum[21]. Sarna et al. correlated plasma levels of motilin–thought to be one of the most important factors controlling motility[23]–with contractile activity in fasted dogs. They found that both spontaneous (i.e. natural) and morphine-induced phase III patterns were associated with high levels of plasma motilin which subsequently declined during phase I[22] Furthermore, polymorphisms of endogenous factors have been reported previously and implicated in certain GI disorders[24–28]. To be certain, characterization of normal GI function is still an ongoing endeavor and will continue to be the focus of many interdisciplinary studies to come. As new technologies continue to develop, a better understanding of these various components of motility will ultimately enhance the ability to predict the fate of oral drug products with greater confidence and accuracy.

4.3 References

[1] Yanxing Wang, Bertil Abrahamsson, Lennart Lindfors, and James G. Brasseur. Comparison and analysis of theoretical models for diffusion-controlled dissolution. *Molecular Pharmaceutics*, 9(5):1052–1066, 2012. ISSN 15438384. doi: 10.1021/mp2002818.

[2] Luciana Lisa Lao, Nicholas A. Peppas, Freddy Yin Chiang Boey, and Subbu S. Venkatraman. Modeling of drug release from bulk-degrading polymers, 2011. ISSN 03785173.

[3] Jianzhuo Wang and Douglas R. Flanagan. General solution for diffusion-controlled dissolution of spherical particles. 1. Theory. *Journal of Pharmaceutical Sciences*, 88(7):731–738, 1999. ISSN 00223549. doi: 10.1021/js980236p.

[4] Lukasz Wiklendt, Marcello Costa, and Phil G Dinning. Inference of mechanical states of intestinal motor activity using hidden Markov models. *BMC physiology*, 13:14, 2013. ISSN 1472-6793. doi: 10.1186/1472-6793-13-14. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3909344&tool=pmcentrez&rendertype=abstract`.

[5] Werner Weitschies, Henning Blume, and Hubert Mönnikes. Magnetic Marker Monitoring: High resolution real-time tracking of oral solid dosage forms in the gastrointestinal tract, 2010. ISSN 09396411.

[6] Andreas Gutzeit, Michael A. Patak, Constantin Von Weymarn, Nicole Graf, Aleksis Doert, Edwin Willemse, Christoph A. Binkert, and Johannes M. Froehlich. Feasibility of small bowel flow rate measurement with MRI. *Journal of Magnetic Resonance Imaging*, 32(2):345–351, 2010. ISSN 10531807. doi: 10.1002/jmri.22254.

[7] Deanna Mudie, Kathryn Murray, Caroline L Hoad, Susan E Pritchard, Martin Garnett, Gordon L Amidon, Penny A Gowland, Robin C Spiller, Gregory E Amidon, and Luca Marciani. Quantification of gastrointestinal liquid volumes and distribution following a 240 mL dose of water in the fasted state. *Mol Pharm*, 11(9):3039–47, September 2014. ISSN 15438392. doi: 10.1021/mp500210c. URL `http://www.ncbi.nlm.nih.gov/pubmed/25115349`.

[8] J. M. Andrews, D. G. O'Donovan, G. S. Hebbard, C. H. Malbert, S. M. Doran, and J. Dent. Human duodenal phase III migrating motor complex activity is predominantly antegrade, as revealed by high-resolution manometry and colour pressure plots. *Neurogastroenterology and Motility*, 14(4):331–338, 2002. ISSN 13501925. doi: 10.1046/j.1365-2982.2002.00337.x.

[9] P. G. Dinning, L. Wiklendt, I. Gibbins, V. Patton, P. Bampton, D. Z. Lubowski, I. J. Cook, and J. W. Arkwright. Low-resolution colonic manometry leads to a gross misinterpretation of the frequency and polarity of propagating sequences: Initial results from fiber-optic high-resolution manometry studies. *Neurogastroenterology and Motility*, 25 (10), 2013. ISSN 13501925. doi: 10.1111/nmo.12170.

[10] A.G. Myers, L.A. Bradshaw, J.G. McDowell, and W.O. Richards. Magnetic vector analysis of gastrointestinal electrical control activity. *Proceedings of the Second Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society] [Engineering in Medicine and Biology*, 2, 2002. ISSN 1094-687X. doi: 10.1109/IEMBS.2002.1106623.

[11] Gregory O'Grady, Peng Du, Leo K Cheng, John U Egbuji, Wim J E P Lammers, John A Windsor, and Andrew J Pullan. Origin and propagation of human gastric slow-wave activity defined by high-resolution mapping. *American journal of physiology. Gastrointestinal and liver physiology*, 299(3):G585–G592, 2010. ISSN 0193-1857. doi: 10.1152/ajpgi.00125.2010.

[12] Wim J E P Lammers, Betty Stephen, and John R Slack. Similarities and differences in the propagation of slow waves and peristaltic waves. *American journal of physiology. Gastrointestinal and liver physiology*, 283(3):G778–G786, 2002. ISSN 0193-1857. doi: 10.1152/ajpgi.00390.2001.

[13] Nick J Spencer, Kenton M Sanders, and Terence K Smith. Migrating motor complexes do not require electrical slow waves in the mouse small intestine. *The Journal of physiology*, 553(Pt 3):881–893, 2003. ISSN 0022-3751. doi: 10.1113/jphysiol.2003.049700.

[14] L. Alan Bradshaw. Biomagnetic Techniques for Assessing Gastric and Small Bowel Electrical Activity. In *AIP Conference Proceedings*, volume 724, pages 8–13, 2004. doi: 10.1063/1.1811812. URL `http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.1811812`.

[15] Jonathan C. Erickson, Chibuike Obioha, Adam Goodale, L. Alan Bradshaw, and William O. Richards. Detection of small bowel slow-wave frequencies from noninvasive biomagnetic measurements. *IEEE Transactions on Biomedical Engineering*, 56(9):2181–2189, 2009. ISSN 00189294. doi: 10.1109/TBME.2009.2024087.

[16] T D Lewis. Morphine and gastroduodenal motility. *Digestive diseases and sciences*, 44(11):2178–2186, 1999. ISSN 0163-2116.

[17] G M Carlson, B S Bedi, and C F Code. Mechanism of propagation of intestinal interdigestive myoelectric complex. *The American journal of physiology*, 222(4):1027–1030, 1972. ISSN 00029513.

[18] F. C. Code and J. A. Marlett. The interdigestive myo-electric complex of the stomach and small bowel of dogs. *Journal of Physiology*, 246(2):289–309, 1975.

[19] ML Grivel and Y Ruckebusch. The propagation of segmental contractions along the small intestine. *The Journal of physiology*, 227(2):611–25, December 1972. ISSN 0022-3751. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1331213&tool=pmcentrez&rendertype=abstracthttp://jp.physoc.org/content/227/2/611.short`.

[20] J H Szurszewski. A migrating electric complex of canine small intestine. *The American journal of physiology*, 217 (6):1757–1763, 1969. ISSN 0002-9513.

[21] M L Siegle, S Bühner, M Schemann, H R Schmid, and H J Ehrlein. Propagation velocities and frequencies of contractions along canine small intestine. *The American journal of physiology*, 258(5 Pt 1):G738–G744, 1990. ISSN 0002-9513.

[22] S Sarna, W Y Chey, R E Condon, W J Dodds, T Myers, and T M Chang. Cause-and-effect relationship between motilin and migrating myoelectric complexes. *The American journal of physiology*, 245(2):G277–G284, 1983. ISSN 00029513.

[23] Zen Itoh. Motilin and clinical application, 1997. ISSN 01969781.

[24] V Annese, A Piepoli, A Andriulli, G Napolitano, L Bisceglia, L Zelante, and P Gasparini. Polymorphism of motilin gene in patients with Crohn's disease. *Digestive diseases and sciences*, 43(4):715–719, 1998.

[25] D I Daikh, J O Douglass, and J P Adelman. Structure and expression of the human motilin gene. *DNA (Mary Ann Liebert, Inc.)*, 8(8):615–621, 1989. ISSN 0198-0238. doi: 10.1089/dna.1989.8.615.

[26] I. Depoortere, P. De Clercq, M. Svoboda, L. Bare, and T. L. Peeters. Identification of motilin mRNA in the brain of man and rabbit. Conservation of polymorphism of the motilin gene across species. *Peptides*, 18(10):1497–1503, 1997. ISSN 01969781. doi: 10.1016/S0196-9781(97)00227-1.

[27] P Gasparini, A Grifa, S Savasta, I Merlo, L Bisceglia, A Totaro, and L Zelante. The motilin gene: subregional localisation, tissue expression, DNA polymorphisms and exclusion as a candidate gene for the HLA-associated immotile cilia syndrome. *Human genetics*, 94(6):671–674, 1994. ISSN 0340-6717. doi: 10.1007/BF00206962.

[28] Siyan Xu, Steven Y. Hua, Ronald Menton, Kerry Barker, Sandeep Menon, and Ralph B. D'Agostino. Inference of bioequivalence for log-normal distributed data with unspecified variances. *Statistics in Medicine*, 33(17):2924–2938, 2014. ISSN 02776715. doi: 10.1002/sim.6081. URL `http://doi.wiley.com/10.1002/sim.6081`.

APPENDICES

APPENDIX A

Sample Python Code for Motility, Dissolution, and Signal Analyses

A.1 Gastrointestinal motility transit

A.1.1 Mass balance Framework

```
# Import required libraries
import numpy as np
from scipy.integrate import odeint
from sympy.functions.special.delta_functions import Heaviside
import time

# Define custom Heaviside function
def u(t):
  t = np.asarray(t)
  is_scalar = False if t.ndim > 0 else True
  t.shape = (1,)*(1-t.ndim) + t.shape
  unit_step = np.arange(t.shape[0])
  lcv = np.arange(t.shape[0])
  for place in lcv:
    if t[place] == 0:
      unit_step[place] = .5
    elif t[place] > 0:
      unit_step[place] = 1
    elif t[place] < 0:
      unit_step[place] = 0
  return (unit_step if not is_scalar else (unit_step[0] if
   unit_step else Heaviside(t)))
```

```
# System of equations
class massBalance:

  def __init__(self, t, to, Kpel, PermRates, KgeParams, LagParams, vol, M0, title):
    self.title = title

    self.vol = vol

    self.t = t
    self.to = to

    self.M0 = M0

    # Gastric emptying parameters
    self.p150,self.p1200 = KgeParams[0],KgeParams[6]
    self.p250, self.p2200 = KgeParams[1],KgeParams[7]
    self.p350, self.p3200 = KgeParams[2],KgeParams[8]
    self.theta50, self.theta200 = KgeParams[3],KgeParams[9]
    self.tau50, self.tau200 = KgeParams[4],KgeParams[10]
    self.s50, self.s200 = KgeParams[5],KgeParams[11]

    # Delay parameters
    self.a50,self.a200 = LagParams[0],LagParams[7]
    self.b50,self.b200 = LagParams[1],LagParams[8]
    self.c50,self.c200 = LagParams[2],LagParams[9]
    self.d50,self.d200 = LagParams[3],LagParams[10]
    self.e50,self.e200 = LagParams[4],LagParams[11]
    self.f50,self.f200 = LagParams[5],LagParams[12]
    self.g50,self.g200 = LagParams[6],LagParams[13]

    # Effective permeation rates
    self.PermRates = PermRates
    self.perm0,self.perm1,self.perm2,self.perm3,self.perm4,self.perm5,self.perm6 = self.PermRates

    # MMC decreases as it propagates along GI tract
    self.KIntFactors = np.array([0.77, 0.76, 0.75, 0.74, 0.73, 0.72, 0.7],float)

    # Plasma elimination half-life
    self.Kpel = Kpel

  # Gastric emptying functions for 50mL and 200mL volumes
  def kge50ml(self,t):
    sum = 0.0
    for k in range(1,26):
      sum += ((-1.0)**k*np.sin(-self.theta50*np.pi*k*(t-self.tau50)))/k+self.p150
    return self.p250*sum**self.p350+self.s50
  def kge200ml(self,t):
    sum = 0.0
    for k in range(1,26):
```

```
            sum += ((-1.0)**k*np.sin(-self.theta200*np.pi*k*(t-self.tau200)))/k+self.p1200
        return self.p2200*sum**self.p3200+self.s200

    # Delay functions for 50mL and 200mL volumes
    def tlag50ml(self,t):
        return (self.a50 - self.b50/(self.c50+self.d50*np.exp(-self.e50*np.mod(t,2.0/self.theta50)+self.f50)))*self.g50
    def tlag200ml(self,t):
        return (self.a200 - self.b200/(self.c200+self.d200*np.exp(-self.e200*np.mod(t,2.0/self.theta200)+self.f200)))*self.g200

    # Gastroretentitive effect reducing rate of emptying function
    # for small particules and solutions
    def Kge(self,t,to):
        T = np.mod(t+to,120)
        if (self.vol>=200): return u(T-self.tlag200ml(t+to))*self.kge200ml(t+to)
        else: return u(T-self.tlag50ml(t+to))*self.kge50ml(t+to)

    # Intestinal transit functions
    def KInt0(self,t,to):
        return self.KIntFactors[0]*(u(np.mod(t+to,120)-self.tlag50ml(t+to))*self.kge50ml(t+to))
    def KInt1(self,t,to):
        return self.KIntFactors[1]*(u(np.mod(t+to,120)-self.tlag50ml(t+to))*self.kge50ml(t+to))
    def KInt2(self,t,to):
        return self.KIntFactors[2]*(u(np.mod(t+to,120)-self.tlag50ml(t+to))*self.kge50ml(t+to))
    def KInt3(self,t,to):
        return self.KIntFactors[3]*(u(np.mod(t+to,120)-self.tlag50ml(t+to))*self.kge50ml(t+to))
    def KInt4(self,t,to):
        return self.KIntFactors[4]*(u(np.mod(t+to,120)-self.tlag50ml(t+to))*self.kge50ml(t+to))
    def KInt5(self,t,to):
        return self.KIntFactors[5]*(u(np.mod(t+to,120)-self.tlag50ml(t+to))*self.kge50ml(t+to))
    def Kie(self,t,to):
        return self.KIntFactors[6]*(u(np.mod(t+to,120)-self.tlag50ml(t+to))*self.kge50ml(t+to))

    # Backflow functions
    def Q1(self,t,to):
        return .15*self.KInt1(t,to)
    def Q2(self,t,to):
        return .15*self.KInt3(t,to)
    def Q3(self,t,to):
        return .15*self.KInt5(t,to)

    # System of equations
    def dF(self,F,t):
        self.F = F
```

```
        self.t = t
        DSolns, DSolnInt0, DSolnInt1, DSolnInt2, DSolnInt3, DSolnInt4, DSolnInt5, DSolnInt6, DSolnPlasma = F[0:9]
        self.Mnew = np.array([-DSolns*self.Kge(t,self.to), \
          DSolns*self.Kge(t,self.to) - DSolnInt0*(self.KInt0(t,self.to) + self.perm0), \
          DSolnInt0*self.KInt0(t,self.to) - DSolnInt1*(self.KInt1(t,self.to) + self.perm1) + DSolnInt2*self.Q1(t,self.to), \
          DSolnInt1*self.KInt1(t,self.to) - DSolnInt2*(self.KInt2(t,self.to) + self.Q1(t,self.to) + self.perm2), \
          DSolnInt2*self.KInt2(t,self.to) - DSolnInt3*(self.KInt3(t,self.to) + self.perm3) + DSolnInt4*self.Q2(t,self.to), \
          DSolnInt3*self.KInt3(t,self.to) - DSolnInt4*(self.KInt4(t,self.to) + self.Q2(t,self.to) + self.perm4), \
          DSolnInt4*self.KInt4(t,self.to) - DSolnInt5*(self.KInt5(t,self.to) + self.perm5) + DSolnInt6*self.Q3(t,self.to), \
          DSolnInt5*self.KInt5(t,self.to) - DSolnInt6*(self.Kie(t,self.to) + self.Q3(t,self.to) + self.perm6), \
          DSolnInt6*self.Kie(t,self.to)],dtype='float')
#         DSolnInt0*self.perm0 + DSolnInt1*self.perm1 + DSolnInt2*self.perm2 + DSolnInt3*self.perm3 + DSolnInt4*self.perm4 + DSolnInt5*self.perm5 + DSolnInt6*self.perm6 - DSolnPlasma*self.Kpel],dtype='float')
        return self.Mnew
    # Jacobian of matrix F defined above (right now this is not used)
    def Fjac(self,F,t):
        self.t = t
        Fmat = np.array([[self.Kge(t,self.to),0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0], \
          [self.Kge(t,self.to), -(self.KInt0(t,self.to) + self.perm0),0.0,0.0,0.0,0.0,0.0,0.0,0.0], \
          [0.0,self.KInt0(t,self.to), -(self.KInt1(t,self.to) + self.perm1), self.Q1(t,self.to),0.0,0.0,0.0,0.0,0.0], \
          [0.0,0.0,self.KInt1(t,self.to), -(self.KInt2(t,self.to) + self.Q1(t,self.to) + self.perm2),0.0,0.0,0.0,0.0,0.0], \
          [0.0,0.0,0.0,self.KInt2(t,self.to), -(self.KInt3(t,self.to) + self.perm3), self.Q2(t,self.to),0.0,0.0,0.0], \
          [0.0,0.0,0.0,0.0,self.KInt3(t,self.to), -(self.KInt4(t,self.to) + self.Q2(t,self.to) + self.perm4),0.0,0.0,0.0], \
          [0.0,0.0,0.0,0.0,0.0,self.KInt4(t,self.to), -(self.KInt5(t,self.to) + self.perm5), self.Q3(t,self.to),0.0], \
          [0.0,0.0,0.0,0.0,0.0,0.0,self.KInt5(t,self.to), -(self.Kie(t,self.to) + self.Q3(t,self.to) + self.perm6),0.0], \
          [0.0,0.0,0.0,0.0,0.0,0.0,0.0,self.Kie(t,self.to),0]],'float')
#         np.concatenate(([0.0],self.PermRates,[self.Kpel]))],'float')
        return Fmat
    # Method call to solve the system of equations
    # relative and absolute tolerances set to 10^-3
    def solveSys(self,t):
        self.t = t
        # Solve system of equations
        start_time = time.time()
```

```
    result = odeint(self.dF,self.M0,t,rtol=1e-3, atol=1e-3) #
   mxstep=500,hmax=100,Dfun=self.Fjac,full_output=True,
    #print("--- %s seconds ---" % np.str(time.time() - start_time)
   )
    return result
```

./AppendixA/massBalanceBE.py

A.1.2 BCS Class I 50 mL Volume Model

```
# Import required libraries
import numpy as np
from scipy.integrate import odeint, simps
from sympy.functions.special.delta_functions import Heaviside
import time, sys
from massBalanceBE import *
from joblib import Parallel, delayed
import multiprocessing
import csv
import pickle
import bz2
from contextlib import closing

init_time = time.time()

'''
Define functions for saving/opening multiple files from/into large
    array
'''
def mySaveFile(title,data):
  # Write the array to disk
  with file(title, 'w') as outfile:
    # I'm writing a header here just for the sake of readability
    # Any line starting with "#" will be ignored by numpy.loadtxt
    outfile.write('# Array shape: {0}\n'.format(np.shape(data)))

    # Iterating through a ndimensional array produces slices along
    # the last axis. This is equivalent to data[i,:,:] in this
   case
    for data_slice in data:

      # The formatting string indicates that I'm writing out
      # the values in left-justified columns 7 characters in width
      # with 2 decimal places.
      np.savetxt(outfile, data_slice, fmt='%-7.2f')

      # Writing out a break to indicate different slices...
      outfile.write('# New slice\n')

def myLoadFile(fname,shape):
```

```
    # Read the array from disk
    new_data = np.loadtxt(fname)

    # Note that this returned a 2D array!
    print new_data.shapest

# Gastric emptying parameters
p150,p1200 = 0.15, .14
p250, p2200 = .285/(np.pi*15000), .16/(np.pi*3900000)
p350, p3200 = 6.65, 10.4
theta50, theta200 = 1/60.0, 1/60.0
tau50, tau200 = 59.4, 60
s50, s200 = 0.0001, 0.1
KgeParams = np.array([p150,p250,p350,theta50,tau50,s50,p1200,p2200,p3200,theta200,tau200,s200],float)

# Delay parameters
a50,a200 = 2190/50.,2591/100.
b50,b200 = 481/10., 141/10.
c50,c200 = 27/25., 11/20.
d50,d200 = 531/25. ,138/5.
e50,e200 = 13/200. ,6/25.
f50,f200 = 41/100., -59/50.
LagParams = np.array([a50,b50,c50,d50,e50,f50,a200,b200,c200,d200,e200,f200],float)

# Randomly generate parameters for 23 more virtual subjects
randKgeParam = []
randKgeParam.append(KgeParams*np.transpose([np.concatenate(([1.0],np.array((10-np.random.uniform(-1,1,23))/10,float)))]))
randKgeParam = randKgeParam[0]
randLagParam = []
randLagParam.append(LagParams*np.transpose([np.concatenate(([1.0],np.array((10-np.random.uniform(-1,1,23))/10,float)))]))
randLagParam = randLagParam[0]

dose = 100.0  # Initial doses
vol = 50    # Initial volume

# 7, 14, 30, 60, and 120 minute plasma elimination half-lives
KpelVals = np.array([0.1, .05, .0231, .011552, .002888],float)

# 4000-minute time range
t = np.arange(0,4000,1)

title = 'BCS I Slow $50mL$'

# Constant permeation rate along all compartments
perm0,perm1,perm2,perm3,perm4,perm5,perm6 = np.ones(7)*.5
PermRates = [perm0,perm1,perm2,perm3,perm4,perm5,perm6]

# Initial vector. Only stomach has content = dose
M0 = np.array([dose,0,0,0,0,0,0,0,0],float)
```

```
# Simulate over t0 = 0 to 120 min
inputs = range(121)

# Determine number of available cores for processing
num_cores = multiprocessing.cpu_count()

# Set up function calls to solve each iteration
# Determine concentration profiles in different compartments
# Calculate Cmax, Tmax, and AUC
def solve_to(i):
  to = i
  x = massBalance( t, to, Kpel, PermRates, patientKgeParams, patientLagParams, vol, M0,title)
  Mresult = x.solveSys(t)
  return Mresult
def c_max(data):
  return np.max(data[:,-1])
def t_max(data):
  return t[np.argmax(data[:,-1])]
def area(data):
  return simps(data[:,-1], dx=1)

# Store all results in matrices
MresultList50, cMaxList50, tMaxList50, AUCList50 = [],[],[],[]
start_time = time.time()
totIts = len(KpelVals)*len(range(24))
bar_length = 20
for j in range(5):
  Kpel = KpelVals[j]
  for k in range(24):
    patientKgeParams = randKgeParam[k]
    patientLagParams = randLagParam[k]
    result = Parallel(n_jobs=num_cores)(delayed(solve_to)(i) for i in inputs)
    cMax = Parallel(n_jobs=num_cores)(delayed(c_max)(result[i]) for i in inputs)
    tMax = Parallel(n_jobs=num_cores)(delayed(t_max)(result[i]) for i in inputs)
    AUC = Parallel(n_jobs=num_cores)(delayed(area)(result[i]) for i in inputs)
    MresultList50.append(np.array(result)[:,:,-1])
    cMaxList50.append(np.array(cMax))
    tMaxList50.append(np.array(tMax))
    AUCList50.append(np.array(AUC))

    percent = float(((j)*24.+(k+1))/totIts)
    hashes = '#' * int(round(percent * bar_length))
    spaces = ' ' * (bar_length - len(hashes))
    runtime = round(time.time()-start_time,2)
    sys.stdout.write("\rPercent: [{0}] {1}% completed in {2} seconds".format(hashes + spaces, int(round(percent * 100)), runtime))
    sys.stdout.flush()
```

```
print("--- Parallelized execution completed: %s seconds ---" %
    float(time.time() - float(start_time)))

# Print summary
print("--- Name: %s  ---" % title)
print("--- Data sizes:   ---")
print("Solutions: %s"  % repr(np.shape(MresultList50)))
print("cMax: %s"%repr(np.shape(cMaxList50)))
print("tMax: %s"%repr(np.shape(tMaxList50)))
print("AUC: %s"%repr(np.shape(AUCList50)))

'''
Save files for conc profiles, cmax, tmax, and auc
'''

fname = title.replace(" ","").replace('$','').replace('_','').
    replace('$','').replace('{','').replace('}','').replace('in','
    _')
fnameConc,fnamecMax,fnametMax,fnameAUC = fname+'_conc',fname+'
    _cMax',fname+'_tMax',fname+'_AUC'

print("--- Saving resutls: %s ---" % fname)
start_time = time.time()
totIts = len(KpelVals)*len(range(24))
bar_length = 20
for j in range(5):
  for k in range(24):
    np.savetxt(fnameConc + '_Elim' + repr(j+1) + '_P_' + repr(k+1)
    + '.csv', MresultList50[(j)*24+k], delimiter=',')
    np.savetxt(fnamecMax + '_Elim' + repr(j+1) + '_P_' + repr(k+1)
    + '.csv', cMaxList50[(j)*24+k], delimiter=',')
    np.savetxt(fnametMax + '_Elim' + repr(j+1) + '_P_' + repr(k+1)
    + '.csv', tMaxList50[(j)*24+k], delimiter=',')
    np.savetxt(fnameAUC + '_Elim' + repr(j+1) + '_P_' + repr(k+1)
   + '.csv', AUCList50[(j)*24+k], delimiter=',')

    percent = float(((j)*24.+(k+1))/totIts)
    hashes = '#' * int(round(percent * bar_length))
    spaces = ' ' * (bar_length - len(hashes))
    runtime = round(time.time()-start_time,2)
    sys.stdout.write("\rPercent: [{0}] {1}% completed in {2}
   seconds".format(hashes + spaces, int(round(percent * 100)),
   runtime))
    sys.stdout.flush()

print("--- Completed! Total execution time: %s seconds ---" %
    float(time.time() - float(init_time)))
```

./AppendixA/BCSI50.py

A.2 Dissolution mass balance analysis

A.2.1 Mass Balance Framework

```
# Import required libraries
import ipybell
import json, matplotlib
s = json.load( open("/Users/arjang/.matplotlib/bmh_matplotlibrc.
    json") )
matplotlib.rcParams.update(s)

%matplotlib inline
from IPython.core.pylabtools import figsize
import numpy as np
from scipy.integrate import odeint, simps
from sympy.functions.special.delta_functions import Heaviside
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure, show, axes, sci
from matplotlib import cm, colors
from matplotlib.font_manager import FontProperties
from numpy import amin, amax, ravel
import seaborn as sns

import time, sys
from Ibuprofen_massBalance import *
from matplotlib.patches import Polygon

%%bell

# Permeation rates, cm^2*s^-1 / 1.143
perm0,perm1,perm2,perm3,perm4,perm5,perm6 = np.ones(7)*8e-4 # 2.0e
    -3
PermRates = [perm0,perm1,perm2,perm3,perm4,perm5,perm6]

Kpel = 8.557e-5          # 2.25 hr plasma elimination half-life
# Kpel = 7.7016e-5     # 2.5hr plasma elimination half-life
# Kpel = 9.627e-5    # 2hr plasma elimination half-life
Kdiss = 46.8/60/60     # Fast disintegration (90% in 3 min)

# Resting volumes (mL)
Grest = 30.0
Irest = 10.0 # 40.0
RestVol = [Grest,Irest]

vol0 = 240.0    # Liquid volume
dose = 200.0   # Initial dose (mg)
ro = 400.0     # Initial particulate radius (um)

# Initial condition matrix
M0 = [vol0 + Grest, Irest, Irest, Irest, Irest, Irest,Irest, Irest
    ,\
  dose,0,0,0,0,0,0,0,\
```

```
  0,0,0,0,0,0,0,0, \
  0,0,0,0,0,0,0,0,0]

# Time vector
t = np.linspace(0,800*60,800*60*2)
# Index of time equaling 4 plasma elimination half-lives
tindex = min(range(len(t)), key=lambda i: np.abs(t[i]-4*np.log(2)/
    Kpel))

#to_vals = np.array([0.0*60,30*60,75.0*60,100.0*60,115.0*60],dtype
    ='float')
to_vals = np.arange(0,121,10)*60.0
dose_vals = np.array([200,400,800],dtype='float')
phase_names = ['Early Phase I','Mid Phase I','Phase II','Early
    Phase III', 'Late Phase III']
lbls = ["Stom","Int0","Int1","Int2","Int3","Int4","Int5","Int6"]
conc_results,cMax_results,tMax_results,AUC_results = np.empty((3,
    len(to_vals),
        len(t))),np.empty((3,len(to_vals))),np.empty((3,len(
    to_vals))),
        np.empty((3,len(to_vals)))
MresultsMatrix = np.empty((len(t),33,len(to_vals)))
title = 'Ibuprofen 200 400 800'

'''
Ibuprofen formulations
'''
%%bell
start_time = time.time()
totIts = len(dose_vals)*len(to_vals)
bar_length = 20
# for D in range(len(dose_vals)):
for D in range(3):
  for To in range(len(to_vals)):
    percent = float(((1.0*D)*len(to_vals)+(To))/totIts)
    hashes = '#' * int(round(percent * bar_length))
    spaces = ' ' * (bar_length - len(hashes))
    runtime = round(time.time()-start_time,2)
    sys.stdout.write("\rPercent: [{0}] {1}% completed in {2}
    seconds".format(hashes + spaces, int(round(percent * 100)),
    runtime))
    sys.stdout.flush()

#     print('--- Dose: %.2f\t\tTo: %.2f ---' % (dose_vals[D],
    to_vals[To]/60.0))
    to = to_vals[To]
    M0[8] = dose_vals[D]
    x = massBalance(t,to,Kpel,Kdiss,ro,RestVol,PermRates,M0,title)
    Mresult = x.solveSys(t)
    MresultsMatrix[:,:,To] = np.copy(Mresult)
    tMax = np.copy(t[np.argmax(Mresult[:,-1])]/60)
    cMax = np.copy(np.max(Mresult[:,-1])/(.1*70))
    AUC= np.copy(np.trapz(Mresult[:,-1]/(.1*70),t/60.0)/60)
    conc_results[D,To,:] = np.copy(Mresult[:,-1])
```

```
    cMax_results[D,To] = np.copy(cMax)
    tMax_results[D,To] = np.copy(tMax)
    AUC_results[D,To] = np.copy(AUC)
print('--- Executiton time: %.4f ---' % float(time.time()-start_time))

'''
Phenol red 65mg dosed
as solution in 240mL liquid volume
Treat as instantaneous dissintegration
and dissolution (on order fo 1e3)
'''
%%bell
start_time = time.time()
totIts = len(dose_vals)
for D in range(1):
  for To in range(len(to_vals)):
    print('--- Dose: %.2f\t\tTo: %.2f ---' % (65,to_vals[To]/60.0))
    to = to_vals[To]
    M0[8] = 65
    x = massBalance(t,to,Kpel,Kdiss*1e3,ro,RestVol,PermRates*np.zeros(7),M0,title)
    x.C0 = 1e3
    Mresult = x.solveSys(t)
    MresultsMatrix[:,:,To] = np.copy(Mresult)
    tMax = np.copy(t[np.argmax(Mresult[:,-1])]/60)
    cMax = np.copy(np.max(Mresult[:,-1])/(.1*70))
    AUC= np.copy(np.trapz(Mresult[:,-1]/(.1*70),t/60.0)/60)
    conc_results[D,To,:] = np.copy(Mresult[:,-1])
    cMax_results[D,To] = np.copy(cMax)
    tMax_results[D,To] = np.copy(tMax)
    AUC_results[D,To] = np.copy(AUC)
print('--- Executiton time: %.4f ---' % float(time.time()-start_time))

'''
mass / (Vd * Wt)
mg / (L/kg * kg)
'''

cMax_results2 = np.concatenate(([Ibu200_cMax],[Ibu400_cMax],[Ibu800_cMax]))
tMax_results2 = np.concatenate(([Ibu200_tMax],[Ibu400_tMax],[Ibu800_tMax]))/60.0
conc_results2 = np.concatenate(([Ibu200_conc],[Ibu400_conc],[Ibu800_conc]))/(.14*70)
# cMax_results2,tMax_results2, conc_results2 = cMax_results, tMax_results,conc_results

# Plot results
pI,pII,pIIIe,pIIIl = 0,70,100,119
figsize(8.5,6)
```

```
f, ax = plt.subplots(nrows=3, ncols=1, sharex=True, sharey=True, dpi=300)
plt.subplots_adjust(hspace=0.1)
# plt.bone()
for i in range(3):
  ax[i].fill_between(t/60, np.min(conc_results2[i,:],axis=0),
    np.max(conc_results2[i,:],axis=0), facecolor='#444444', alpha=0.15)
  ax[i].plot(t/60,conc_results2[i,pI],label=r'Early Phase I')
  ax[i].plot(t/60,conc_results2[i,pII],label=r'Phase II')
  ax[i].plot(t/60,conc_results2[i,pIIIe],label=r'Early Phase III')
  ax[i].plot(t/60,conc_results2[i,pIIIl],label=r'Late Phase III')
  ax[i].plot(tMax_results2[i,pI],cMax_results2[i,pI],'ro')
  ax[i].plot(tMax_results2[i,pII],cMax_results2[i,pII],'ro')
  ax[i].plot(tMax_results2[i,pIIIe],cMax_results2[i,pIIIe],'ro')
  ax[i].plot(tMax_results2[i,pIIIl],cMax_results2[i,pIIIl],'ro')
  if i==2: ax[i].set_xlabel('time ($min$)')
  if i==0: ax[i].set_title('Ibuprofen Phase-Dependent\nPlasma Profile')
  if i==1: ax[i].set_ylabel('Conc. ($mg/L$)')
  props = dict(boxstyle='round', facecolor='white', alpha=0.5)
  ax[0].text(240,80,'200mg',size=12,verticalalignment='top',horizontalalignment='left',bbox=props)
  ax[1].text(240,80,'400mg',size=12,verticalalignment='top',horizontalalignment='left',bbox=props)
  ax[2].text(240,80,'800mg',size=12,verticalalignment='top',horizontalalignment='left',bbox=props)
  # place a text box in upper left in axes coords
  if i==2: leg = ax[i].legend(bbox_to_anchor=(0.25, -.7, 1., .102), loc=3,ncol=2, borderaxespad=0.)
plt.ylim(0,100)
plt.xlim(0,500)
plt.savefig('IbupPlasmaProfile.png',bbox_inches='tight',dpi=300)
plt.show()

# x.savePlots(Mresult,t,title)

# Function for visualizaton
def plot_results(Mresult,title):
  # Plot results
  tindex = min(range(len(t)), key=lambda i: np.abs(t[i]-4*np.log(2)/Kpel))
  fig, axes = plt.subplots(dpi=300, nrows=4, ncols=1, sharex=False, sharey=False)
  n=8
  # Fluid volumes
  color=iter(cm.rainbow(np.linspace(0,1,n)))
  axes[0].set_title(title + '\nDelayed Gastric Emptying')
  for i in range(8):
    c=next(color)
    axes[0].plot(t[0:tindex]/60,Mresult[0:tindex,i],c=c,label=lbls[i])
  axes[0].set_ylabel('Vol ($mL$)')
  axes[0].set_ylim(-5,Grest+vol0+5)
```

```
  # Solid drug
  color=iter(cm.rainbow(np.linspace(0,1,n)))
  for i in range(8):
    c=next(color)
    axes[1].plot(t[0:tindex]/60,Mresult[0:tindex,i+8],c=c,label=lbls[i])
  axes[1].set_ylabel('Solid Drug ($mg$)')
  axes[1].set_ylim(-5,dose+5)
  color=iter(cm.rainbow(np.linspace(0,1,n)))
  for i in range(8):
    c=next(color)
    axes[2].plot(t[0:tindex]/60,Mresult[0:tindex,i+16],c=c,label=lbls[i])
  axes[2].set_ylabel('Drug Particulate ($mg$)')
  axes[2].set_ylim(-5,dose+5)
  color=iter(cm.rainbow(np.linspace(0,1,n)))
  n=9
  color=iter(cm.rainbow(np.linspace(0,1,n)))
  for i in range(8):
    c=next(color)
    axes[3].plot(t[0:tindex]/60,Mresult[0:tindex,i+24]*1000/Mresult[0:tindex,i],c=c,label=lbls[i])
  maxConc = np.ceil(np.max([Mresult[0:tindex,i+24]*1000 for i in range(8)])/10)*10
  axes[3].set_xlabel('time ($min$)')
  axes[3].set_ylabel('Dissolved Drug ($\mu g/mL$)')
  axes[3].set_ylim(1e-4,maxConc)
  axes[3].set_yscale('log')
  leg = axes[3].legend(bbox_to_anchor=(0.0, -.95, 1., .102), loc=3,
      ncol=3, mode="expand", borderaxespad=0.)
  plt.tight_layout()
  # plt.savefig(title,bbox_inches='tight',bbox_extra_artist=[leg],dpi=300)
  plt.show()

def plot_cmax(Mresult,title):
  plt.figure(dpi=300)
  #     fig = plt.figure(figsize=(6,4), dpi=80, facecolor='w', edgecolor='k')
  tMax,cMax = t[np.argmax(Mresult[:,-1])],np.max(Mresult[:,-1]/(.14*70))
  plt.bone()
  plt.plot(t/60,Mresult[:,-1]/(.14*70))
  plt.plot(tMax/60,cMax,'ro')
  plt.xlabel('time ($min$)')
  plt.ylabel('Drug ($mg/mL$)')
  plt.title(title + '\nPlasma Profile')
  # plt.text(tMax+3,cMax+2, r'$C_{max}$', fontsize=12)
  #     plt.ylim(0,100)
  # place a text box in upper left in axes coords
  # props = dict(boxstyle='round', facecolor='wheat', alpha=0.5)
```

```
  # textstr = '$C_{max}=%.2f mg$\n$T_{max}=%.2f min$\nDisintegration HL=$%.2f min$'%(cMax, tMax,np.log(2)/Kdiss)
  # plt.text(t[-100], 95, textstr, fontsize=9,
  # verticalalignment='top', horizontalalignment='right',bbox=props)
  plt.tight_layout()
  # plt.savefig(title + ' Plasma Profile',bbox_inches='tight',dpi=300)
  plt.show()

sns.set_style("whitegrid")
tt = np.linspace(0,800*60,800*60*2)

'''
Various functions that help visualize distributions of content along
the GI tract (compartments)
Take as input the result concentration profiles from solutions to sys of eq above,
title, color map, and whether to save and under what title

plotIntConcs shows intestinal contents of dissolved solutions
plotIntSolids shows intestinal contents of solid particles
plotIntDissSolids shows both dissolved and particulate contents
'''

class ImageFollower:
    'update image in response to changes in clim or cmap on another image'
    def __init__(self, follower):
        self.follower = follower
    def __call__(self, leader):
        self.follower.set_cmap(leader.get_cmap())
        self.follower.set_clim(leader.get_clim())

def plotIntConcs(Mresult,figtitle,cmap=plt.cm.bone,savePlt=0,saveName=''):
  Nr = 1
  Nc = 8

  fig = figure(dpi=300)

  t = fig.text(0.5, .75, figtitle,
          horizontalalignment='center',
          fontproperties=FontProperties(size=12))

  cax = fig.add_axes([0.95, 0.2, 0.05, 0.45])
  h = 0.5
  w = 0.1
  ax = []
  images = []
  vmin = 1e40
  vmax = -1e40
  for i in range(Nr):
```

```
    for j in range(Nc):
      pos = [0.075 + j*1.1*w, 0.18 + i*1.2*h, w, h]
      a = fig.add_axes(pos)
      a.set_xticklabels([])
      a.grid(False)
      if j == 0:
        a.set_yticks(np.linspace(0,50000,6))
        a.set_yticklabels([np.round(10*k)/10 for k in np.linspace(0,tt[50000]/60.0,6)])
        a.set_ylabel('Time (min)')
      else: a.set_yticklabels([])
      a.set_title(lbls[j])
      # Make some fake data with a range that varies
      # somewhat from one plot to the next.
      data = np.tile(Mresult[0:tindex,j+24]*1000/Mresult[0:tindex,j],(5e1,1)).T
      dd = ravel(data)
      # Manually find the min and max of all colors for
      # use in setting the color scale.
      vmin = min(vmin, amin(dd))
      vmax = max(vmax, amax(dd))
      images.append(a.imshow(data, aspect="auto",interpolation="none",cmap=cmap))

      ax.append(a)

  # Set the first image as the master, with all the others
  # observing it for changes in cmap or norm.

  norm = colors.Normalize(vmin=vmin, vmax=vmax)
  for i, im in enumerate(images):
    im.set_norm(norm)
    if i > 0:
      images[0].callbacksSM.connect('changed', ImageFollower(im))

  # The colorbar is also based on this master image.
  cbar = fig.colorbar(images[0], cax=cax, orientation='vertical')
  cbar.ax.set_ylabel('Conc. ($\mu g/mL$)')
  # We need the following only if we want to run this interactively and
  # modify the colormap:

  axes(ax[0])     # Return the current axes to the first one,
  sci(images[0])  # because the current image must be in current axes.
  if savePlt: plt.savefig(saveName +'.png', bbox_inches='tight', dpi=300)
  plt.show()
########################

def plotIntSolids(Mresult,figtitle,cmap=plt.cm.bone,savePlt=0,saveName=''):
  Nr = 1
  Nc = 8
```

```

fig = figure(dpi=300,figsize=(8.5,4.5))

t = fig.text(0.5, .75, figtitle,
         horizontalalignment='center',
         fontproperties=FontProperties(size=12))

cax = fig.add_axes([0.85, 0.2, 0.05, 0.45])
h = 0.5
w = 0.1
ax = []
images = []
vmin = 1e40
vmax = -1e40
for i in range(Nr):
  for j in range(Nc):
    pos = [0.075 + j*1.1*w, 0.18 + i*1.2*h, w, h]
    a = fig.add_axes(pos)
    a.set_xticklabels([])
    a.grid(False)
    if j == 0:
      a.set_yticks(np.linspace(0,50000,6))
      a.set_yticklabels([np.round(10*k)/10 for k in np.linspace(0,tt[50000]/60.0,6)])
      a.set_ylabel('Time (min)')
    else: a.set_yticklabels([])
    a.set_title(lbls[j])
    # Make some fake data with a range that varies
    # somewhat from one plot to the next.
    data = np.tile(Mresult[0:tindex,j+16]*1000/Mresult[0:tindex,j],(5e1,1)).T
    dd = ravel(data)
    # Manually find the min and max of all colors for
    # use in setting the color scale.
    vmin = min(vmin, amin(dd))
    vmax = max(vmax, amax(dd))
    images.append(a.imshow(data, aspect="auto",interpolation="none",cmap=cmap))

    ax.append(a)

# Set the first image as the master, with all the others
# observing it for changes in cmap or norm.

norm = colors.Normalize(vmin=vmin, vmax=vmax)
for i, im in enumerate(images):
  im.set_norm(norm)
  if i > 0:
    images[0].callbacksSM.connect('changed', ImageFollower(im))

# The colorbar is also based on this master image.
cbar = fig.colorbar(images[0], cax=cax, orientation='vertical')
cbar.ax.set_ylabel('Conc. ($\mu g/mL$)')
```

```
  # We need the following only if we want to run this interactively and
  # modify the colormap:

  axes(ax[0])      # Return the current axes to the first one,
  sci(images[0])   # because the current image must be in current axes.

  if savePlt: plt.savefig(saveName +'.png', bbox_inches='tight', dpi=300)
  plt.show()

########################

def plotIntDissSolids(Mresult,figtitle,savePlt=0,saveName='', thresh=False,threshMax1=200.0,threshMax2=35.0):
  Nr = 1
  Nc = 8

  fig = figure(dpi=300,figsize=(9.5,5))

  t = fig.text(0.3, .75, figtitle,
         horizontalalignment='center',
         fontproperties=FontProperties(size=12))

  cmap1 = plt.cm.afmhot_r
  cmap2 = plt.cm.GnBu
  ax_cb1 = fig.add_axes((0.62, 0.2, 0.05, 0.45))
  ax_cb2 = fig.add_axes((0.77, 0.2, 0.05, 0.45))

  h = 0.5
  w = 0.03
  ax = []
  images1,images2 = [],[]
  vmin1 = 1e40
  vmax1 = -1e40
  vmin2 = 1e40
  vmax2 = -1e40
  for i in range(Nr):
    for j in range(Nc):
      pos1 = [0.075 + j*1.1*w*2, 0.18 + i*1.2*h, w, h]
      pos2 = [0.075 +w + j*1.1*w*2, 0.18 + i*1.2*h, w, h]
      a1 = fig.add_axes(pos1)
      a1.grid(False)
      a1.set_xticklabels([])
      a2 = fig.add_axes(pos2)
      a2.grid(False)
      a2.set_xticklabels([])
      a2.set_yticklabels([])
      if j == 0:
        a1.set_yticks(np.linspace(0,50000,6))
        a1.set_yticklabels([np.round(10*k)/10 for k in np.linspace(0,tt[50000]/60.0,6)])
        a1.set_ylabel('Time (min)')
```

```
#         a2.set_yticks(np.linspace(0,50000,6))
#         a2.set_yticklabels([np.round(10*k)/10 for k in np.linspace(0,tt[50000]/60.0,6)])
#         a2.set_ylabel('Time (min)')
      else:
        a1.set_yticklabels([])

      a1.set_title(lbls[j])
#       a2.set_title(lbls[j])
      # Make some fake data with a range that varies
      # somewhat from one plot to the next.
      data1 = np.tile(Mresult[0:tindex,j+16]*1000/Mresult[0:tindex,j],(5e1,1)).T
      dd1 = ravel(data1)
      data2 = np.tile(Mresult[0:tindex,j+24]*1000/Mresult[0:tindex,j],(5e1,1)).T
      dd2 = ravel(data2)
      # Manually find the min and max of all colors for
      # use in setting the color scale.
      vmin1 = min(vmin1, amin(dd1))
      vmax1 = max(vmax1, amax(dd1))
      vmin2 = min(vmin2, amin(dd2))
      vmax2 = max(vmax2, amax(dd2))
      images1.append(a1.imshow(data1, aspect="auto",interpolation="none",cmap=cmap1,alpha=0.5))
      images2.append(a2.imshow(data2, aspect="auto",interpolation="none",cmap=cmap2,alpha=0.5))

      ax.append(a1)
      ax.append(a2)

  # Set the first image as the master, with all the others
  # observing it for changes in cmap or norm.
  if thresh==True: vmax1=threshMax1
  norm1 = colors.Normalize(vmin=vmin1, vmax=vmax1)
  for i, im in enumerate(images1):
    im.set_norm(norm1)
    if i > 0:
      images1[0].callbacksSM.connect('changed', ImageFollower(im))
  if thresh==True: vmax2=threshMax2
  norm2 = colors.Normalize(vmin=vmin2, vmax=vmax2)
  for i, im in enumerate(images2):
    im.set_norm(norm2)
    if i > 0:
      images2[0].callbacksSM.connect('changed', ImageFollower(im))

  # The colorbar is also based on this master image.
  cbar1 = fig.colorbar(images1[0], cax=ax_cb1, orientation='vertical')
  cbar2 = fig.colorbar(images2[0], cax=ax_cb2, orientation='vertical')
  if thresh==True:
    cbar1.set_ticks(np.arange(0,threshMax1*1.1,threshMax1/10.0))
```

```
    cbar1.ax.set_yticklabels(np.concatenate(([str(i) for i in np.arange(0,threshMax1,threshMax1/10.0)],[str(threshMax1)+'$\leq$'])))
  if thresh==True:
    cbar2.set_ticks(np.arange(0,threshMax2*1.1,threshMax2/10.0))
    cbar2.ax.set_yticklabels(np.concatenate(([str(i) for i in np.arange(0,threshMax2,threshMax2/10.0)],[str(threshMax2)+'$\leq$'])))
  cbar1.ax.set_ylabel('Particulate Conc. ($\mu g/mL$)', labelpad=-1)
  cbar2.ax.set_ylabel('Dissolved Conc. ($\mu g/mL$)', labelpad=-1)
  # We need the following only if we want to run this interactively and
  # modify the colormap:

  axes(ax[0])      # Return the current axes to the first one,
  sci(images1[0])  # because the current image must be in current axes.
#   sci(images2[0])  # because the current image must be in current axes.

  if savePlt: plt.savefig(saveName +'.png', bbox_inches='tight', dpi=300)
  plt.show()

########################
# Look at the spread along GI tract during different phases
pI,pII,pIIIe,pIIIl = 0,7,10,12
tindex = min(range(len(t)), key=lambda i: np.abs(t[i]-4*np.log(2)/Kpel))
plotIntConcs(MresultsMatrix[:,:,pI],'Early Phase I',cmap=plt.cm.Reds,savePlt=True,saveName='PhenolRed_PhaseIEarly')
plotIntConcs(MresultsMatrix[:,:,pII],'Phase II',cmap=plt.cm.Reds,savePlt=True,saveName='PhenolRed_PhaseII')
plotIntConcs(MresultsMatrix[:,:,pIIIe],'Early Phase III',cmap=plt.cm.Reds,savePlt=True,saveName='PhenolRed_PhaseIIIEarly')
plotIntConcs(MresultsMatrix[:,:,pIIIl],'Late Phase III',cmap=plt.cm.Reds,savePlt=True,saveName='PhenolRed_PhaseIIILate')

pI,pII,pIIIe,pIIIl = 0,7,10,12
tindex = np.array([np.abs(t[i]/60 - 250.0) for i in range(len(t))]).argmin() # Only plot until 250min
plotIntDissSolids(MresultsMatrix[:,:,pI],'Early Phase I',savePlt=True,saveName='Ibu_PhaseIEarly',thresh=True,threshMax1=1000,threshMax2=1000)
plotIntDissSolids(MresultsMatrix[:,:,pII],'Phase II',savePlt=True,saveName='Ibu_PhaseII',thresh=True,threshMax1=1000,threshMax2=1000)
plotIntDissSolids(MresultsMatrix[:,:,pIIIe],'Early Phase III',savePlt=True,saveName='Ibu_PhaseIIIEarly',thresh=True,threshMax1=1000,threshMax2=1000)
plotIntDissSolids(MresultsMatrix[:,:,pIIIl],'Late Phase III',savePlt=True,saveName='Ibu_PhaseIIILate',thresh=True,threshMax1=1000,threshMax2=1000)
```

```
#########################

'''
Compare predicted Cmax, Tmax, and AUC to reported values from
    Davies paper
This is all generating a fancy bar chart for 200, 400, and 800 mg
    ibuprofen tablets
'''

exp_200 = np.array([[22.0, 1.5, np.NAN], [21.8, 1.0, np.NAN],
     [27.0, 1.375, 76.0], [19.0, 1.95, 63.0], [14.94, 2.605,
    58.37],
     [20.96, 1.737, 78.41], [25.79, 1.694, 106.2], [24.46, 1.858,
    106.2],
     [23.24, 2.028, 122.8], [23.43, 2.126, 109.8], [16.3, 1.5,
    246.78],
     [19.6, 1.59, 60.3], [19.2, 1.06, 58.8]],dtype='float')
exp_400 = np.array([[27.92,1.1,np.NAN],[30.15,1.165,np.NAN],
     [23.6,2.0,np.NAN],[26.8,2.0,np.NAN
    ],[27.6,1.17,319.8],[22.0,1.5,79.2],

    [21.1,2.6,100.0],[27.9,1.9,103.0],[37.7,1.3,122],[25.6,2.48,113.6],

     [36.4,2.07,128.8],[32.5,1.0,np.NAN
    ],[56,1.38,203],[32.3,1.5,113.2],

    [35.3,1.28,127.53],[29,2.13,105.0],[32.4,1.5,173],[43.0,1.06,429.66],


    [26.7,1.1,114],[42.28,0.93,120.08],[42.55,0.97,118.32],[44.7,0.53,100.87]],
    dtype='float')
exp_800 = np.array([[60.2,1.5,np.NAN
    ],[55.6,1.59,217.78],[61,1.6,98],
     [54.7,1.67,214.3],[45.23,2.56,213.66],[np.NAN,np.NAN,239.7],[
    np.NAN,np.NAN,286.8]],dtype='float')

'''
Import saved results

conc_results,cMax_results,tMax_results,AUC_results = np.empty((3,
    len(to_vals),len(t))),
     np.empty((3,len(to_vals))),np.empty((3,len(to_vals))),np.empty
    ((3,len(to_vals)))

MresultsMatrix = np.empty((3,len(t),33,len(to_vals)))

for i in range(3):
  AUC_results[i,:] = np.loadtxt('IbuprofenData/Abs1e-3_Irest40/
    Ibuprofen200400800_AUC_dose' + str(i+1) +
          '.csv',delimiter=',')
  cMax_results[i,:] = np.loadtxt('IbuprofenData/Abs1e-3_Irest40/
    Ibuprofen200400800_cMax_dose' + str(i+1) +
          '.csv',delimiter=',')
```

```
  tMax_results[i,:] = np.loadtxt('IbuprofenData/Abs1e-3_Irest40/Ibuprofen200400800_tMax_dose' + str(i+1) + '.csv',delimiter=',')
  conc_results[i,:,:] = np.loadtxt('IbuprofenData/Abs1e-3_Irest40/Ibuprofen200400800_conc_dose' + str(i+1) + '.csv',delimiter=',')
#   MresultsMatrix[i,:,:] = np.loadtxt('IbuprofenData/Abs1e-3_Irest40/Ibuprofen200400800_result_dose' + str(i+1) + '.csv',delimiter=',')
'''

# filebase = 'IbuprofenData/Abs1e-3_Irest40/Ibuprofen200400800_'
filebase = 'IbuprofenData/Ibuprofen200400800_'

Ibu200_cMax  = np.genfromtxt(filebase + 'cMax_dose1.csv', delimiter=',')
Ibu200_tMax  = np.genfromtxt(filebase + 'tMax_dose1.csv', delimiter=',')
Ibu200_AUC  = np.genfromtxt(filebase + 'AUC_dose1.csv', delimiter=',')
Ibu400_cMax  = np.genfromtxt(filebase + 'cMax_dose2.csv', delimiter=',')
Ibu400_tMax  = np.genfromtxt(filebase + 'tMax_dose2.csv', delimiter=',')
Ibu400_AUC  = np.genfromtxt(filebase + 'AUC_dose2.csv', delimiter=',')
Ibu800_cMax  = np.genfromtxt(filebase + 'cMax_dose3.csv', delimiter=',')
Ibu800_tMax  = np.genfromtxt(filebase + 'tMax_dose3.csv', delimiter=',')
Ibu800_AUC  = np.genfromtxt(filebase + 'AUC_dose3.csv', delimiter=',')
Ibu200_conc  = np.genfromtxt(filebase + 'conc_dose1.csv', delimiter=',')
Ibu400_conc  = np.genfromtxt(filebase + 'conc_dose2.csv', delimiter=',')
Ibu800_conc  = np.genfromtxt(filebase + 'conc_dose3.csv', delimiter=',')


ibu_200 = np.array([Ibu200_cMax,Ibu200_tMax/60.0/60.0,Ibu200_AUC]).T
ibu_400 = np.array([Ibu400_cMax,Ibu400_tMax/60.0/60.0,Ibu400_AUC]).T
ibu_800 = np.array([Ibu800_cMax,Ibu800_tMax/60.0/60.0,Ibu800_AUC]).T

# ibu_200 = np.array([[cMax_results[0,i],tMax_results[0,i]/60,AUC_results[0,i]] for i in range(13)])
# ibu_400 = np.array([[cMax_results[1,i],tMax_results[1,i]/60,AUC_results[1,i]] for i in range(13)])
# ibu_800 = np.array([[cMax_results[2,i],tMax_results[2,i]/60,AUC_results[2,i]] for i in range(13)])

```

```
data_cMax = [ibu_200[:,0], exp_200[:,0], ibu_400[:,0], exp_400[:,0], ibu_800[:,0], exp_800[:,0]]
data_tMax = [ibu_200[:,1], exp_200[:,1], ibu_400[:,1], exp_400[:,1], ibu_800[:,1], exp_800[:,1]]
data_AUC = [ibu_200[:,2], exp_200[:,2], ibu_400[:,2], exp_400[:,2], ibu_800[:,2], exp_800[:,2]]

# Boxplot of predicted and experimental cMax, tMax, and AUC for 200, 400, and 800 mg doses of ibuprofen

numDists = 6
randomDists = ['200mg',' 400mg', '800mg']

c1, c2, c3 = sns.color_palette("Set1", 3)

fig, ax = plt.subplots(figsize=(5.5,9), nrows=3, ncols=1, dpi=300, sharex=True, sharey=False)
#fig.canvas.set_window_title('A Boxplot Example')
plt.subplots_adjust(left=0.075, right=0.95, top=0.9, bottom=0.25)

bp0 = ax[0].boxplot(data_cMax, notch=0, sym='+', vert=1, whis=1.5, widths=0.3)
bp1 = ax[1].boxplot(data_tMax, notch=0, sym='+', vert=1, whis=1.5, widths=0.3)
bp2 = ax[2].boxplot(data_AUC, notch=0, sym='+', vert=1, whis=1.5, widths=0.3)
plt.setp(bp0['boxes'], color='black')
plt.setp(bp0['whiskers'], color='black', linestyle='solid', linewidth=0.5)
plt.setp(bp0['fliers'], marker='o', color='#e7298a', alpha=0.5)
# plt.setp(bp0['caps'], color='none')
plt.setp(bp1['boxes'], color='black')
plt.setp(bp1['whiskers'], color='black', linestyle='solid', linewidth=0.5)
plt.setp(bp1['fliers'], marker='o', color='#e7298a', alpha=0.5)
# plt.setp(bp1['caps'], color='none')
plt.setp(bp2['boxes'], color='black')
plt.setp(bp2['whiskers'], color='black', linestyle='solid', linewidth=0.5)
plt.setp(bp2['fliers'], marker='o', color='#e7298a', alpha=0.5)
# plt.setp(bp2['caps'], color='none')

ax[0].spines['left']._linewidth = 0.5
ax[1].spines['left']._linewidth = 0.5
ax[2].spines['left']._linewidth = 0.5

# Hide these grid behind plot objects
ax[0].set_axisbelow(True)
ax[1].set_axisbelow(True)
ax[2].set_axisbelow(True)
ax[0].set_ylabel('Conc. ($mg\cdot mL^{-1}$)')
ax[1].set_ylabel('Time ($hr$)')
ax[2].set_ylabel('Bioavail. ($mg\cdot mL^{-1}\cdot hr}$)')
```

```
# Now fill the boxes with desired colors
boxColors = ['#1b9e77','#FF8000']
numBoxes = 6
medians0,medians1,medians2 = range(numBoxes),range(numBoxes),range(numBoxes)
for i in range(numBoxes):
  box0,box1,box2 = bp0['boxes'][i],bp1['boxes'][i],bp2['boxes'][i]
  boxX0,boxX1,boxX2 = [],[],[]
  boxY0,boxY1,boxY2 = [],[],[]
  for j in range(5):
    boxX0.append(box0.get_xdata()[j])
    boxY0.append(box0.get_ydata()[j])
    boxX1.append(box1.get_xdata()[j])
    boxY1.append(box1.get_ydata()[j])
    boxX2.append(box2.get_xdata()[j])
    boxY2.append(box2.get_ydata()[j])
  boxCoords0,boxCoords1,boxCoords2 = zip(boxX0,boxY0),zip(boxX1,boxY1),zip(boxX2,boxY2)
  # Alternate between Dark Khaki and Royal Blue
  k = i % 2
  boxPolygon0,boxPolygon1,boxPolygon2 = Polygon(boxCoords0, facecolor=boxColors[k]),Polygon(boxCoords1, facecolor=boxColors[k]),Polygon(boxCoords2, facecolor=boxColors[k])
  ax[0].add_patch(boxPolygon0)
  ax[1].add_patch(boxPolygon1)
  ax[2].add_patch(boxPolygon2)
  # Now draw the median lines back over what we just filled in
  med0,med1,med2 = bp0['medians'][i],bp1['medians'][i],bp2['medians'][i]
  medianX0,medianX1,medianX2 = [],[],[]
  medianY0,medianY1,medianY2 = [],[],[]
  for j in range(2):
    medianX0.append(med0.get_xdata()[j])
    medianY0.append(med0.get_ydata()[j])
    ax[0].plot(medianX0, medianY0, 'k')
    medians0[i] = medianY0[0]

    medianX1.append(med1.get_xdata()[j])
    medianY1.append(med1.get_ydata()[j])
    ax[1].plot(medianX1, medianY1, 'k')
    medians1[i] = medianY1[0]

    medianX2.append(med2.get_xdata()[j])
    medianY2.append(med2.get_ydata()[j])
    ax[2].plot(medianX2, medianY2, 'k')
    medians2[i] = medianY2[0]
  # Finally, overplot the sample averages, with horizontal alignment
  # in the center of each box
  ax[0].plot([np.average(med0.get_xdata())], [np.average(data_cMax[i][~np.isnan(data_cMax[i])])],
       color=c2, marker='*', markeredgecolor='k')
  ax[1].plot([np.average(med1.get_xdata())], [np.average(data_tMax[i][~np.isnan(data_tMax[i])])],
```

```
        color=c2, marker='*', markeredgecolor='k')
    ax[2].plot([np.average(med2.get_xdata())], [np.average(data_AUC[i][~np.isnan(data_AUC[i])])],
        color=c2, marker='*', markeredgecolor='k')

# Set the axes ranges and axes labels
ax[0].set_xlim(0.5, numBoxes+0.5)
ax[1].set_xlim(0.5, numBoxes+0.5)
ax[2].set_xlim(0.5, numBoxes+0.5)

ax[0].set_ylim(0, 80)
ax[1].set_ylim(.25, 4)
ax[2].set_ylim(20, 500)
xtickNames = plt.setp(ax[2], xticklabels=np.repeat(randomDists, 2))
plt.setp(xtickNames, rotation=45, fontsize=12)

# Due to the Y-axis scale being different across samples, it can be
# hard to compare differences in medians across the samples. Add upper
# X-axis tick labels with the sample medians to aid in comparison
# (just use two decimal places of precision)
pos = np.arange(numBoxes)+1
upperLabels0,upperLabels1,upperLabels2 = [str(np.round(s, 2)) for s in medians0],[str(np.round(s, 2)) for s in medians1],[str(np.round(s, 2)) for s in medians2]
weights = ['bold', 'semibold']
for tick,label in zip(range(numBoxes),ax[0].get_xticklabels()):
   k = tick % 2
   ax[0].text(pos[tick], 80-(80*0.1), upperLabels0[tick],
    horizontalalignment='center', size='medium', weight=weights[k],
    color=boxColors[k])
   ax[1].text(pos[tick], 4-(4*0.1), upperLabels1[tick],
    horizontalalignment='center', size='medium', weight=weights[k],
    color=boxColors[k])
   ax[2].text(pos[tick], 500-(500*0.1), upperLabels2[tick],
    horizontalalignment='center', size='medium', weight=weights[k],
    color=boxColors[k])

# Finally, add a basic legend
plt.figtext(0.75, 0.15,  'Predicted' ,
        backgroundcolor=boxColors[0], color='black', weight='roman',
        size='medium')
plt.figtext(0.75, 0.12, 'Davies, 1998',
        backgroundcolor=boxColors[1], color='white', weight='roman',
        size='medium')
plt.figtext(0.75, 0.09, '*', color=c2, backgroundcolor='white',
        weight='roman', fontsize=15)# size='extralarge')
```

```
plt.figtext(0.765, 0.095, ' Average Value', color='black', weight
    ='roman',
         size='medium')

plt.savefig('IbupPK.png',bbox_inches='tight',dpi=300)
plt.show()
```

./AppendixA/Ibuprofen_gastric_fluid.py

A.2.2 Ibuprofen 800 mg Model

```
# Import required libraries
import json, matplotlib
s = json.load( open("/Users/arjang/.matplotlib/bmh_matplotlibrc.
    json") )
matplotlib.rcParams.update(s)

from IPython.core.pylabtools import figsize
import numpy as np
from scipy.integrate import odeint
from sympy.functions.special.delta_functions import Heaviside
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import time

# Define custom Heaviside function
def u(t):
  t = np.asarray(t)
  is_scalar = False if t.ndim > 0 else True
  t.shape = (1,)*(1-t.ndim) + t.shape
  unit_step = np.arange(t.shape[0])
  lcv = np.arange(t.shape[0])
  for place in lcv:
    if t[place] == 0:
      unit_step[place] = .5
    elif t[place] > 0:
      unit_step[place] = 1
    elif t[place] < 0:
      unit_step[place] = 0
  return (unit_step if not is_scalar else (unit_step[0] if
    unit_step else Heaviside(t)))

# Gastric emptying parameters
p150,p1200 = 0.15, .14
p250, p2200 = .285/(np.pi*15000)/60.0, .16/(np.pi*3900000)/60.0
p350, p3200 = 6.65-1/np.log(60.0), 10.4-1/np.log(60.0)
theta50, theta200 = 1/60.0, 1/60.0
tau50, tau200 = 59.4*60, 60*60.0
s50, s200 = 0.0001/60.0, 0.1/60.0

```

```
# Gastric emptying functions for 50mL and 200mL volumes
def kge50ml(t):
  sum = 0.0
  for k in range(1,26):
    sum += ((-1)**k*np.sin(-theta50/60.0*np.pi*k*(t-tau50)))/k+p150
  return p250*sum**p350+s50
def kge200ml(t):
  sum = 0.0
  for k in range(1,26):
    sum += ((-1)**k*np.sin(-theta200/60.0*np.pi*k*(t-tau200)))/k+p1200
  return p2200*sum**p3200+s200

# Delay parameters
a50,a200 = 2190/(50/60.0),2591/(100.0/60.0)
b50,b200 = 481/10*60.0, 141/10.*60.0
c50,c200 = 27/25., 11/20.
d50,d200 = 531/25. ,138/5.
e50,e200 = 13/200./60.0 ,6/25./60.0
f50,f200 = 41/100./60.0, -59/50./60.0

# Delay functions for 50mL and 200mL volumes
def tlag50ml(t):
  return a50 - b50/(c50+d50*np.exp(-e50*np.mod(t,2/(theta50/60.0))+f50))
def tlag200ml(t):
  return a200 - b200/(c200+d200*np.exp(-e200*np.mod(t,2/(theta200/60.0))+f200))

'''
Set up class for mass balance analysis
Take in time, dose time, plasma elimination rate, disintegration rate,
initial particle radius, resting volume vector, permeation rate vector,
initial concetration vector, and the title
'''
class massBalance:

  def __init__(self, t, to, Kpel, Kdiss, ro, RestVol, PermRates, M0, title):
    self.t = t
    self.to = to
    self.Kpel = Kpel
    self.Kdiss = Kdiss
    self.M0 = M0
    self.title = title
    self.perm0,self.perm1,self.perm2,self.perm3,self.perm4,self.perm5,self.perm6 = PermRates
    self.Grest,self.Irest = RestVol
    self.KIntFactors = np.array([0.77, 0.76, 0.75, 0.74, 0.73, 0.72, 0.7],float)
```

```
  self.KIntShifts = np.array([450*i for i in np.arange
 (0,.3,.3/7)],float)*0.0

  self.vol = M0[0]                # Volume, mL
  self.molwt = 206.3                # Molecular weight, g/mol
  self.MT = M0[8]                 # Initial dose, mg
  self.ro = ro                  # particle radius, um
  self.Vpo = (4.0/3)*np.pi*(self.ro*1.0e-4)**3  # Particle
 volume, cm^3
  self.Pp = 1.1e3                 # Particle density, mg/cm^3
  self.N = (self.MT)/(self.Vpo*self.Pp)     # Number of
 particles
  self.C0 =  .066                 # Intrinsic solubility at 37C,
  g/L
  self.pKa = 4.4                  # Logarithmic acid
 dissociation constant
  self.Ka = 10.0**(-self.pKa)           # Acid dissociation
 constant
  self.krel = 27.63/60/60             # Release rate from solid
 dosage ( sec ^ -1)
  self.peff = 1.413*5.84**-4            # Peff (cm/sec)
  self.kel = 0.0001925              # Plasma elimination rate (
 sec ^ -1)
  self.DHA = 7.5e-6                # diffusivity, cm^2/s
  self.heff = 30e-4                # effective boundary layer, cm

  self.pHS,self.pHInt0,self.pHInt1,self.pHInt2,self.pHInt3,self.
 pHInt4,self.pHInt5,self.pHInt6 = np.array
 ([2,5,5.5,6,6.63,7.41,7.49,7.5],dtype='float')
  self.rS,self.rInt0,self.rInt1,self.rInt2,self.rInt3,self.rInt4
 ,self.rInt5,self.rInt6 = np.ones(8)*self.ro*1.0e-4

# Emptying rate factor as function of particle radius
def kge_mod(self,r):
  return 1.01-.99/(1.0+10.0*np.exp(-r*25+2.47))

# Delayed gastric emptying function for small particules and
 solutions
def Kge(self,t,to):
  self.t = t
  self.to = to
  T = np.mod(t+to,120.0*60.0)
  if (t+to<120.0*60.0): return u(T-tlag200ml(t+to))*kge200ml(t+
 to)
  else: return u(T-tlag50ml(t+to))*kge50ml(t+to)

# Intestinal transit functions
def KInt0(self,t,to):
  return self.KIntFactors[0]*self.Kge(t-self.KIntShifts[0],to)
def KInt1(self,t,to):
  return self.KIntFactors[1]*self.Kge(t-self.KIntShifts[1],to)
def KInt2(self,t,to):
  return self.KIntFactors[2]*self.Kge(t-self.KIntShifts[2],to)
def KInt3(self,t,to):
```

```
    return self.KIntFactors[3]*self.Kge(t-self.KIntShifts[3],to)
  def KInt4(self,t,to):
    return self.KIntFactors[4]*self.Kge(t-self.KIntShifts[4],to)
  def KInt5(self,t,to):
    return self.KIntFactors[5]*self.Kge(t-self.KIntShifts[5],to)
  def Kie(self,t,to):
    return self.KIntFactors[6]*self.Kge(t-self.KIntShifts[6],to)

  # Backflow functions
  def Q1(self,t,to):
    return .15*self.KInt1(t,to)
  def Q2(self,t,to):
    return .15*self.KInt3(t,to)
  def Q3(self,t,to):
    return .15*self.KInt5(t,to)

  # Gastric secretion function
  def Kgs(self,VS,t,to):
    return 2*self.Grest*self.Kge(t,to)/(VS+self.Grest)

  # Take as input the pH and particulate mass
  # Determine solubility, number of particles, individual radius,
  # effective boundary layer and surface area
  def part_props(self,pH, Mp):
    S = self.C0*(1+(10**-self.pKa)/(10**-pH))
    N = Mp/(self.Vpo*self.Pp)
    if N<=1e-9: r = 0
    # ((wt/no. particles) / (density))^1/3 = (mg/(mg/cm^3))^1/3 = cm
    else: r = (3*(Mp/N)/(self.Pp*4.0*np.pi))**(1./3)
    if (r <= 1e-9 or np.isnan(r)):
      r = 0
      heff = 1e22
      A = 0.0
    else:
      heff = 1/(1/r+1/.003)
      A = 4.0*np.pi*r**2*N
    return [S,r,heff,A]

    # Calculate radius based on mass remaining
  def solid_rad(self,Dmass):
    if Dmass<=1e-6: return 0.0
    else: return (Dmass/(self.Pp/1000.0)/(4*np.pi/3.0))**(1/3.)

    # System of equations
  def dM(self,M,t):
    self.M = M
    self.t = t

    VS, VInt0, VInt1, VInt2, VInt3, VInt4, VInt5, VInt6 = M[0:8]
    DSolids, DSolidInt0, DSolidInt1, DSolidInt2, DSolidInt3, DSolidInt4, DSolidInt5, DSolidInt6 = M[8:16]
    DParts, DPartInt0, DPartInt1, DPartInt2, DPartInt3, DPartInt4, DPartInt5, DPartInt6 = M[16:24]
```

```
 DSolns, DSolnInt0, DSolnInt1, DSolnInt2, DSolnInt3, DSolnInt4,
 DSolnInt5, DSolnInt6, DSolnPlasma = M[24:33]

 SS,rS,hS,AS = self.part_props(self.pHS,DParts)
 SInt0,rInt0,hInt0,AInt0 = self.part_props(self.pHInt0,
DPartInt0)
 SInt1,rInt1,hInt1,AInt1 = self.part_props(self.pHInt1,
DPartInt1)
 SInt2,rInt2,hInt2,AInt2 = self.part_props(self.pHInt2,
DPartInt2)
 SInt3,rInt3,hInt3,AInt3 = self.part_props(self.pHInt3,
DPartInt3)
 SInt4,rInt4,hInt4,AInt4 = self.part_props(self.pHInt4,
DPartInt4)
 SInt5,rInt5,hInt5,AInt5 = self.part_props(self.pHInt5,
DPartInt5)
 SInt6,rInt6,hInt6,AInt6 = self.part_props(self.pHInt6,
DPartInt6)

 VS_ = VS*(self.Kgs(VS,t,self.to)-self.Kge(t,self.to))
 VInt0_ = VS*self.Kge(t,self.to) - VInt0*self.KInt0(t,self.to)
 VInt1_ = VInt0*self.KInt0(t,self.to) - VInt1*self.KInt1(t,self
.to) + VInt2*self.Q1(t,self.to)
 VInt2_ = VInt1*self.KInt1(t,self.to) - VInt2*(self.KInt2(t,
self.to) + self.Q1(t,self.to))
 VInt3_ = VInt2*self.KInt2(t,self.to) - VInt3*self.KInt3(t,self
.to) + VInt4*self.Q2(t,self.to)
 VInt4_ = VInt3*self.KInt3(t,self.to) - VInt4*(self.KInt4(t,
self.to)+self.Q2(t,self.to))
 VInt5_ = VInt4*self.KInt4(t,self.to) - VInt5*self.KInt5(t,self
.to) + VInt6*self.Q3(t,self.to)
 VInt6_ = VInt5*self.KInt5(t,self.to) - VInt6*(self.Kie(t,self.
to)+self.Q3(t,self.to))

 # Solid drug undergoing disintegration
 # amount(g) / (1.1 g/cm^3) = cm^3
 # radius of solid = (vol/(4*pi/3))^(1/3) cm
 DSolids_ = (-DSolids*(self.Kge(t,self.to)*self.kge_mod(self.
solid_rad(DSolids)) + self.Kdiss))

 DSolidInt0_  =(DSolids*self.Kge(t,self.to)*self.kge_mod(self.
solid_rad(DSolids)) - DSolidInt0*(self.KInt0(t,self.to) + self
.Kdiss))

 DSolidInt1_ = (DSolidInt0*self.KInt0(t,self.to) - DSolidInt1*(
self.KInt1(t,self.to) + self.Kdiss) + DSolidInt2*self.Q1(t,
self.to))
 DSolidInt2_ = (DSolidInt1*self.KInt1(t,self.to) - DSolidInt2*(
self.KInt2(t,self.to) + self.Q1(t,self.to) + self.Kdiss))

 DSolidInt3_ = (DSolidInt2*self.KInt2(t,self.to) - DSolidInt3*(
self.KInt3(t,self.to) + self.Kdiss) + DSolidInt4*self.Q2(t,
self.to))
```

```
    DSolidInt4_ = (DSolidInt3*self.KInt3(t,self.to) - DSolidInt4*(self.KInt4(t,self.to) + self.Q2(t,self.to) + self.Kdiss))

    DSolidInt5_ = (DSolidInt4*self.KInt4(t,self.to) - DSolidInt5*(self.KInt5(t,self.to) + self.Kdiss) + DSolidInt6*self.Q3(t,self.to))
    DSolidInt6_ = (DSolidInt5*self.KInt5(t,self.to) - DSolidInt6*(self.Kie(t,self.to) + self.Q3(t,self.to) + self.Kdiss))

    # Drug particulates in suspension
    DParts_ = (-DParts*(self.Kge(t,self.to) + AS*self.DHA/hS*(SS-DSolns/VS)) + self.Kdiss*DSolids)

    DPartInt0_ = (DParts*self.Kge(t,self.to) - DPartInt0*(self.KInt0(t,self.to)
    + (AInt0*self.DHA/hInt0*(SInt0-DSolnInt0/VInt0))) + self.Kdiss*DSolidInt0)

    DPartInt1_ = (DPartInt0*self.KInt0(t,self.to) - DPartInt1*(self.KInt1(t,self.to)
    + (AInt1*self.DHA/hInt1*(SInt1-DSolnInt1/VInt1))) + DPartInt2*self.Q1(t,self.to) + self.Kdiss*DSolidInt1)
    DPartInt2_ = (DPartInt1*self.KInt1(t,self.to) - DPartInt2*(self.KInt2(t,self.to)
    + (AInt2*self.DHA/hInt2*(SInt2-DSolnInt2/VInt2))) + self.Kdiss*DSolidInt2)

    DPartInt3_ = (DPartInt2*self.KInt2(t,self.to) - DPartInt3*(self.KInt3(t,self.to)
    + (AInt3*self.DHA/hInt3*(SInt3-DSolnInt3/VInt3))) + DPartInt4*self.Q2(t,self.to) + self.Kdiss*DSolidInt3)
    DPartInt4_ = (DPartInt3*self.KInt3(t,self.to) - DPartInt4*(self.KInt4(t,self.to)
    + (AInt4*self.DHA/hInt4*(SInt4-DSolnInt4/VInt4))) + self.Kdiss*DSolidInt4)

    DPartInt5_ = (DPartInt4*self.KInt4(t,self.to) - DPartInt5*(self.KInt5(t,self.to)
    + (AInt5*self.DHA/hInt5*(SInt5-DSolnInt5/VInt5))) + DPartInt6*self.Q3(t,self.to) + self.Kdiss*DSolidInt5)
    DPartInt6_ = (DPartInt5*self.KInt5(t,self.to) - DPartInt6*(self.Kie(t,self.to)
    + (AInt6*self.DHA/hInt6*(SInt6-DSolnInt6/VInt6))) + self.Kdiss*DSolidInt6)


    # Solution of dissolved drug
    DSolns_ = (-DSolns*self.Kge(t,self.to) + (AS*self.DHA/hS*(SS-DSolns/VS))*DParts)

    DSolnInt0_ = (DSolns*self.Kge(t,self.to) - DSolnInt0*(self.KInt0(t,self.to) + self.perm0)
    + (AInt0*self.DHA/hInt0*(SInt0-DSolnInt0/VInt0))*DPartInt0)
```

```
    DSolnInt1_ = (DSolnInt0*self.KInt0(t,self.to) - DSolnInt1*(
   self.KInt1(t,self.to) + self.perm1) + DSolnInt2*self.Q1(t,self
   .to)
    + (AInt1*self.DHA/hInt0*(SInt1-DSolnInt1/VInt1))*DPartInt1)
    DSolnInt2_ = (DSolnInt1*self.KInt1(t,self.to) - DSolnInt2*(
   self.KInt2(t,self.to) + self.Q1(t,self.to) + self.perm2)
    + (AInt2*self.DHA/hInt1*(SInt2-DSolnInt2/VInt2))*DPartInt2)
    DSolnInt3_ = (DSolnInt2*self.KInt2(t,self.to) - DSolnInt3*(
   self.KInt3(t,self.to) + self.perm3) + DSolnInt4*self.Q2(t,self
   .to)
    + (AInt3*self.DHA/hInt2*(SInt3-DSolnInt3/VInt3))*DPartInt3)
    DSolnInt4_ = (DSolnInt3*self.KInt3(t,self.to) - DSolnInt4*(
   self.KInt4(t,self.to) + self.Q2(t,self.to) + self.perm4)
    + (AInt4*self.DHA/hInt3*(SInt4-DSolnInt4/VInt4))*DPartInt4)
    DSolnInt5_ = (DSolnInt4*self.KInt4(t,self.to) - DSolnInt5*(
   self.KInt5(t,self.to) + self.perm5) + DSolnInt6*self.Q3(t,self
   .to)
    + (AInt5*self.DHA/hInt4*(SInt5-DSolnInt5/VInt5))*DPartInt5)
    DSolnInt6_ = (DSolnInt5*self.KInt5(t,self.to) - DSolnInt6*(
   self.Kie(t,self.to) + self.Q3(t,self.to) + self.perm6)
    + (AInt6*self.DHA/hInt5*(SInt6-DSolnInt6/VInt6))*DPartInt6)
    DSolnPlasma_ = (DSolnInt0*self.perm0 + DSolnInt1*self.perm1 +
   DSolnInt2*self.perm2
      + DSolnInt3*self.perm3 + DSolnInt4*self.perm4 + DSolnInt5*
   self.perm5
      + DSolnInt6*self.perm6 - DSolnPlasma*self.Kpel)

    Mnew = np.array([VS_,VInt0_,VInt1_,VInt2_,VInt3_,VInt4_,VInt5_
   ,VInt6_,
      DSolids_,DSolidInt0_,DSolidInt1_,DSolidInt2_,DSolidInt3_,
   DSolidInt4_,DSolidInt5_,DSolidInt6_,
      DParts_,DPartInt0_,DPartInt1_,DPartInt2_,DPartInt3_,
   DPartInt4_,DPartInt5_,DPartInt6_,
      DSolns_,DSolnInt0_,DSolnInt1_,DSolnInt2_,DSolnInt3_,
   DSolnInt4_,DSolnInt5_,DSolnInt6_,DSolnPlasma_],dtype='float')
    return Mnew

    # Call for solving system of equations
  def solveSys(self,t):
    self.t = t
    # Solve system of equations
    start_time = time.time()
    Mresult = odeint(self.dM,self.M0,t,rtol=1e-6, atol=1e-6)#
   mxstep=500,hmax=50)
    #print("--- %s seconds ---" % np.str(time.time() - start_time)
   )
    return Mresult
```

./AppendixA/Ibuprofen_massBalance.py

A.3 Manometry signal analsysis

A.3.1 Kernel Density Estimation

```
# Import required libraries
import json, matplotlib
s = json.load( open("/Users/arjang/.matplotlib/bmh_matplotlibrc.
    json") )
matplotlib.rcParams.update(s)

%matplotlib inline
from IPython.core.pylabtools import figsize
import numpy as np
from matplotlib import pyplot as plt
import matplotlib.patches as patches
import seaborn as sns
from sklearn.gaussian_process import GaussianProcess
from scipy.signal import detrend
from scipy.signal import medfilt
import time, sys
from joblib import Parallel, delayed
import multiprocessing

# Determine number of cores
num_cores = multiprocessing.cpu_count()

# Loads the data to be analysed.
data1 = np.loadtxt('00540301_01.csv')
data2 = np.loadtxt('00540301_02.csv')
data3 = np.loadtxt('00540301_03.csv')
data4 = np.loadtxt('00540301_04.csv')
data5 = np.loadtxt('00540301_05.csv')
data6 = np.loadtxt('00540301_06.csv')
data7 = np.loadtxt('00540301_07.csv')
data8 = np.loadtxt('00540301_08.csv')
data9 = np.loadtxt('00540301_09.csv')
data10 = np.loadtxt('00540301_10.csv')
data11 = np.loadtxt('00540301_11.csv')
data12 = np.loadtxt('00540301_12.csv')
data13 = np.loadtxt('00540301_13.csv')
mydata = np.array([data1,data2,data3,data4,data5,data6,data7,data8
    ,data9,data10,data11,data12,data13]).T
dataMat1 = np.array([data1,data2,data3,data4]).T
dataMat2 = np.array([data7,data8,data9,data10,data11,data12,data13
    ]).T

# Baseline correction via scipy.signal
mydata_corr = []
[mydata_corr.append(detrend(mydata[:,i])) for i in range(mydata.
    shape[1])]
mydata_corr = np.squeeze(np.transpose(mydata_corr))
```

```
# sampling frequency: 10Hz
samp_freq = 10

# total time (minutes)
N = len(mydata)
T = N/samp_freq/60.0
ts = np.linspace(0,T,num=N)

min_snr, noise_perc = 1,2 # 1.0, 0.3, 2
widths = np.array([2**i for i in range(0,np.mod(2**(samp_freq), samp_freq),1)])
min_thres = 50.0

f = plt.figure(figsize=(8.5,4),dpi=120)
plt.plot(ts,mydata[:,0])
plt.xlabel('Time (min)')
plt.ylabel('Pressure (mmHg)')
plt.title('Duodenal Recording (10Hz)')
plt.xlim(ts[0],250)
plt.ylim(-25,225)
plt.savefig('signal1.png',dpi=300)

'''
Pre-spike activity in phase III portion of signal that is not seen in other regions
'''
f, ax = plt.subplots(nrows=3,ncols=3,dpi=120,sharex=False,sharey=True,figsize=(15.5,8))
plt.subplots_adjust(hspace=0.25,wspace=0.1)
f.suptitle("Pre-spike Activity", fontsize=12)
ax[0,0].set_title('Phase I')
ax[0,1].set_title('Phase II')
ax[0,2].set_title('Phase III')
ax[1,0].set_ylabel('Pressure (mmHg)')
ax[2,1].set_xlabel('Time (min)')

t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-50)),min(range(len(ts)), key=lambda i: abs(ts[i]-90))
ax[0,2].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[0,2].set_xlim(ts[t0],ts[t1])
t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-64)),min(range(len(ts)), key=lambda i: abs(ts[i]-70))
ax[1,2].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[1,2].set_xlim(ts[t0],ts[t1])
t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-64.5)),min(range(len(ts)), key=lambda i: abs(ts[i]-65.5))
ax[2,2].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[2,2].set_xlim(ts[t0],ts[t1])

t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-120)),min(range(len(ts)), key=lambda i: abs(ts[i]-160))
ax[0,1].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[0,1].set_xlim(ts[t0],ts[t1])
```

```
t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-135)),min(range(len(ts)), key=lambda i: abs(ts[i]-141))
ax[1,1].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[1,1].set_xlim(ts[t0],ts[t1])
t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-140)),min(range(len(ts)), key=lambda i: abs(ts[i]-141))
ax[2,1].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
#ax[2,1].set_xlim(ts[0],ts[t1])

t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-90)),min(range(len(ts)), key=lambda i: abs(ts[i]-130))
ax[0,0].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[0,0].set_xlim(ts[t0],ts[t1])
t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-92)),min(range(len(ts)), key=lambda i: abs(ts[i]-99))
ax[1,0].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[1,0].set_xlim(ts[t0],ts[t1])
t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-94)),min(range(len(ts)), key=lambda i: abs(ts[i]-95))
ax[2,0].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[2,0].set_xlim(ts[t0],ts[t1])

ax[0,2].add_patch(patches.Rectangle((64,0),6,180,fill=False,edgecolor="g",linewidth=5,alpha=.5))
ax[1,2].arrow( 64.8, 175, 0, -60, fc="r", ec="r",head_width=0.15, head_length=10)
ax[1,2].arrow( 64.95, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[1,2].arrow( 64.2, 175, 0, -60, fc="r", ec="r",head_width=0.15, head_length=10)
ax[1,2].arrow( 64.3, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[1,2].arrow( 65.60, 175, 0, -60, fc="r", ec="r",head_width=0.15, head_length=10)
ax[1,2].arrow( 65.70, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[1,2].arrow( 67.15, 175, 0, -60, fc="r", ec="r",head_width=0.15, head_length=10)
ax[1,2].arrow( 67.25, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[1,2].arrow( 67.95, 175, 0, -60, fc="r", ec="r",head_width=0.15, head_length=10)
ax[1,2].arrow( 68.10, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[1,2].arrow( 68.7, 175, 0, -60, fc="r", ec="r",head_width=0.15, head_length=10)
ax[1,2].arrow( 68.9, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[2,2].arrow( 65.0, 175, 0, -60, fc="k", ec="k",head_width=0.05, head_length=10)
ax[2,2].arrow( 64.85, 175, 0, -60, fc="r", ec="r",head_width=0.05, head_length=10)

```

```
ax[0,1].add_patch(patches.Rectangle((135,0),6,180,fill=False,
    edgecolor="g",linewidth=5,alpha=.5))
ax[1,1].arrow( 131.75, 175, 0, -60, fc="k", ec="k",head_width
    =0.15, head_length=10)
ax[1,1].arrow( 132.5, 175, 0, -60, fc="k", ec="k",head_width=0.15,
     head_length=10)
ax[1,1].arrow( 135.75, 175, 0, -60, fc="k", ec="k",head_width
    =0.15, head_length=10)
ax[1,1].arrow( 140.3, 175, 0, -60, fc="k", ec="k",head_width=0.15,
     head_length=10)
ax[2,1].arrow( 140.25, 175, 0, -60, fc="k", ec="k",head_width
    =0.06, head_length=10)

ax[0,0].add_patch(patches.Rectangle((92,0),6,180,fill=False,
    edgecolor="g",linewidth=5,alpha=.5))
ax[1,0].arrow( 94.55, 175, 0, -60, fc="k", ec="k",head_width=0.15,
     head_length=10)
ax[2,0].arrow( 94.55, 175, 0, -60, fc="k", ec="k",head_width=0.05,
     head_length=10)
plt.savefig('pre_spike_activity.png', bbox_inches='tight',dpi=300)

'''
Set up class for Gaussian process analysis
Take in data vector, time vector, optionally the nugget, minimum
    lengths for phases III and II,
the theta value and ranges (see documentation)
'''

class GP_analysis:

  def __init__(self, data, ts, nugget=.2, p3_delta=4, p2_delta=2,
    theta0=5e-1, thetaL=1e-3, thetaU=1):
    self.data = data
    self.ts = ts
    # Divide signal into <step> segments
    self.M = self.data.shape[0]
    self.delta = 30 # Fit data in increments of 30 time steps
    self.step = np.int(self.M/self.delta*1.0)

#     self.nugget_ = None
#     if nugget==None: self.nugget = (np.var(data)/data)**2
#     else: self.nugget = nugget
    self.nugget = nugget

    self.X_, self.y_,self.sigma_  = [],[],[]

    self.bar_length = 20
    self.step_range = range(1,self.delta)

    self.p3_delta,self.p2_delta = p3_delta,p2_delta

    self.theta0=theta0
    self.thetaL=thetaL
    self.thetaU=thetaU
```

```
    self.G = None

  # Function to fit Gaussian process in increments
  def fit_data(self):
    start_time = time.time()
    for i in self.step_range:
      step_i = i

      count_data = np.copy(self.data[(step_i-1)*self.step:step_i*
   self.step])
#       count_data[count_data<0] = 0
      n_count_data = len(count_data)

      X, y = ts[(step_i-1)*self.step:step_i*self.step][:,None],
   count_data
#       if len(self.nugget)>1: self.nugget_ = self.nugget[(step_i
   -1)*self.step:step_i*self.step]
#       else: self.nugget_ = self.nugget

      # Estimate probable signal mean for segment using
      # Gaussian process
      self.G = GaussianProcess(corr='squared_exponential',
        theta0=self.theta0, thetaL=self.thetaL, thetaU=self.thetaU
   ,nugget=self.nugget)
      self.G.fit(X, y)

      # Calculate the predicted pressure and the mean squared
   error
      X_pred = np.linspace(X.min(), X.max())[:, None]
      y_pred, MSE = self.G.predict(X_pred, eval_MSE=True)
      sigma = np.sqrt(MSE)

      # Append to prediction matrix
      self.X_.append(X_pred)
      self.y_.append(y_pred)
      self.sigma_.append(sigma)

      # Just a simple progress bar visualization
      percent = float(i-self.step_range[0]) / (self.step_range
   [-1]-self.step_range[0])
      hashes = '#' * int(round(percent * self.bar_length))
      spaces = ' ' * (self.bar_length - len(hashes))
      runtime = int(time.time()-start_time)
      sys.stdout.write("\rPercent: [{0}] {1}% completed in {2}
   seconds".format(hashes + spaces, int(round(percent * 100)),
   runtime))
      sys.stdout.flush()

    self.y_pctiles = np.percentile(np.ravel(self.y_),(25,75))

  def contiguous_regions(self,cond):
    self.cond = cond
```

```
    """Finds contiguous True regions of the boolean array "condition". Returns
    a 2D array where the first column is the start index of the region and the
    second column is the end index."""
    # Find the indicies of changes in "condition"
    d = np.diff(self.cond)
    idx, = d.nonzero()
    # We need to start things after the change in "condition". Therefore,
    # we'll shift the index by 1 to the right.
    idx += 1
    if self.cond[0]:
      # If the start of condition is True prepend a 0
      idx = np.r_[0, idx]
    if self.cond[-1]:
      # If the end of condition is True, append the length of the array
      idx = np.r_[idx, self.cond.size] # Edit
    # Reshape the result into two columns
    idx.shape = (-1,2)
    return idx

  # Return summary of findings
  def summary(self):
    y__ = np.ravel(self.y_)
    X__ = np.ravel(self.X_)
    self.condition = y__ > self.y_pctiles[-1]
    # Print the start and stop indicies of each region where the absolute
    # values of x are below 1, and the min and max of each of these regions
    for start, stop in self.contiguous_regions(self.condition):
      segment = y__[start:stop]
      seg_time = X__[stop]-X__[start]
      if seg_time >= self.p3_delta:
        print("--- Continuous segment length: %.2f min [%.2f : %.2f] ---" % (X__[stop]-X__[start],X__[start], X__[stop]))

  # Plot results
  def plot_phases(self):
#     figsize(12.5, 4)
    for i in range(len(self.step_range)):
      step_i = i+self.step_range[0]

      X__,y__,sigma__ = self.X_[i],self.y_[i],self.sigma_[i]

      plt.plot(ts[(step_i-1)*self.step:step_i*self.step], self.data[(step_i-1)*self.step:step_i*self.step], color="#348ABD", alpha=.25)
      plt.plot(self.X_[i],self.y_[i],'g:')
      plt.plot(X__[y__<self.y_pctiles[0]], y__[y__<self.y_pctiles[0]], 'k.', label=u'PI')
```

```
      plt.plot(X__[(self.y_pctiles[0]<=y__) * (y__<self.y_pctiles[1])], y__[(self.y_pctiles[0]<=y__) * (y__<self.y_pctiles[1])], 'b.', label=u'PII')
      plt.plot(X__[y__>=self.y_pctiles[1]], y__[y__>=self.y_pctiles[1]], 'r.', label=u'PIII')
      plt.fill(np.concatenate([X__, X__[::-1]]),
          np.concatenate([y__ - 1.9600 * sigma__,
                   (y__ + 1.9600 * sigma__)[::-1]]),
          alpha=.3, fc='k', ec='None', label='95% confidence interval')
#       plt.legend(loc='upper left')
  def plot_regions(self):
    y__ = np.ravel(self.y_)
    X__ = np.ravel(self.X_)
    self.condition = y__ > self.y_pctiles[-1]

    plt.plot(self.ts,self.data,alpha=.5)
    plt.xlim((self.ts[0],self.ts[-1]))
    for start, stop in self.contiguous_regions(self.condition):
      seg_time = X__[stop]-X__[start]
      [ts_start,ts_stop] = [np.abs(self.ts - X__[start]).argmin(), np.abs(self.ts - X__[stop]).argmin()]
      if seg_time >= self.p3_delta: plt.plot(self.ts[ts_start:ts_stop],self.data[ts_start:ts_stop],'g',alpha=1)

# Return summary of findings
def summary():
  cond1 = X_classified==1
  # Print the start and stop indicies of each region where the absolute
  # values of x are below 1, and the min and max of each of these regions
  for start, stop in contiguous_regions(cond1):
    segment = X_classified[start:stop]
    seg_time = ts[stop]-ts[start]
    if seg_time >= 1: print("--- Phase I: %.2f min [%.2f : %.2f] ---" % (ts[stop]-ts[start],ts[start], ts[stop]))
  cond2 = X_classified==2
  # Print the start and stop indicies of each region where the absolute
  # values of x are below 1, and the min and max of each of these regions
  for start, stop in contiguous_regions(cond2):
    segment = X_classified[start:stop]
    seg_time = ts[stop]-ts[start]
    if seg_time >= 1: print("--- Phase II: %.2f min [%.2f : %.2f] ---" % (ts[stop]-ts[start],ts[start], ts[stop]))
  cond3 = X_classified==3
  # Print the start and stop indicies of each region where the absolute
  # values of x are below 1, and the min and max of each of these regions
  for start, stop in contiguous_regions(cond3):
    segment = X_classified[start:stop]
```

```
    seg_time = ts[stop]-ts[start]
    if seg_time >= 1: print("--- Phase III: %.2f min [%.2f : %.2f]
    ---" % (ts[stop]-ts[start],ts[start], ts[stop]))

def contiguous_regions(cond):
  """Finds contiguous True regions of the boolean array "condition
   ". Returns
  a 2D array where the first column is the start index of the
   region and the
  second column is the end index."""
  # Find the indicies of changes in "condition"
  d = np.diff(cond)
  idx, = d.nonzero()
  # We need to start things after the change in "condition".
   Therefore,
  # we'll shift the index by 1 to the right.
  idx += 1
  if cond[0]:
    # If the start of condition is True prepend a 0
    idx = np.r_[0, idx]
  if cond[-1]:
    # If the end of condition is True, append the length of the
   array
    idx = np.r_[idx, cond.size] # Edit
  # Reshape the result into two columns
  idx.shape = (-1,2)
  return idx

'''
Use the test data from MMS (loaded above)
'''


# Signal from first channel
indeces = [0]
X = np.ndarray((len(indeces),),dtype=np.object)
for ind in range(len(indeces)):
  X[ind] = GP_analysis(mydata_corr[:,indeces[ind]],ts=ts)

start_time = time.time()
for i in range(len(indeces)):
  print("\n--- Channel %d ---" % indeces[i])
  X[i].fit_data()
print("--- Parallelized execution completed: %s seconds ---" %
    float(time.time() - float(start_time)))

[X[i].summary() for i in indeces]

'''
Print results
'''

figsize(7.5,3)
fig = plt.figure(dpi=120)
```

```
plt.plot(ts,X[0].data,alpha=.95,label='Signal')
plt.xlim(ts[0],ts[-1])
plt.title('Original Signal')
plt.ylabel('Pressure (mmHg)')
plt.xlabel('Time (min)')
plt.savefig('gaus_process_fig0', bbox_inches='tight',dpi=300)
plt.show()

y__ = np.ravel(X[0].y_)
X__ = np.ravel(X[0].X_)
sigma__ = np.ravel(X[0].sigma_)
condition = y__ > X[0].y_pctiles[-1]

figsize(7.5,3)
fig = plt.figure(dpi=120)
plt.plot(ts,X[0].data,alpha=.25,label='Signal')
plt.xlim(ts[0],ts[-1])
plt.plot(X__[y__<X[0].y_pctiles[0]], y__[y__<X[0].y_pctiles[0]], '
    k.', label=u'Low')
plt.plot(X__[(X[0].y_pctiles[0]<=y__) * (y__<X[0].y_pctiles[1])],
    y__[(X[0].y_pctiles[0]<=y__) * (y__<X[0].y_pctiles[1])], 'b.',
     label=u'Medium')
plt.plot(X__[y__>=X[0].y_pctiles[1]], y__[y__>=X[0].y_pctiles[1]],
     'r.', label=u'High')
plt.fill(np.concatenate([X__, X__[::-1]]),
     np.concatenate([y__ - 1.9600 * sigma__,
                (y__ + 1.9600 * sigma__)[::-1]]),
     alpha=.3, fc='k', ec='None', label='95% CI')
plt.title('Predicting Average Pressures')
plt.ylabel('Pressure (mmHg)')
plt.xlabel('Time (min)')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.savefig('gaus_process_fig1', bbox_inches='tight',dpi=300)
plt.show()


y__ = np.ravel(X[0].y_)
X__ = np.ravel(X[0].X_)
X_labeled = np.empty_like(X__)
X_labeled[y__<X[0].y_pctiles[0]] = 1
X_labeled[(X[0].y_pctiles[0]<=y__) * (y__<X[0].y_pctiles[1])] = 2
X_labeled[y__>=X[0].y_pctiles[1]] = 3
y_filt = medfilt(medfilt(X_labeled,kernel_size=21),kernel_size=21)

X_class = np.zeros(np.shape(ts))

plt.plot(ts,mydata_corr[:,0],':k',alpha=.5)
plt.xlim((ts[0],ts[-1]))

condition = y_filt > 2 # y__ > X[0].y_pctiles[-1]
ind = 1
tmp_reg = contiguous_regions(condition)
for start, stop in contiguous_regions(condition):
  seg_time = X__[stop]-X__[start]
```

```
  [ts_start,ts_stop] = [np.abs(ts - X__[start]).argmin(),np.abs(ts
    - X__[stop]).argmin()]

  X_class[ts_start:ts_stop] = 3

  if seg_time >= 5: plt.plot(ts[ts_start:ts_stop],mydata_corr[
   ts_start:ts_stop,0],'r',alpha=1)

  if ind<len(tmp_reg):
    if (X__[tmp_reg[ind][0]] - X__[stop]<= 10):
      [ts_start,ts_stop] = [np.abs(ts - X__[stop]).argmin(),np.abs
   (ts - X__[tmp_reg[ind][1]]).argmin()]
      X_class[ts_start:ts_stop] = 3
      plt.plot(ts[ts_start:ts_stop],mydata_corr[ts_start:ts_stop
   ,0],'r',alpha=1)
  ind += 1
condition = (y_filt <= 2) * (y_filt > 1)
# (y__ <= X[0].y_pctiles[-1])*(y__ > X[0].y_pctiles[0])
ind = 1
tmp_reg = contiguous_regions(condition)
for start, stop in contiguous_regions(condition):
  if stop<len(X__):
    seg_time = X__[stop]-X__[start]
    [ts_start,ts_stop] = [np.abs(ts - X__[start]).argmin(),np.abs(
   ts - X__[stop]).argmin()]
    X_class[ts_start:ts_stop] = 2
    if seg_time >= 2:
#       plt.plot(ts[ts_start:ts_stop],mydata_corr[ts_start:ts_stop
   ,0],'b',alpha=1)
      if ind<len(tmp_reg):
        if (X__[tmp_reg[ind][0]] - X__[stop]<= 5):
          [ts_start,ts_stop] = [np.abs(ts - X__[stop]).argmin(),np
   .abs(ts - X__[tmp_reg[ind][1]]).argmin()]
          X_class[ts_start:ts_stop] = 2
          plt.plot(ts[ts_start:ts_stop],mydata_corr[ts_start:
   ts_stop,0],'b',alpha=.5)
  ind += 1
condition =  (y_filt == 1)

ind = 1
tmp_reg = contiguous_regions(condition)
for start, stop in contiguous_regions(condition):
  if stop<len(X__):
    seg_time = X__[stop]-X__[start]
    [ts_start,ts_stop] = [np.abs(ts - X__[start]).argmin(),np.abs(
   ts - X__[stop]).argmin()]
    X_class[ts_start:ts_stop] = 1
    if seg_time >= 1:
      if ind<len(tmp_reg):
        if (X__[tmp_reg[ind][0]] - X__[stop]<= 30):
          [ts_start,ts_stop] = [np.abs(ts - X__[stop]).argmin(),np
   .abs(ts - X__[tmp_reg[ind][1]]).argmin()]
          X_class[ts_start:ts_stop] = 1
```

```
          plt.plot(ts[ts_start:ts_stop],mydata_corr[ts_start:ts_stop,0],'g',alpha=.5)
  ind += 1

'''
Make sure the transitions are appropriate, else it is a false positive detection

1 -> 2: -1
2 -> 3: -1
3 -> 1: 2

1 -> 3: -2
2 -> 1: 1
3 -> 2: 1

'''

X_classified = np.copy(X_class)
for i in range(len(X_class)-1):
    if (X_classified[i]-X_classified[i+1]==-2): X_classified[i+1] = X_classified[i]+1
    elif (X_classified[i]-X_classified[i+1]==1): X_classified[i+1] = X_classified[i]

figsize(7.5,7)
sns.set_style("whitegrid")
f, (ax2,ax0,ax1) = plt.subplots(nrows=3,ncols=1,dpi=120,sharex=True,sharey=False)
ax2.plot(ts,mydata_corr[:,0],':k',alpha=.5,label=u'Signal')
ax2.set_xlim((ts[0],ts[-1]))
ax2.set_ylim((-5,250))
ax2.plot(ts[X_classified==1], mydata_corr[X_classified==1,0], '.g', label=u'PI')
ax2.plot(ts[X_classified==2], mydata_corr[X_classified==2,0], '.b', label=u'PII')
ax2.plot(ts[X_classified==3], mydata_corr[X_classified==3,0], '.r', label=u'PIII')
ax2.set_title('Phase Classification')
ax2.set_ylabel('Pressure (mmHg)')
ax2.legend(loc='upper center', bbox_to_anchor=(0.5, 1.05),
          ncol=4, fancybox=True, shadow=True)
ax2.text(10, 190, '58.60 min', style='normal',
        bbox={'facecolor':'blue', 'alpha':0.25, 'pad':10})
ax2.text(60, 190, '24.80 min', style='normal',
        bbox={'facecolor':'red', 'alpha':0.25, 'pad':10})
ax2.text(110, 190, '71.33 min', style='normal',
        bbox={'facecolor':'green', 'alpha':0.25, 'pad':10})
ax2.text(165, 190, '55.42 min', style='normal',
        bbox={'facecolor':'blue', 'alpha':0.25, 'pad':10})
ax2.text(210, 190, '27.54 min', style='normal',
        bbox={'facecolor':'red', 'alpha':0.25, 'pad':10})

ax0.plot(ts,.009*X_classified/np.max(X_classified))
```

```
ax0.set_ylim(0,.01)
ax0.set_ylabel('Phase')
majorticks = np.array([1,2,3],dtype='float')/3*.009
ax0.set_yticks(majorticks)
ax0.set_yticklabels(['I','II','III'])
ax0.set_xlim((ts[0],ts[-1]))
sns.rugplot(np.array(data), c=c1, ax=ax0)

# Set up the plots
c1, c2 = sns.color_palette("husl", 3)[:2]
# Plot the summed basis functions
summed_kde = np.sum(kernels, axis=0)
# ax1.plot(xx, summed_kde, c=c1)
sns.kdeplot(np.array(data), bw=bandwidth, linewidth=1, shade=True,
    color=c1, label=r'Peak Density', ax=ax1)
sns.rugplot(np.array(data), c=c1, ax=ax1)
ax1.set_yticks([])
ax1.set_ylabel('Density')
ax1.set_xlabel('Time (min)')
plt.savefig('gaus_process_fig2', bbox_inches='tight',dpi=300)
```

./AppendixA/gauss_proc.py

A.3.2 Gaussian Process Regression

```
# Import required libraries
import json, matplotlib
s = json.load( open("/Users/arjang/.matplotlib/bmh_matplotlibrc.
   json") )
matplotlib.rcParams.update(s)

%matplotlib inline
from IPython.core.pylabtools import figsize
import numpy as np
from matplotlib import pyplot as plt
import matplotlib.patches as patches
import seaborn as sns
from sklearn.gaussian_process import GaussianProcess
from scipy.signal import detrend
from scipy.signal import medfilt
import time, sys
from joblib import Parallel, delayed
import multiprocessing

# Determine number of cores
num_cores = multiprocessing.cpu_count()

# Loads the data to be analysed.
data1 = np.loadtxt('00540301_01.csv')
data2 = np.loadtxt('00540301_02.csv')
```

```
data3 = np.loadtxt('00540301_03.csv')
data4 = np.loadtxt('00540301_04.csv')
data5 = np.loadtxt('00540301_05.csv')
data6 = np.loadtxt('00540301_06.csv')
data7 = np.loadtxt('00540301_07.csv')
data8 = np.loadtxt('00540301_08.csv')
data9 = np.loadtxt('00540301_09.csv')
data10 = np.loadtxt('00540301_10.csv')
data11 = np.loadtxt('00540301_11.csv')
data12 = np.loadtxt('00540301_12.csv')
data13 = np.loadtxt('00540301_13.csv')
mydata = np.array([data1,data2,data3,data4,data5,data6,data7,data8
    ,data9,data10,data11,data12,data13]).T
dataMat1 = np.array([data1,data2,data3,data4]).T
dataMat2 = np.array([data7,data8,data9,data10,data11,data12,data13
    ]).T

# Baseline correction via scipy.signal
mydata_corr = []
[mydata_corr.append(detrend(mydata[:,i])) for i in range(mydata.
    shape[1])]
mydata_corr = np.squeeze(np.transpose(mydata_corr))

# sampling frequency: 10Hz
samp_freq = 10

# total time (minutes)
N = len(mydata)
T = N/samp_freq/60.0
ts = np.linspace(0,T,num=N)

min_snr, noise_perc = 1,2 # 1.0, 0.3, 2
widths = np.array([2**i for i in range(0,np.mod(2**(samp_freq),
    samp_freq),1)])
min_thres = 50.0

f = plt.figure(figsize=(8.5,4),dpi=120)
plt.plot(ts,mydata[:,0])
plt.xlabel('Time (min)')
plt.ylabel('Pressure (mmHg)')
plt.title('Duodenal Recording (10Hz)')
plt.xlim(ts[0],250)
plt.ylim(-25,225)
plt.savefig('signal1.png',dpi=300)

'''
Pre-spike activity in phase III portion of signal that is not seen
     in other regions
'''
f, ax = plt.subplots(nrows=3,ncols=3,dpi=120,sharex=False,sharey=
    True,figsize=(15.5,8))
plt.subplots_adjust(hspace=0.25,wspace=0.1)
f.suptitle("Pre-spike Activity", fontsize=12)
ax[0,0].set_title('Phase I')
```

```
ax[0,1].set_title('Phase II')
ax[0,2].set_title('Phase III')
ax[1,0].set_ylabel('Pressure (mmHg)')
ax[2,1].set_xlabel('Time (min)')

t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-50)),min(
    range(len(ts)), key=lambda i: abs(ts[i]-90))
ax[0,2].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[0,2].set_xlim(ts[t0],ts[t1])
t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-64)),min(
    range(len(ts)), key=lambda i: abs(ts[i]-70))
ax[1,2].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[1,2].set_xlim(ts[t0],ts[t1])
t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-64.5)),min(
    range(len(ts)), key=lambda i: abs(ts[i]-65.5))
ax[2,2].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[2,2].set_xlim(ts[t0],ts[t1])

t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-120)),min(
    range(len(ts)), key=lambda i: abs(ts[i]-160))
ax[0,1].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[0,1].set_xlim(ts[t0],ts[t1])
t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-135)),min(
    range(len(ts)), key=lambda i: abs(ts[i]-141))
ax[1,1].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[1,1].set_xlim(ts[t0],ts[t1])
t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-140)),min(
    range(len(ts)), key=lambda i: abs(ts[i]-141))
ax[2,1].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
#ax[2,1].set_xlim(ts[0],ts[t1])

t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-90)),min(
    range(len(ts)), key=lambda i: abs(ts[i]-130))
ax[0,0].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[0,0].set_xlim(ts[t0],ts[t1])
t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-92)),min(
    range(len(ts)), key=lambda i: abs(ts[i]-99))
ax[1,0].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[1,0].set_xlim(ts[t0],ts[t1])
t0, t1 = min(range(len(ts)), key=lambda i: abs(ts[i]-94)),min(
    range(len(ts)), key=lambda i: abs(ts[i]-95))
ax[2,0].plot(ts[t0:t1],mydata_corr[t0:t1,0],alpha=.75)
ax[2,0].set_xlim(ts[t0],ts[t1])

ax[0,2].add_patch(patches.Rectangle((64,0),6,180,fill=False,
    edgecolor="g",linewidth=5,alpha=.5))
ax[1,2].arrow( 64.8, 175, 0, -60, fc="r", ec="r",head_width=0.15,
    head_length=10)
ax[1,2].arrow( 64.95, 175, 0, -60, fc="k", ec="k",head_width=0.15,
     head_length=10)
ax[1,2].arrow( 64.2, 175, 0, -60, fc="r", ec="r",head_width=0.15,
    head_length=10)
ax[1,2].arrow( 64.3, 175, 0, -60, fc="k", ec="k",head_width=0.15,
    head_length=10)
```

```
ax[1,2].arrow( 65.60, 175, 0, -60, fc="r", ec="r",head_width=0.15, head_length=10)
ax[1,2].arrow( 65.70, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[1,2].arrow( 67.15, 175, 0, -60, fc="r", ec="r",head_width=0.15, head_length=10)
ax[1,2].arrow( 67.25, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[1,2].arrow( 67.95, 175, 0, -60, fc="r", ec="r",head_width=0.15, head_length=10)
ax[1,2].arrow( 68.10, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[1,2].arrow( 68.7, 175, 0, -60, fc="r", ec="r",head_width=0.15, head_length=10)
ax[1,2].arrow( 68.9, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[2,2].arrow( 65.0, 175, 0, -60, fc="k", ec="k",head_width=0.05, head_length=10)
ax[2,2].arrow( 64.85, 175, 0, -60, fc="r", ec="r",head_width=0.05, head_length=10)

ax[0,1].add_patch(patches.Rectangle((135,0),6,180,fill=False, edgecolor="g",linewidth=5,alpha=.5))
ax[1,1].arrow( 131.75, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[1,1].arrow( 132.5, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[1,1].arrow( 135.75, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[1,1].arrow( 140.3, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[2,1].arrow( 140.25, 175, 0, -60, fc="k", ec="k",head_width=0.06, head_length=10)

ax[0,0].add_patch(patches.Rectangle((92,0),6,180,fill=False, edgecolor="g",linewidth=5,alpha=.5))
ax[1,0].arrow( 94.55, 175, 0, -60, fc="k", ec="k",head_width=0.15, head_length=10)
ax[2,0].arrow( 94.55, 175, 0, -60, fc="k", ec="k",head_width=0.05, head_length=10)
plt.savefig('pre_spike_activity.png', bbox_inches='tight',dpi=300)

'''
Set up class for Gaussian process analysis
Take in data vector, time vector, optionally the nugget, minimum lengths for phases III and II,
the theta value and ranges (see documentation)
'''

class GP_analysis:

  def __init__(self, data, ts, nugget=.2, p3_delta=4, p2_delta=2, theta0=5e-1, thetaL=1e-3, thetaU=1):
    self.data = data
```

```
    self.ts = ts
    # Divide signal into <step> segments
    self.M = self.data.shape[0]
    self.delta = 30 # Fit data in increments of 30 time steps
    self.step = np.int(self.M/self.delta*1.0)

#     self.nugget_ = None
#     if nugget==None: self.nugget = (np.var(data)/data)**2
#     else: self.nugget = nugget
    self.nugget = nugget

    self.X_, self.y_,self.sigma_  = [],[],[]

    self.bar_length = 20
    self.step_range = range(1,self.delta)

    self.p3_delta,self.p2_delta = p3_delta,p2_delta

    self.theta0=theta0
    self.thetaL=thetaL
    self.thetaU=thetaU

    self.G = None

  # Function to fit Gaussian process in increments
  def fit_data(self):
    start_time = time.time()
    for i in self.step_range:
      step_i = i

      count_data = np.copy(self.data[(step_i-1)*self.step:step_i*
   self.step])
#       count_data[count_data<0] = 0
      n_count_data = len(count_data)

      X, y = ts[(step_i-1)*self.step:step_i*self.step][:,None],
   count_data
#       if len(self.nugget)>1: self.nugget_ = self.nugget[(step_i
   -1)*self.step:step_i*self.step]
#       else: self.nugget_ = self.nugget

      # Estimate probable signal mean for segment using
      # Gaussian process
      self.G = GaussianProcess(corr='squared_exponential',
        theta0=self.theta0, thetaL=self.thetaL, thetaU=self.thetaU
   ,nugget=self.nugget)
      self.G.fit(X, y)

      # Calculate the predicted pressure and the mean squared
   error
      X_pred = np.linspace(X.min(), X.max())[:, None]
      y_pred, MSE = self.G.predict(X_pred, eval_MSE=True)
      sigma = np.sqrt(MSE)

```

```
      # Append to prediction matrix
      self.X_.append(X_pred)
      self.y_.append(y_pred)
      self.sigma_.append(sigma)

      # Just a simple progress bar visualization
      percent = float(i-self.step_range[0]) / (self.step_range[-1]-self.step_range[0])
      hashes = '#' * int(round(percent * self.bar_length))
      spaces = ' ' * (self.bar_length - len(hashes))
      runtime = int(time.time()-start_time)
      sys.stdout.write("\rPercent: [{0}] {1}% completed in {2} seconds".format(hashes + spaces, int(round(percent * 100)), runtime))
      sys.stdout.flush()

    self.y_pctiles = np.percentile(np.ravel(self.y_),(25,75))

  def contiguous_regions(self,cond):
    self.cond = cond
    """Finds contiguous True regions of the boolean array "condition". Returns
    a 2D array where the first column is the start index of the region and the
    second column is the end index."""
    # Find the indicies of changes in "condition"
    d = np.diff(self.cond)
    idx, = d.nonzero()
    # We need to start things after the change in "condition". Therefore,
    # we'll shift the index by 1 to the right.
    idx += 1
    if self.cond[0]:
      # If the start of condition is True prepend a 0
      idx = np.r_[0, idx]
    if self.cond[-1]:
      # If the end of condition is True, append the length of the array
      idx = np.r_[idx, self.cond.size] # Edit
    # Reshape the result into two columns
    idx.shape = (-1,2)
    return idx

  # Return summary of findings
  def summary(self):
    y__ = np.ravel(self.y_)
    X__ = np.ravel(self.X_)
    self.condition = y__ > self.y_pctiles[-1]
    # Print the start and stop indicies of each region where the absolute
    # values of x are below 1, and the min and max of each of these regions
    for start, stop in self.contiguous_regions(self.condition):
      segment = y__[start:stop]
```

```
   seg_time = X__[stop]-X__[start]
   if seg_time >= self.p3_delta:
    print("--- Continuous segment length: %.2f min [%.2f : %.2
  f] ---" % (X__[stop]-X__[start],X__[start], X__[stop]))

 # Plot results
 def plot_phases(self):
#  figsize(12.5, 4)
  for i in range(len(self.step_range)):
   step_i = i+self.step_range[0]

   X__,y__,sigma__ = self.X_[i],self.y_[i],self.sigma_[i]

   plt.plot(ts[(step_i-1)*self.step:step_i*self.step], self.
  data[(step_i-1)*self.step:step_i*self.step], color="#348ABD",
  alpha=.25)
   plt.plot(self.X_[i],self.y_[i],'g:')
   plt.plot(X__[y__<self.y_pctiles[0]], y__[y__<self.y_pctiles
  [0]], 'k.', label=u'PI')
   plt.plot(X__[(self.y_pctiles[0]<=y__) * (y__<self.y_pctiles
  [1])], y__[(self.y_pctiles[0]<=y__) * (y__<self.y_pctiles[1])
  ], 'b.', label=u'PII')
   plt.plot(X__[y__>=self.y_pctiles[1]], y__[y__>=self.
  y_pctiles[1]], 'r.', label=u'PIII')
   plt.fill(np.concatenate([X__, X__[::-1]]),
     np.concatenate([y__ - 1.9600 * sigma__,
          (y__ + 1.9600 * sigma__)[::-1]]),
     alpha=.3, fc='k', ec='None', label='95% confidence
  interval')
#   plt.legend(loc='upper left')
 def plot_regions(self):
  y__ = np.ravel(self.y_)
  X__ = np.ravel(self.X_)
  self.condition = y__ > self.y_pctiles[-1]

  plt.plot(self.ts,self.data,alpha=.5)
  plt.xlim((self.ts[0],self.ts[-1]))
  for start, stop in self.contiguous_regions(self.condition):
   seg_time = X__[stop]-X__[start]
   [ts_start,ts_stop] = [np.abs(self.ts - X__[start]).argmin(),
  np.abs(self.ts - X__[stop]).argmin()]
   if seg_time >= self.p3_delta: plt.plot(self.ts[ts_start:
  ts_stop],self.data[ts_start:ts_stop],'g',alpha=1)

# Return summary of findings
def summary():
 cond1 = X_classified==1
 # Print the start and stop indicies of each region where the
  absolute
 # values of x are below 1, and the min and max of each of these
  regions
 for start, stop in contiguous_regions(cond1):
  segment = X_classified[start:stop]
  seg_time = ts[stop]-ts[start]
```

```
        if seg_time >= 1: print("--- Phase I: %.2f min [%.2f : %.2f] ---" % (ts[stop]-ts[start],ts[start], ts[stop]))
    cond2 = X_classified==2
    # Print the start and stop indicies of each region where the absolute
    # values of x are below 1, and the min and max of each of these regions
    for start, stop in contiguous_regions(cond2):
        segment = X_classified[start:stop]
        seg_time = ts[stop]-ts[start]
        if seg_time >= 1: print("--- Phase II: %.2f min [%.2f : %.2f] ---" % (ts[stop]-ts[start],ts[start], ts[stop]))
    cond3 = X_classified==3
    # Print the start and stop indicies of each region where the absolute
    # values of x are below 1, and the min and max of each of these regions
    for start, stop in contiguous_regions(cond3):
        segment = X_classified[start:stop]
        seg_time = ts[stop]-ts[start]
        if seg_time >= 1: print("--- Phase III: %.2f min [%.2f : %.2f] ---" % (ts[stop]-ts[start],ts[start], ts[stop]))

def contiguous_regions(cond):
    """Finds contiguous True regions of the boolean array "condition". Returns
    a 2D array where the first column is the start index of the region and the
    second column is the end index."""
    # Find the indicies of changes in "condition"
    d = np.diff(cond)
    idx, = d.nonzero()
    # We need to start things after the change in "condition". Therefore,
    # we'll shift the index by 1 to the right.
    idx += 1
    if cond[0]:
        # If the start of condition is True prepend a 0
        idx = np.r_[0, idx]
    if cond[-1]:
        # If the end of condition is True, append the length of the array
        idx = np.r_[idx, cond.size] # Edit
    # Reshape the result into two columns
    idx.shape = (-1,2)
    return idx

'''
Use the test data from MMS (loaded above)
'''


# Signal from first channel
indeces = [0]
```

```python
X = np.ndarray((len(indeces),),dtype=np.object)
for ind in range(len(indeces)):
  X[ind] = GP_analysis(mydata_corr[:,indeces[ind]],ts=ts)

start_time = time.time()
for i in range(len(indeces)):
  print("\n--- Channel %d ---" % indeces[i])
  X[i].fit_data()
print("--- Parallelized execution completed: %s seconds ---" % float(time.time() - float(start_time)))

[X[i].summary() for i in indeces]

'''
Print results
'''

figsize(7.5,3)
fig = plt.figure(dpi=120)
plt.plot(ts,X[0].data,alpha=.95,label='Signal')
plt.xlim(ts[0],ts[-1])
plt.title('Original Signal')
plt.ylabel('Pressure (mmHg)')
plt.xlabel('Time (min)')
plt.savefig('gaus_process_fig0', bbox_inches='tight',dpi=300)
plt.show()

y__ = np.ravel(X[0].y_)
X__ = np.ravel(X[0].X_)
sigma__ = np.ravel(X[0].sigma_)
condition = y__ > X[0].y_pctiles[-1]

figsize(7.5,3)
fig = plt.figure(dpi=120)
plt.plot(ts,X[0].data,alpha=.25,label='Signal')
plt.xlim(ts[0],ts[-1])
plt.plot(X__[y__<X[0].y_pctiles[0]], y__[y__<X[0].y_pctiles[0]], 'k.', label=u'Low')
plt.plot(X__[(X[0].y_pctiles[0]<=y__) * (y__<X[0].y_pctiles[1])], y__[(X[0].y_pctiles[0]<=y__) * (y__<X[0].y_pctiles[1])], 'b.', label=u'Medium')
plt.plot(X__[y__>=X[0].y_pctiles[1]], y__[y__>=X[0].y_pctiles[1]], 'r.', label=u'High')
plt.fill(np.concatenate([X__, X__[::-1]]),
     np.concatenate([y__ - 1.9600 * sigma__,
              (y__ + 1.9600 * sigma__)[::-1]]),
     alpha=.3, fc='k', ec='None', label='95% CI')
plt.title('Predicting Average Pressures')
plt.ylabel('Pressure (mmHg)')
plt.xlabel('Time (min)')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.savefig('gaus_process_fig1', bbox_inches='tight',dpi=300)
plt.show()
```

```
y__ = np.ravel(X[0].y_)
X__ = np.ravel(X[0].X_)
X_labeled = np.empty_like(X__)
X_labeled[y__<X[0].y_pctiles[0]] = 1
X_labeled[(X[0].y_pctiles[0]<=y__) * (y__<X[0].y_pctiles[1])] = 2
X_labeled[y__>=X[0].y_pctiles[1]] = 3
y_filt = medfilt(medfilt(X_labeled,kernel_size=21),kernel_size=21)

X_class = np.zeros(np.shape(ts))

plt.plot(ts,mydata_corr[:,0],':k',alpha=.5)
plt.xlim((ts[0],ts[-1]))

condition = y_filt > 2 # y__ > X[0].y_pctiles[-1]
ind = 1
tmp_reg = contiguous_regions(condition)
for start, stop in contiguous_regions(condition):
  seg_time = X__[stop]-X__[start]
  [ts_start,ts_stop] = [np.abs(ts - X__[start]).argmin(),np.abs(ts
     - X__[stop]).argmin()]

  X_class[ts_start:ts_stop] = 3

  if seg_time >= 5: plt.plot(ts[ts_start:ts_stop],mydata_corr[
   ts_start:ts_stop,0],'r',alpha=1)

  if ind<len(tmp_reg):
    if (X__[tmp_reg[ind][0]] - X__[stop]<= 10):
      [ts_start,ts_stop] = [np.abs(ts - X__[stop]).argmin(),np.abs
   (ts - X__[tmp_reg[ind][1]]).argmin()]
      X_class[ts_start:ts_stop] = 3
      plt.plot(ts[ts_start:ts_stop],mydata_corr[ts_start:ts_stop
   ,0],'r',alpha=1)
  ind += 1
condition = (y_filt <= 2) * (y_filt > 1)
# (y__ <= X[0].y_pctiles[-1])*(y__ > X[0].y_pctiles[0])
ind = 1
tmp_reg = contiguous_regions(condition)
for start, stop in contiguous_regions(condition):
  if stop<len(X__):
    seg_time = X__[stop]-X__[start]
    [ts_start,ts_stop] = [np.abs(ts - X__[start]).argmin(),np.abs(
   ts - X__[stop]).argmin()]
    X_class[ts_start:ts_stop] = 2
    if seg_time >= 2:
#       plt.plot(ts[ts_start:ts_stop],mydata_corr[ts_start:ts_stop
   ,0],'b',alpha=1)
      if ind<len(tmp_reg):
        if (X__[tmp_reg[ind][0]] - X__[stop]<= 5):
          [ts_start,ts_stop] = [np.abs(ts - X__[stop]).argmin(),np
   .abs(ts - X__[tmp_reg[ind][1]]).argmin()]
          X_class[ts_start:ts_stop] = 2
```

```
            plt.plot(ts[ts_start:ts_stop],mydata_corr[ts_start:
    ts_stop,0],'b',alpha=.5)
    ind += 1
condition =  (y_filt == 1)

ind = 1
tmp_reg = contiguous_regions(condition)
for start, stop in contiguous_regions(condition):
    if stop<len(X__):
        seg_time = X__[stop]-X__[start]
        [ts_start,ts_stop] = [np.abs(ts - X__[start]).argmin(),np.abs(
    ts - X__[stop]).argmin()]
        X_class[ts_start:ts_stop] = 1
        if seg_time >= 1:
            if ind<len(tmp_reg):
                if (X__[tmp_reg[ind][0]] - X__[stop]<= 30):
                    [ts_start,ts_stop] = [np.abs(ts - X__[stop]).argmin(),np
    .abs(ts - X__[tmp_reg[ind][1]]).argmin()]
                    X_class[ts_start:ts_stop] = 1
                    plt.plot(ts[ts_start:ts_stop],mydata_corr[ts_start:
    ts_stop,0],'g',alpha=.5)
    ind += 1

'''
Make sure the transitions are appropriate, else it is a false
    positive detection

1 -> 2: -1
2 -> 3: -1
3 -> 1: 2

1 -> 3: -2
2 -> 1: 1
3 -> 2: 1

'''

X_classified = np.copy(X_class)
for i in range(len(X_class)-1):
        if (X_classified[i]-X_classified[i+1]==-2): X_classified[i+1]
    = X_classified[i]+1
        elif (X_classified[i]-X_classified[i+1]==1): X_classified[i+1]
        = X_classified[i]

figsize(7.5,7)
sns.set_style("whitegrid")
f, (ax2,ax0,ax1) = plt.subplots(nrows=3,ncols=1,dpi=120,sharex=
    True,sharey=False)
ax2.plot(ts,mydata_corr[:,0],':k',alpha=.5,label=u'Signal')
ax2.set_xlim((ts[0],ts[-1]))
ax2.set_ylim((-5,250))
ax2.plot(ts[X_classified==1], mydata_corr[X_classified==1,0], '.g
    ', label=u'PI')
```

```
ax2.plot(ts[X_classified==2], mydata_corr[X_classified==2,0], '.b
    ', label=u'PII')
ax2.plot(ts[X_classified==3], mydata_corr[X_classified==3,0], '.r
    ', label=u'PIII')
ax2.set_title('Phase Classification')
ax2.set_ylabel('Pressure (mmHg)')
ax2.legend(loc='upper center', bbox_to_anchor=(0.5, 1.05),
          ncol=4, fancybox=True, shadow=True)
ax2.text(10, 190, '58.60 min', style='normal',
        bbox={'facecolor':'blue', 'alpha':0.25, 'pad':10})
ax2.text(60, 190, '24.80 min', style='normal',
        bbox={'facecolor':'red', 'alpha':0.25, 'pad':10})
ax2.text(110, 190, '71.33 min', style='normal',
        bbox={'facecolor':'green', 'alpha':0.25, 'pad':10})
ax2.text(165, 190, '55.42 min', style='normal',
        bbox={'facecolor':'blue', 'alpha':0.25, 'pad':10})
ax2.text(210, 190, '27.54 min', style='normal',
        bbox={'facecolor':'red', 'alpha':0.25, 'pad':10})

ax0.plot(ts,.009*X_classified/np.max(X_classified))
ax0.set_ylim(0,.01)
ax0.set_ylabel('Phase')
majorticks = np.array([1,2,3],dtype='float')/3*.009
ax0.set_yticks(majorticks)
ax0.set_yticklabels(['I','II','III'])
ax0.set_xlim((ts[0],ts[-1]))
sns.rugplot(np.array(data), c=c1, ax=ax0)

# Set up the plots
c1, c2 = sns.color_palette("husl", 3)[:2]
# Plot the summed basis functions
summed_kde = np.sum(kernels, axis=0)
# ax1.plot(xx, summed_kde, c=c1)
sns.kdeplot(np.array(data), bw=bandwidth, linewidth=1, shade=True,
     color=c1, label=r'Peak Density', ax=ax1)
sns.rugplot(np.array(data), c=c1, ax=ax1)
ax1.set_yticks([])
ax1.set_ylabel('Density')
ax1.set_xlabel('Time (min)')
plt.savefig('gaus_process_fig2', bbox_inches='tight',dpi=300)
```

./AppendixA/gauss_proc.py

APPENDIX B

A Machine Vision Approach to Visualize and Analyze Intracellular Crystalloid Objects in Transmission Electron Microscope Images

B.1 Introduction

With the increasing use of electron microscopy, more studies rely on the use of computational tools to not only to detect and quantify morphological features, but also the ability to do so in an objective and efficient manner for large volumes of data. Structural comparisons were carried out, for example, between different microfibrils composed of collagen tetramers and banded aggregates[11]. For the analysis of repetitive texture features, fast Fourier transform (FFT) analysis was used to study protein aggregates composed of laterally-assembled microbfibrils which were themselves composed of collagen VI, an extracellular matrix component that forms structural links with cells. In a more chemical realm, the topology of monolayer graphene was analyzed for surface defects or rippling effects that were revealed by FFT procedures, helping to visualize changes in bond lengths[6]. Similarly, an FFT-based analysis of nanoparticle super-lattices and temperature-induced restructuring revealed hexagonal and distorted hexagonal structures in the range of 4-8 nm, suggesting ordered self-assembly of these structures based on center-to-center distances[15].

Lattice-like morphological features have also been identified in electron microscope images of intracellular structures. For example, endoplasmic reticulum differentiated into stacked arrays[17]; sinusoidally-packed compressed bodies[3]; membranous whorls [10]; crystalloid ER[7]; and stacked cytoplasmic membranes surrounding yeast nuclei [20]. Recently, studies have suggested cubic membranes that represent curved, three-dimensionally periodic structures based on mathematically-defined surfaces as models for such phenomena[12,13]. To study TEM micrographs in the context of theoretical 3D structures, a direct template correlative (DTC) matching method has been employed based on matching images with projections of three fundamental families of cubic membranes based on recognizing pattern and symmetry[1,8,12]. These templates rely, however, on a three dimensional structural hypothesis of the components they represent.

Here, we developed a more empirical, image analysis approach employing fast Fourier transforms to extract repetitive textured features in TEM images of specimens, whose molecular organization may not be fully known, such as crystalloid features that have been observed in certain drug-treated cells. Morphologically, we considered the smallest repeating feature of a textured pattern present in a crystalloid object as being analogous to the unit cell of a crystal without exactly corresponding to the physical repeating unit of a crystal. Therefore, as has been done with other crystals and crystalline-like objects, we hypothesized that crystalloid objects present in cells of living organisms may display symmetry and properties similar to long-range orientation and translational order of true crystals, which could thus be analyzed by FFT to identify the simplest morphological repetitive feature ("unit cell") as well as its orientation, spacings, and its relation to higher-order morphological features such as the variation of the unit cell and its alignment in relation to the overall shape of the crystalloid.

B.2 Methods

B.2.1 Development of algorithm

The algorithm was developed to select points of interest in the Fourier domain that appear brightly in the amplitude spectrum–here the logarithmic absolute value of the Fourier coefficients. Taking advantage of the symmetry of Fourier transforms, only the first and second quadrants were considered. The pixel intensity histogram of the amplitude spectrum displayed bi-modality for images containing lattice-like features. A band-pass filter was applied to keep only relevant frequencies in the 5 to 25 nm/cycle range. This effectively shifted the mean of the first distribution to the origin. A regression in MATLAB was used to fit the bimodal distribution, determining the two means and variances. Pixels that were two at least standard deviations above the mean were then retained as peaks from the Fourier domain. To validate the selected peaks and test against noise due to orientation, the images were rotated at two randomly-generated angles where-upon the same analysis was conducted, and only the preserved points were kept. The reconstructed image based on the detected points was compared to the original image.

B.2.2 Optimization image sets

An initial set a set of images was used based on proposed computer-generated two-dimensional projections of hypothetical three-dimensional structures representing periodic supramolecular structures[2,18]. This set, along with intensity- and noise-altered versions of each image, was used to help fine-tune the algorithm in the detection of lattice-like features. A test set based on 48 patterns was used to help fine-tune the fitting and ensure that the algorithm could detect lattice-like features. The test images were modified by adjusting the contrast and adding noise. Further testing was done on the images which these projections were suggested to match[2,18].

B.2.3 Quantitative analysis of lattice-like features

For carrying out intra-image comparisons, images were divided into non-overlapping square regions for analysis of local features to compare with long-range repetitive textures and symmetries. The size of these squares, 480 x 480 pixels, was determined based on the Nyquist-Shannon sampling theorem which, for the spatial domain, requires using more than twice the highest desired frequency to fully capture the repetitive feature and resolution while avoiding anti-aliasing[16] as well as the suggestion that the image frame need be several times larger than the characteristic frequencies to ensure the FFT captures them with adequate resolution[9]. Treating the FFT as a linear regression for each non-overlapping segment of the image, the sum of squares of the regression (reconstructed image versus original image) of the region had to be at least one percent of the total sum of squares (variance in the original image) of the region. The detected peaks in each region were then used to reconstruct and classify the patterns based on the dimensions of the unit cells composing them–the smallest repeating pattern that can account for the entire detected structure.

B.2.4 Application to a test set of images from treated animals

To test our visualization and analysis algorithm, we analyzed electron microscope images obtained from animals fed with clofazimine, a drug that induces crystalloid features to form inside macrophages[5]. These drug induced crystalloid objects possessed repetitive texture features when observed by TEM[14,19]. Briefly, mice were fed with drug with powder chow (3 mg/ml clofazimine in sesame oil, mixed at 0.01% oil to chow). Blood was collected from euthanized mice and fixed by perfusing 0.1M

Sorensen's buffer and Karnovsky's fixative (3% paraformaldehyde, 2.5% glutaraldehyde) infused to left ventricle and egressed to vena cava (2.5 ml/min). Tissues were minced smaller than 1 mm in each dimension followed by TEM sample preparation and imaging[4]. Control mice were fed with 0.01% oil to chow, and wash out mice were fed drug- and oil-free chow. Representative images were acquired using a Philips CM-100 electron microscope at magnifications from 4600X to 130000X.

B.3 Results

B.3.1 Development of machine vision algorithm

To help visualize the lattice-like features, we explored the use of FFTs. To identify peaks in the spatial domain corresponding to repetitive lines or bands in the raw images, we developed an algorithm to identify local maxima. This algorithm was optimized using sets of images adapted from[2,18] (Figure B.1). This included altering the sets of images to account for detected peaks in the presence of background noise; optimizing the algorithm; quantitatively analyzing the resultant peaks; and testing the algorithm on images with both the presence and absence of drug-cell aggregates; and overlaying the reconstruction of repetitive patterns onto the images for highlighting the lattice-like features and confirming their existence by visual inspection.

By visual inspection, the reconstructed spatial patterns of features detected with the algorithm were directly comparable to the spatial patterns of features in the raw images. In the training set of images (Figure B.1), the reconstructed structural features clearly overlapped with the actual features present in the images. For fine-tuning the peak detection algorithm, noise was added to a repeating structure (Figure B.1A; adapted from[2], Figure 1B), contrast was lowered between the image features and background (Figure B.1B; adapted from[2], Figure 1B), and as a further test, an EM image of an organized smooth endoplasmic reticulum (Figure B.1C; adapted from

[18], Figure 7D) was used. The resolutions were set such that the repeating patterns shown corresponded to features in the 5-25 nm range.

Each image was first converted to a corresponding discrete Fourier transform (Figure B.1D-F). Applying the algorithm, peaks in the 5-25 nm per cycle range were selected (Figure B.1G-I). By visual inspection, reconstructions based on the detected peaks appeared true to the original images. The noisy image had five clusters of contributive points that sufficed to recreate the pattern (Figure B.1J). The FFT of the low-contrast image proved more difficult to visually distinguish the peaks, however, with the selected points the recreated pattern appeared to match the original image (Figure B.1K). The EM image contained not only noise and varying contrast but also distortions in the repeating patterns due to being a biological sample. However, the detected peaks revealed a regular repeating structure throughout the image that indeed corresponded to one of the two-dimensional projections used in the DTC scheme[2].

B.3.2 Visualizing regions with repetitive texture features in electron microscope images

After the parameters were adjusted, images of cells with objects containing textured patterns with hints of lattice-like features were visualized and compared (Figure B.4). Displaying the results of repetitive feature detection algorithm directly on these EM images, the green highlighted regions corresponded to segments of the FFT image where peaks were detected. From these peaks, images were reconstructed using the inverse Fourier trans-form algorithm, to visualize the correspondence between the detected repetitive patterns and the texture of the segments. In this manner, we looked for macrophage-like cells in clofazimine-treated mice containing stained polyhedral objects with texture patterns (Figure B.4A-C). The red squares correspond to the magnified segments of the images. These segments displayed certain regular textures that appeared diagonally parallel but the repetitive details not readily perceptible by the eye (Figure B.4A), hexagonally-arranged repeating unit cells (Figure B.4E), and a meshwork pattern that was not easily discernible (Figure B.4F). The FFTs were next considered for each segment. The parallel pattern in Figure B.4A had five peaks arranged as a rectangle surrounding the origin of the FFT and set to an angle corresponding to the direction of the texture (Figure B.4G). The repeating pattern from Figure B.4B had six peaks in the frequency domain arranged in a hexagonal pattern. The meshwork from Figure B.4F had four peaks also creating a rectangle but set closer to the origin of the FFT (Figure B.4I). To facilitate visualization of the repetitive patterns in the selected regions, the image segments were reconstructed based on the peaks selected by the algorithm using the inverse Fourier transform function. The diagonally parallel striations were maintained (Figure B.4J). The repeating structures appeared to correspond to the original image (Figure B.4K). The meshed pattern was created from intersecting parallel lines in the original image that corresponded to each pair of peaks in the FFT image; one set in the north-west to south-east direction and the other in the south-west to north-east direction of the image (Figure B.4L).

B.3.3 Machine vision-assisted morphometric analysis

Next, we proceeded to study the morphological features of crystalloid objects observed in the cells of clofazimine-treated mice. For analysis, we identified clusters of selected regions containing lattice-like features as identified by the algorithm (Figure B.5A). Zooming into one of these regions (Figure B.5B), revealed what seemed like a lattice-like textured pattern. Magnification of a different, adjacent region (Figure B.5C) similarly revealed a repetitive texture pattern. After applying the Fourier transform to these two segments (Figure B.5D and E, respectively) a pattern of peaks was clearly observed in the FFT images. With the most prominent selected peaks in the spatial frequency domain, images were reconstructed corresponding to the patterns seen in the images. The re-construction for the first region revealed the underlying lattice-like features (Figure B.5F). The colored diagonals corresponded to the two major axes of symmetry. The intersection of repeating diagonals in along both of these axes yielded a unit cell that repeated throughout the image. A similar arrangement was also evident in the second region (Figure B.5G). To confirm the detected patterns, a reconstructed image based solely on selected frequencies was compared to the original segment being analyzed (Figure B.6A). Overlaying a semi-transparent copy of the reconstruction over the original image was used to highlight the corresponding lattice-like features in the original image with the unit cell shown in the center (Figure B.6B). The lattice arrangement, the shortest two directions in which the unit cell repeated, as well as the angle separating the vectors shown as purple arrows in Figure B.6C.

The internal arrangement of the unit cells was then compared for the entire image (Figure B.6). The extracted unit cells for each of the non-overlapping quadrants were used as metrics to define the regions analyzed by the vector lengths and angle of separation (Figure B.6A). Variations in the unit cells in Figure B.6A reflect slight variations in the detected peaks of the frequency do-mains which defined the unit cell shapes and sizes, which in turn affects the morphological definition of the unit cell boundaries. The distribution of the regions' properties, along with the two larger segments analyzed in Figure B.6, were plotted as a homogenous cluster, with the first edge measuring 2$\pm$1.99$\pm$0.07 nm, the second edge 11.14$\pm$0.15 nm, and the angle separating them 87.80$\pm$0.41 degrees (Figure B.6B). This indicated the presence of long-range order across the entire crystalloid object.

B.4 Discussion

Here, we developed a machine vision-based algorithm that can be fine-tuned for the detection, visualization and analysis of repetitive texture features in electron microscope images. A tailored version of this algorithm was implemented for detecting and enhancing repetitive structures at the nanometer scale on the order of 5 to 25 nm. Such repetitive, nanometer-scale crystal-like features were difficult to unambiguously detect with the naked eye, especially given images with low contrast and noisy background. Applied to electron microscope images of macrophages of mice fed with a clofazimine-supplemented diet for a period of several months, the algorithm revealed structural details about the internal organization of drug induced, crystalloid objects evident in macrophages. Consistent with previous reports, these crystalloid objects possessed a bounding membrane and were polyhedral in shape, yet their internal organization seemed almost featureless, aside from global textures, when viewed by the naked eye. By fine-tuning the machine vision algorithm with a training set of images, pattern detection and visualization of lattice-like features was readily per-formed. By applying the algorithm to non-overlapping square regions of images, we were able to further optimize the algorithm so it only detected regions containing the putative, nanometer-scaled periodic features. After optimization, the algorithm mostly detected nanometer scale lattice-like features in the regions that were associated with specific objects of interest. The algorithm did not detect as many regions from the same image outside the polyhedral membrane. The algorithm also did not detect regions of images of control, untreated animals that lacked lattice-like features.

Applying this optimized algorithm to larger domains of the drug-induced polyhedral structures present in macrophages of clofazimine-treated animals, we were able to obtain detailed in-formation about the spatial arrangement, symmetry and spacings of periodic features present in those domains. Within a single polyhedral structure, different domains could be compared, to establish the long range similarities or differences across the interior of the object. Furthermore, applying the same algorithm to different polyhedral structures found in the same image, we were able to establish the similarities and differences in the interior organization of different structures. Strikingly, within an individual polyhedral structure, the machine vision algorithm revealed the presence of an asymmetric, repetitive unit, with internal features in the order of 5 nm in size, and a unit cell size of 10 to 20 nm (Figure B.6). Without computational assistance, this level of detail was not so apparent by visual inspection (Figure B.6). Different domains within the membrane-bound polyhedral object possessed similar features in the same orientation, indicating a long-range order (Figure B.6. In the particular example analyzed in this study (Figure B.6), the mean perimeter of the parallelogram circumscribing pairs of red and blue segments was 47.60 ± 1.80 nm with calculated interior angles of 84.61 ± 5.12 and 95.39 ∓ 5.12 and degrees. The observed variations in morphological unit cell across a single crystalloid object were small, and could have been due to (i) lack of complete homogeneity in the repetitive texture of the samples being imaged, (ii) distortive noise perturbing the position of detected peaks in the Fourier domains, and (iii) ambiguity in the selection and filling of the boundaries of the region that defines the unit cell as the smallest repeating segments that describes the texture pattern across the entire image.). Interestingly, the overall shape of the outer membrane bound perimeter of the entire structure was parallel to the lattice vectors of the unit cell, suggesting that the observed unit cell arrangement may be structurally linked to the overall shape of the crystalloid.

Nevertheless, to interpret the observed patterns, it is important to note that one of the difficulties of standard electron microscopy is that it only allows one to view a thin cross sectional plane across the structure of interest. Therefore, the plane at which the structure is cut, and the thickness of the section may influence the appearance of the internal organization of the structure. In the future, more sophisticated electron microscopy tomography techniques that allow for visualizing the organization of three-dimensional structures may prove useful for gaining further in-sights into the morphology of drug induced membrane aggregates.

B.5 Conclusion

We have successfully developed and demonstrated the usefulness of a machine vision approach for detecting, extracting and enhancing low-contrast and distorted repetitive morphological features in transmission electron microscope images, without a priori knowledge about their molecular organization. Beyond detection and visualization of lattice-like features, the algorithm could be further developed and applied towards quantitative analysis of the shape, angles, spacings of the unit cell, and spatial analysis of variation in unit cell features and orientation across the entire object and with respect to the object perimeter and surrounding morphological features, in an objective, quantitative and reproducible manner.

B.6 Acknowledgement

Work funded by NIH grant R01GM078200 to GRR.

B.7 Figures

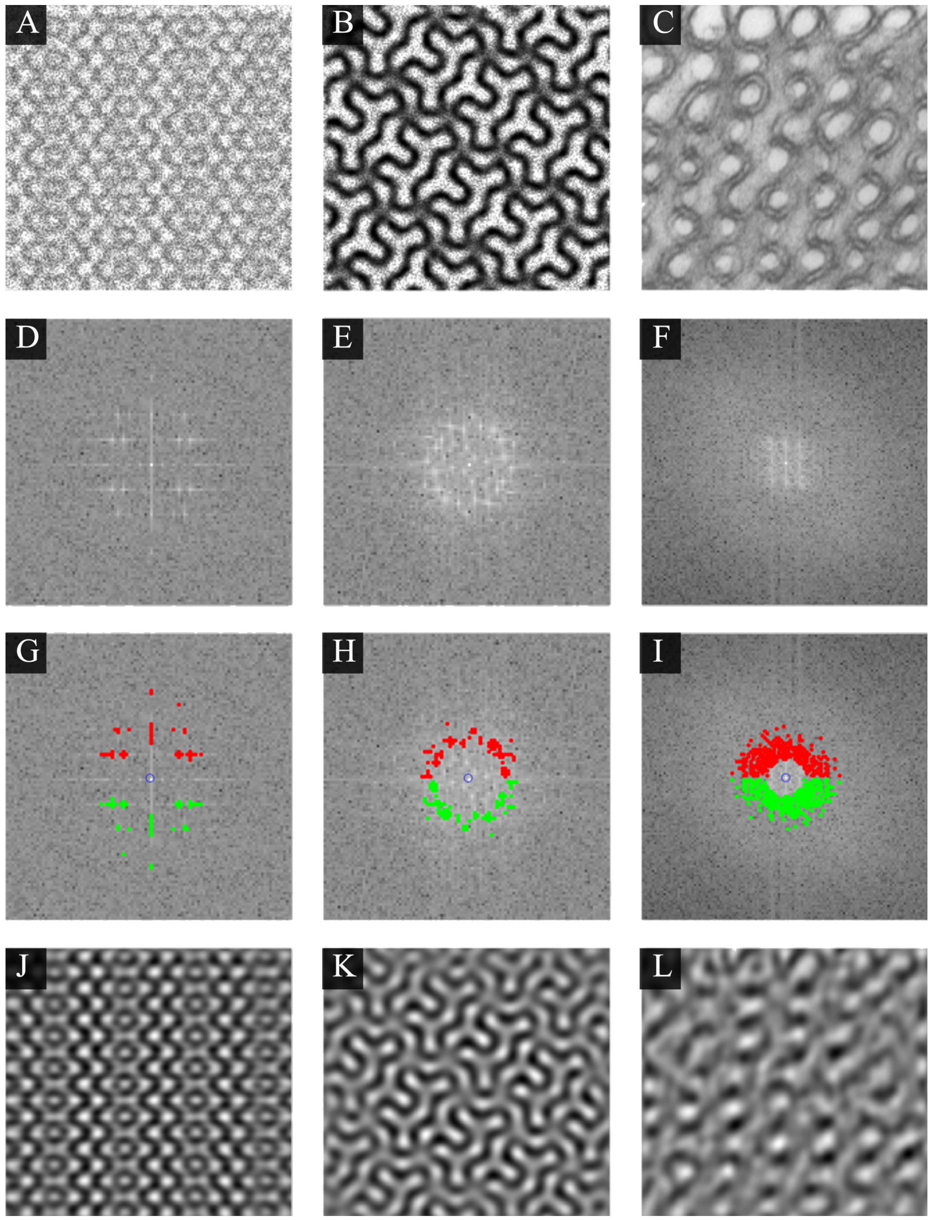


Figure B.1: (A-B) Images adapted from [2] and (C) [18]. (D-F) The FFTs of the template images (G-H) Detected peaks in the Fourier domain (J-H) Reconstructions based on the detected peaks.

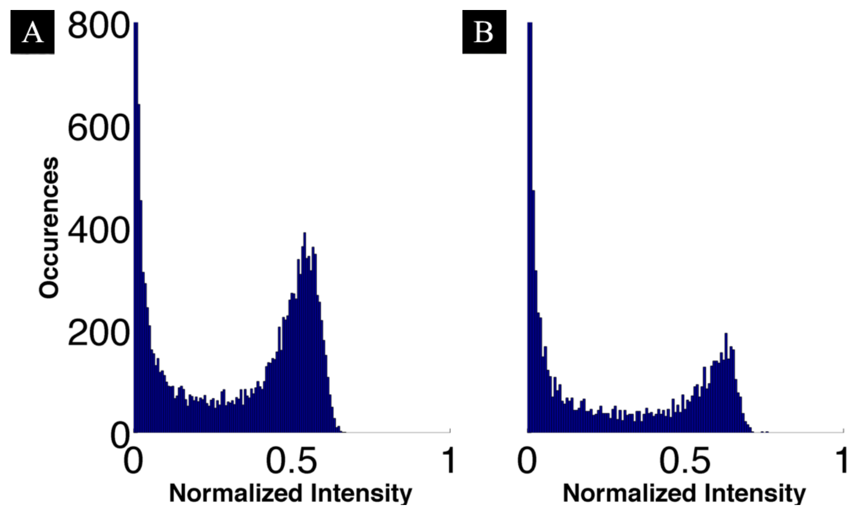


Figure B.2: (A) Normalized Fourier domain histograms of Figure B.1C and (B) Figure B.4F.

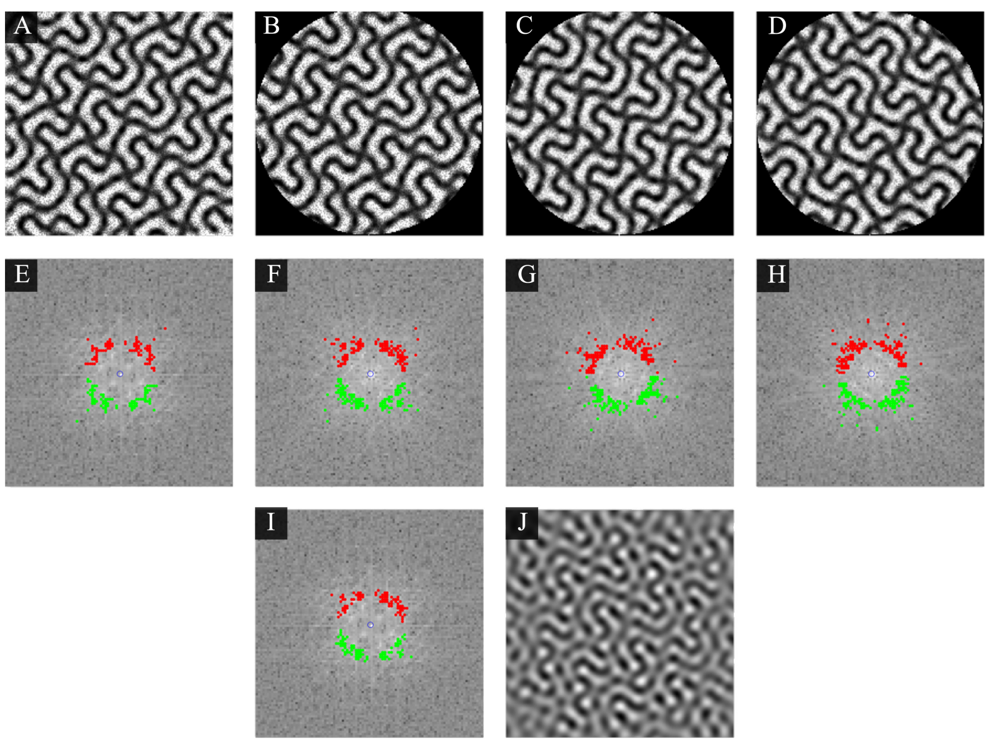


Figure B.3: (A) Image from Figure B.1B. (B) A circularly cropped. (C-D) B rotated 15° and 95°, respectively. (E-F) The detected peaks from the FFTs of images A-D. (I) The peaks that are maintained under rotation. (J) Reconstructions based on the maintained peaks

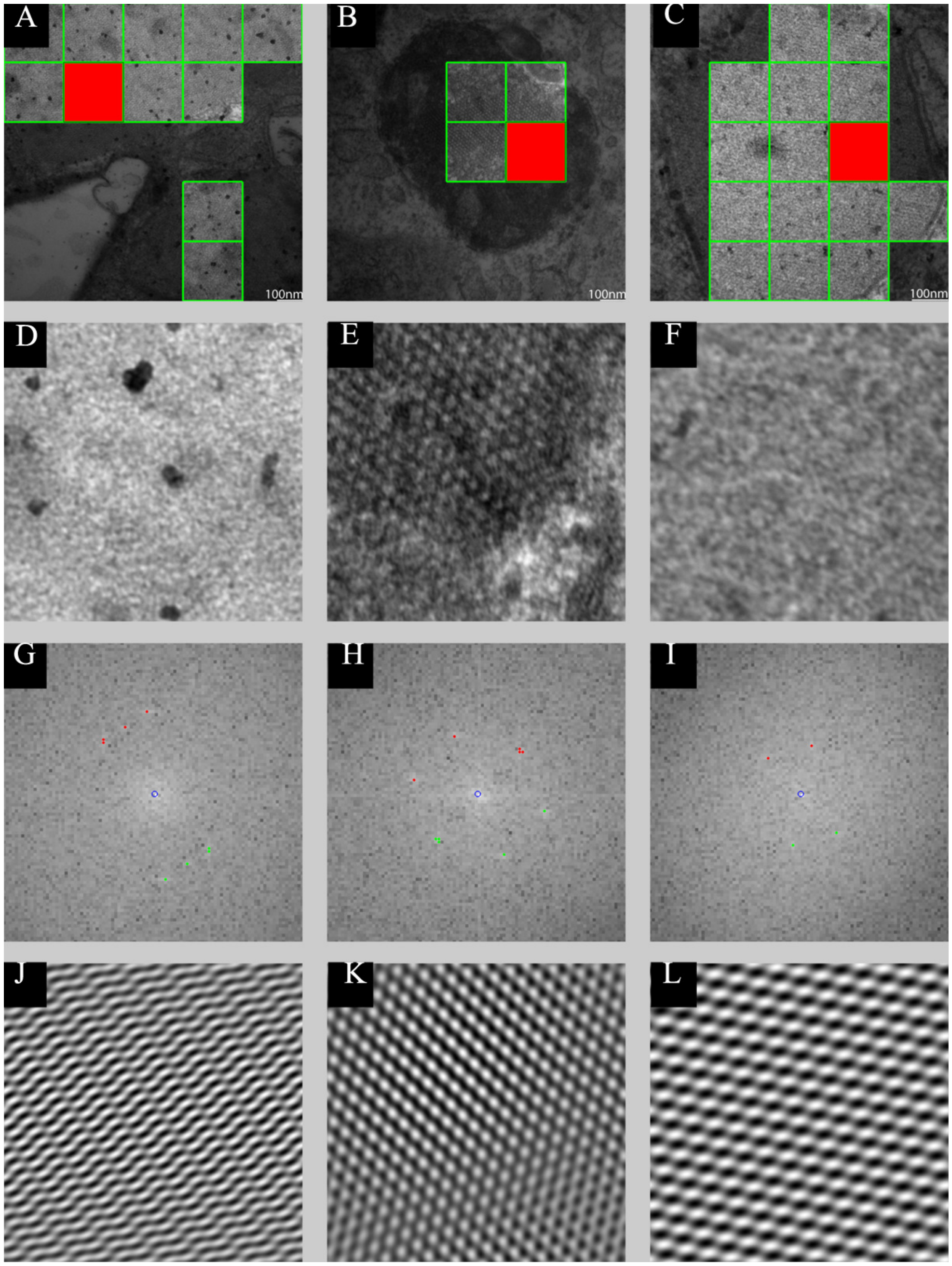


Figure B.4: Detected regions of interest in images. (A), (B), and (C) The highlighted boxes show regions where peaks were detected in images of intestinal villi of treated mice. The red regions in each image are magnified in (D), (E), and (F). Detected points are shown in (G), (H), and (I). The detected points were used to reconstruct the images in (J), (K), and (L), in which regular patterns are seen.

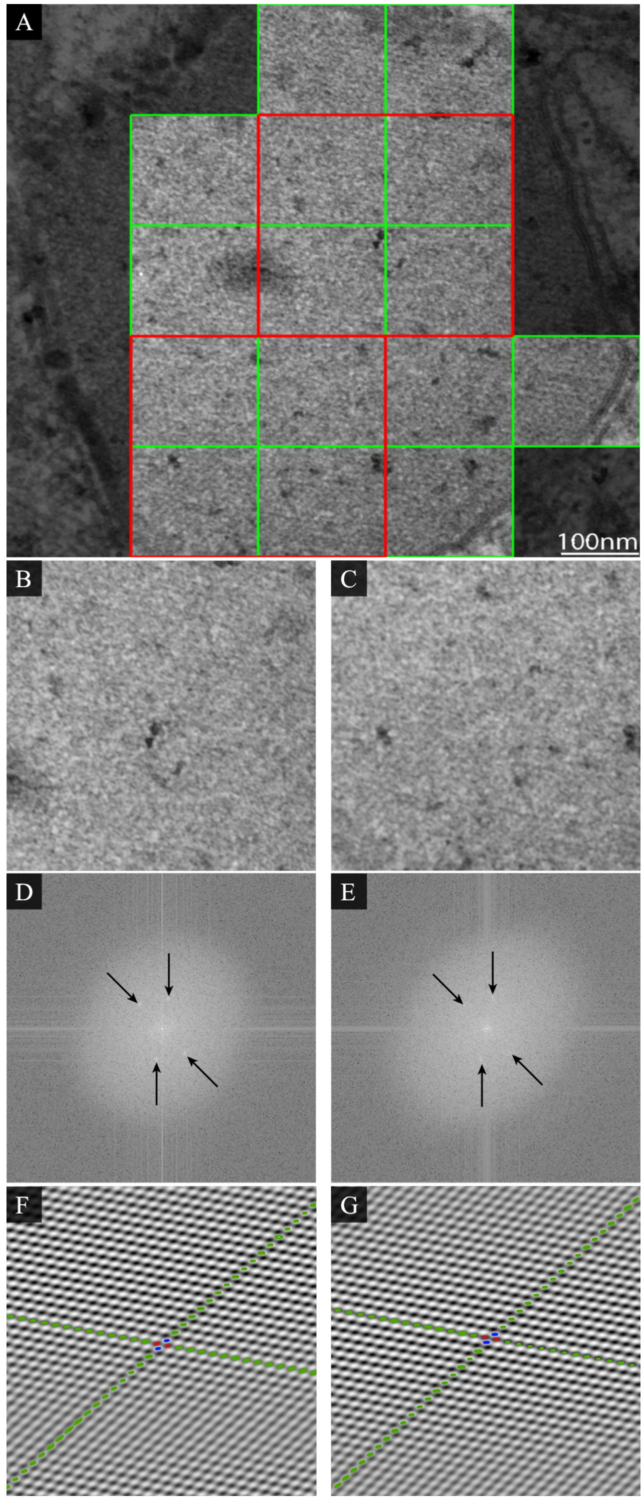


Figure B.5: (A) Contiguous regions were selected from figure B.4C, magnified in (B) and (C); the Fourier transforms are shown in (D) and (E), along with corresponding reconstructions showing hexagonal patterns highlighted in blue in (F) and (G), respectively.

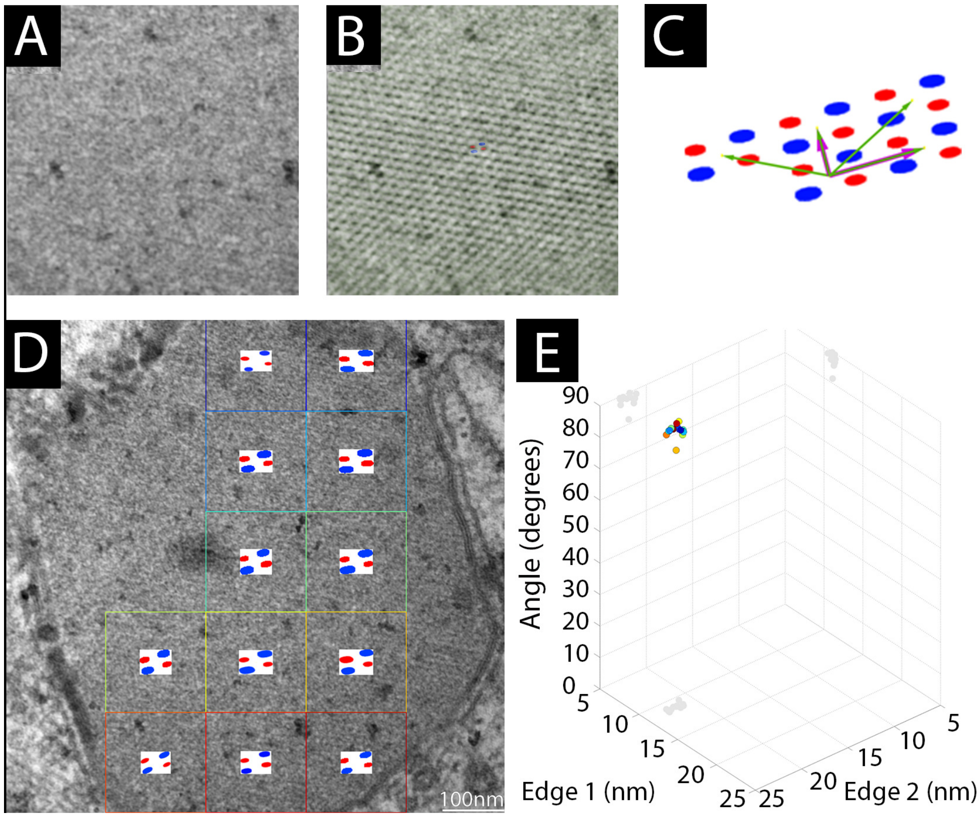


Figure B.6: (A) Selected region in a crystalloid object. (B) Inverse Fourier transform of the peaks in the FFT, superimposed on the same selected region as in A. (C) Repetitive pattern in the crystalloid object, with the unit cell indicated by purple arrows. (D) Quadrants in a hexagonal crystalloid were peaks were detected in the FFT. Superimposed are the enlarged unit cells of each quadrant. (E) Plot of the two shortest directions and the angle of separation. The colors of the circles correspond to the colored quadrants in D. The average measure for edge 1 was 21.99±0.07 nm, edge 2 11.14±0.15 nm, and the angle separating them 87.80±0.41 degrees.

B.8 MATLAB Code

```
function [pve heatmap points] = fft_analysis(img,fact,res,scale)
% Run entire analysis on an image. Given the input image, threshold factor
% (how many standard deviations above mean should be considered),
% resolution, and scale bar, the PVE, corresponding heatmap, and detected
% points are output. The PVE is just used to map where on the figure
% detected patterns are found. The input image is assumed to be 2400x2400
% pixels as output by the transmission electron microscopy software at MIL.

global temp_filt;

scale_r = scale(:,1);
scale_c = scale(:,2);

% Step size, or the dimension (one side) of non-overlapping square region
% to be analyzed.
cutoff = 0.01; step = 480;

% If a resolution is given as input use it else default
if res, temp_filt = fft_filt(step/2,0,res);
else temp_filt = fft_filt(step/2); end

% Scan the image in the required step size without displaying the results.
[pve, heatmap, points] = fft_scan(img(1:2400,1:2400),step,2,res,0);

%% Display results of scan

figure(1); clf;
mysubplot(4,1,1)

% Show image with the regions of detected patterns highlighted
temp_img_selected = 0.4.*img(:,:,1);
for i = 1:length(scale_r)
    temp_img_selected(2200-1+scale_r(i),2075-1+scale_c(i)) = 255; end;
imshow(temp_img_selected(1:2400,1:2400));
hold on;
g = imshow(img(1:2400,1:2400));
set(g,'AlphaData',(heatmap>cutoff))

% Wherever a pattern is discovered (PVE>cutoff) draw a square
[m n] = find(pve>cutoff);
for i=1:length(m)
```

```
            hold on; rectangle('Position',[(m(i)-1)*step+1 (n(i)-1)*step+1 step-1 step-1],'EdgeColor','g','LineWidth',2);
end;

% Select one of the pattern-containing squares as an example
[r c] = find(pve(3:end,3:end)>cutoff,1,'first') ;  c = (c-1+3); r = (r-1+3);
temp_img = img((r*step)-(step-1):r*step,(c*step)-(step-1):c*step);
hold on; rectangle('Position',[(c*step)-(step-1),(r*step)-(step-1),step,step],'FaceColor','r');
mysubplot(4,1,2);
imshow(temp_img,[]); % Display the segment
mysubplot(4,1,3);
temp_pts = fft_points3(fftshift(fft2(temp_img)),fact,1); xlim([2*step/5 3*step/5]); ylim([2*step/5 3*step/5]); % Show the FFT and detected peaks for that region
reduced_3 = fft_reconstruct(fftshift(fft2(temp_img)),(temp_pts));
mysubplot(4,1,4);
imshow((ifft2(ifftshift(reduced_3))),[]); % Reconstruct and display image based on detected peaks

%% Analyze detected region
% figure(2); clf;
% points = struct('pts',[0 0],'theta',0,'radius',0);
%
% [M N] = size(pve);
%
% angle_count_map = ones(M,N).*-1;
% angle_mean_map = ones(M,N).*-1;
% angle_skew_map = ones(M,N).*-1;
% angle_kurtosis_map = ones(M,N).*-1;
%
% rad_var_map = ones(M,N).*-1;
% rad_mean_map = ones(M,N).*-1;
% rad_skew_map = ones(M,N).*-1;
% rad_kurtosis = ones(M,N).*-1;
%
% % Show image with highlighted regions corresponding to areas where patterns
% % were detected
% temp_img_selected = 0.4.*img(:,:,1);
% for i = 1:length(scale_r)
%     temp_img_selected(2200-1+scale_r(i),2075-1+scale_c(i)) = 255; end;
% imshow(temp_img_selected(1:2400,1:2400));
% hold on;
% g = imshow(img(1:2400,1:2400));
% set(g,'AlphaData',(heatmap>cutoff))
%
% % Determine angle and radius for each detected region
% theta = 0:.5:180; radians = .5:.5:step/2;
% ind = 0;
% [m n] = find(pve>=cutoff);
% for i=1:length(m)
```

```
%               ind = ind+1;
%               hold on; rectangle('Position',[(m(i)-1)*step+1 (n(i)-1)*step+1 step-1 step-1],'EdgeColor','g','LineWidth',2);
%
%               k = (n(i)-1)*step+1;
%               j = (m(i)-1)*step+1;
%
%               temp_segment = img((n(i)-1)*step+1:(n(i)-1)*step+1+step-1,(m(i)-1)*step+1:(m(i)-1)*step+1+step-1);
%               temp_fft = fftshift(fft2(double(temp_segment)));
%               points(ind).pts = fft_points3(temp_fft,fact,0);
%               if(points(ind).pts)
%                   clear m_t m_r bin_t bin_r
%                   [points(ind).theta, points(ind).radius] = fft_angle(temp_fft,points(ind).pts);
%
%                   points(ind).theta(points(ind).theta<0) = points(ind).theta(points(ind).theta<0)+180;
%                   points(ind).theta(points(ind).theta==0) = 180;
%                   [m_t,bin_t] = histc(points(ind).theta,theta);
%                   [m_r,bin_r] = histc(points(ind).radius,radians);
%                   num_bins = length(bin_t);
%                   % Draw vectors proportional to angle and radius for each
%                   % set of points.
%                   for kk = 1:length(bin_t)
%                       arrow_rad = (radians(bin_r(kk))/max(radians(bin_r(:))))*(step/2);
%                       u = arrow_rad*cosd(theta(bin_t(kk))); v = arrow_rad*sind(theta(bin_t(kk)));
%                       hold on; quiver(j+step/2,k+step/2,u,v,'Color',[1 1 1],'LineWidth',3); hold off;
%                       hold on; quiver(j+step/2,k+step/2,-u,-v,'Color',[1 1 1],'LineWidth',3); hold off;
%
%                   end
%                   % Statistics of radii and angles
%                   rad_var_map(n(i),m(i)) = var(radians(bin_r));
%                   rad_mean_map(n(i),m(i)) = mean(radians(bin_r));
%                   rad_skew_map(n(i),m(i)) = skewness(radians(bin_r));
%                   rad_kurtosis(n(i),m(i)) = kurtosis(radians(bin_r));
%
%                   angle_count_map(n(i),m(i)) = length(theta(bin_t));
%                   angle_mean_map(n(i),m(i)) = median(theta(bin_t));
%                   angle_skew_map(n(i),m(i)) = skewness(theta(bin_t));
%                   angle_kurtosis_map(n(i),m(i)) = kurtosis(theta(bin_t));
%               end
% end;
```

```
%
% % Resize into matrix
% j = 0; k = 1; l = 1;
% [m n] = find(pve>=cutoff);
% for i=1:length(m)
%         j=j+1;
%         if(points(j).pts)
%                 temp_size = length(points(j).theta(:,1));
%                 temp_ang(k:k+temp_size-1,1) = points(j).theta;
%                 k = k+temp_size;
%                 temp_size = length(points(j).radius(:,1));
%                 temp_rad(l:l+temp_size-1,1) = points(j).radius;
%                 l = l+temp_size;
%         end
% end
%
%
% %% Display the average radius and histogram of angle distsributions
% figure(3); clf;
% max_val = round(max(rad_mean_map(:)));
% cmap = mysubplot(2,1,1);
% imagesc(rad_mean_map); % title('Mean Radius');           %# Create a colored plot of the matrix values
% % caxis([0 max_val]);  h_cb2 = colorbar();
% textStrings = num2str(rad_mean_map(:),'%0.2f');  %# Create strings from the matrix values
% textStrings = strtrim(cellstr(textStrings));  %# Remove any space padding
% textStrings = strrep(textStrings,'-1.00','');
% [x,y] = meshgrid(1:length(rad_mean_map));   %# Create x and y coordinates for the strings
% hStrings = text(x(:),y(:),textStrings(:),...      %# Plot the strings
%                'HorizontalAlignment','center');
% midValue = mean(get(cmap,'CLim'));  %# Get the middle value of the color range
% textColors = repmat(rad_mean_map(:) > midValue,1,3);  %# Choose white or black for the
%                                                %#   text color of the strings so
%                                                %#   they can be easily seen over
%                                                %#   the background color
% set(hStrings,{'Color'},num2cell(textColors,2));  %# Change the text colors
% set(cmap,'XTickLabel',{},...  %#  Clear axes
%         'YTickLabel',{},...
%         'TickLength',[0 0]);
%
%  colormap((flipud(bone+pink)./2).^.6)
%  freezeColors;
%
```

```
% fplot = mysubplot(2,1,2);
% numbins = 100; n = length(temp_ang);
% binwidth = range(temp_ang)/numbins;
% edg = 0:binwidth:180;
% [count,bin] = histc(temp_ang,edg);
% % h = bar(edg,count,'histc');
% [temp_count,temp_bin] = histc(temp_ang,0:1:180);
% h = bar(0:1:180,temp_count./max(temp_count));
%   set(h, 'facecolor', [0.2 0.2 1]); % change the color of the bins
%   set(h, 'edgecolor', [0.2 0.2 1]);
% p = count;
% hold on;
% % plot(edg,p./max(p),'Color',[1 .5 .5],'LineWidth',4);
% xlim([0 180]); % plot(p,edg) is the smooth curve representing the probability density function you are looking for.
% % set(fplot,'FontSize',55);
% xlabel('Angle');%,'fontsize',55,'fontweight','b');
% ylabel('Frequency');%,'fontsize',55,'fontweight','b');


% set(findobj(gcf,'Type','text'),'FontSize',55,'fontweight','b');
```

./AppendixB/src/fft_analysis.m

```
function [theta, radius] = fft_angle(fourier,points,res)
% Take a Fourier transform and the detected points of interest, compute
% angles and radii for each set of symmetric points and return as vectors
% theta and radius. The resolution of the corresponding image can be input
% to scale accordingly.

global Fx Fy;

% Check for number of input arguments. If resolution is not given, assume
% standard (292, corresponding to 130,000X magnification image).
if nargin < 3, res = 292; end;

[M N] = size(fourier);

% Sampling frequeny (pixels per nm)
fsy = res/100; fsx = res/100;

% nm per pixel
dx = 1/fsx; dy = 1/fsy;

% pixels
x = dx*(0:N)'; % nm
y = dy*(0:M)';
```

```
% cycles per nm
dFx = fsx/(N); dFy = fsy/(M);
Fx = (-fsx/2:dFx:fsx/2-dFx)';
Fy = (-fsy/2:dFy:fsy/2-dFy)';

[M N] = size(fourier);

num_pts = length(points)/2;
radius = zeros(num_pts,1); theta = zeros(num_pts,1);

% Iterate through the input points and calculate the radius and angle at
% which they fall with respect to the center (analogous to origin if we
% assume Cartesian coordinates centered over the image).
for i = 1:num_pts
   y = points(i,1); x = points(i,2);
   yy = M/2-y;
   if (x>=N/2), xx = x-N/2; theta(i) = atan2d(yy,xx);
   else xx = N/2-x; theta(i) = atan2d(yy,xx); theta(i) = 180-theta(i);
   end;
   radius(i) = 1/((Fx(x)^2+Fy(y)^2)^.5);
end
```

./AppendixB/src/fft_angle.m

```
function [filt] = fft_filt(dim,show,resolution)
% Create a Butterworth band pass filter of size dim given an input
% resolution corresponding to the image which is being analyzed.

if nargin < 2, show = 0;  resolution = 292;
elseif nargin < 3, resolution = 292; end;

ind = (10*dim/400);
d0 = (13*dim/400);
d1 = (60*dim/400);

% Scale with respect to default resolution (corresponds to image of 130000X
% magnification)
ind = ind/resolution*292;
d0 = d0/resolution*292;
d1 = d1/resolution*292;

filt1 = ones(2*dim,2*dim);
filt2 = ones(2*dim,2*dim);
% Use Butterworth band pass filter.
for i=1:2*dim
    for j = 1:2*dim
        % Radial distance from center
        dist = ((i-(dim+1))^2 + (j-(dim+1))^2)^.5;

```

```
        % high pass filter
        filt1(i,j)= 1/(1 + (dist/d0)^(2*ind));
        filt1(i,j)= 1.0 - filt1(i,j);

        % low-pass filter
        filt2(i,j)= 1/(1 + (dist/d1)^(2*ind));
    end
end

% Combine the two filters and normalize
filt = filt1.*filt2;
filt = filt./max(filt(:));

% Display filter if required by user
if show, subplot(1,3,1); imshow(filt1); subplot(1,3,2);
    imshow(filt2); subplot(1,3,3); imshow(filt), end;
```

./AppendixB/src/fft_filt.m

```
function [ points ] = fft_points3( fourier, fact , show,
    resolution)
% fft_points3
% takes as input the FFT of an image along with the
% multiplication factor determining how many standard deviations
    above mean
% to look for peaks, a show toggle to display the results, and a
    resolution
% scaling factor (default resolution is assumed to be 130000X,
    this is the
% number of pixels per nm based on the scale bar in the TEM image)
    .

global temp_filt scrsz Fx Fy;
points = [0 0];
sym_points = [0 0];

% Check input arguments and fill incomplete ones.
if nargin < 4 || resolution == 0, resolution = 292;
elseif nargin < 3, show = 0; resolution = 292;
elseif nargin < 2, show = 0; fact = 1; resolution = 292;
end

% Ensure conversion to double and create the filter.
[M N] = size(fourier); fourier = double(fourier);
temp_filt = fft_filt(M/2,0,resolution);

% Sampling frequeny (pixels per nm)
fsy = resolution/100; fsx = resolution/100;

% nm per pixel
dx = 1/fsx; dy = 1/fsy;

% pixels
```

```
x = dx*(0:N)'; % nm
y = dy*(0:M)';

% cycles per nm
dFx = fsx/(N); dFy = fsy/(M);
Fx = (-fsx/2:dFx:fsx/2-dFx)';
Fy = (-fsy/2:dFy:fsy/2-dFy)';

% Take the log of the absolute value, this is the power spectrum.
transform = log(1+abs(fourier));
% Normalize
original_fft = transform./max(transform(:));
temp_fft = transform;
temp_fft = temp_fft./max(temp_fft(:));
% Apply Butterworth filter
temp_fft = temp_fft.*temp_filt;

% Take non-zero elements (assume anything smaller than order of 10^-3 is
% zero
temp = temp_fft(find(temp_fft>5e-3));
numbins = 120; n = length(temp);
binwidth = range(temp)/numbins;
edg = 0:binwidth:1;
[count,bin] = histc(temp,edg); p = count;
[temp_count,temp_bin] = histc(temp,0:.001:1);

% fit bimodal distributions to our defined function using an extreme value
% possibility density function (with most values distributed near 0) and a
% normal possibility density function.
clear x y;
f = @(y,x)y(1)*evpdf(x,y(2),exp(y(3)))+(1-y(1))*normpdf(x,y(4),exp(y(5)));
% Set options
temp_opts = statset('MaxIter',700, 'MaxFunEvals',1000,'FunValCheck','off');
% Assume starting values, contribution of each pdf is a half here. The
% log variances are used in the input vector.
% [fraction contribution, mu1, sigma1, mu2, sigma2]
t0 = [0.5 0.01 -7 .6 -5.3];
% Use non-linear least squares regression to fit the defined function above
% The iteration ensures the function is fit within 800 trials and that we
% are not getting extraneous solutions (bounded variance).
temp_pdf = nlinfit(edg',p./max(p(:)),f,t0,temp_opts);
iter = 1;
while(temp_pdf(5) < -6.4 || temp_pdf(5) > -1 && iter < 800)
    if temp_pdf(5)<-6.4, t0(5) = t0(5)+.01; end;
    if temp_pdf(5)>-1, t0(5) = t0(5)-.01; end;
    temp_pdf = nlinfit(edg',p./max(p(:)),f,t0,temp_opts);
```

```
    iter = iter+1;
end

% Iterate through the probability distribution function and find the point
% that satisfy our condition of being a factor of the variance fact*sigma
% greater than the mean (generally a factor of 2).
if(temp_pdf(4)>temp_pdf(2) && temp_pdf(1)<=1 && temp_pdf(1)>0 && temp_pdf(2)>0)
    [r c] = find(temp_fft>=(temp_pdf(4)+fact*exp(temp_pdf(5))));
    temp_mat = zeros(M,N);
    for m=1:length(r), temp_mat(r(m),c(m)) = original_fft(r(m),c(m)); end;

    % Fill in the matrix for the corresponding symmetry.
    [r c] = find(temp_mat(1:M/2,1:N)>1e-1);
    points = [r c];
    sym_points = [M-r+(2*~mod(M,2)+mod(M,2)) N-c+(2*~mod(M,2)+mod(M,2))];
end

% Display the FFT and detected peaks if user has requested
if show
     imshow(original_fft); hold on; plot(round(N+1)/2,round(M+1)/2,'bo');
     if ([points])
         plot(points(:,2),points(:,1),'r.','MarkerSize',10); hold on ; plot(sym_points(:,2),sym_points(:,1),'g.','MarkerSize',10); hold off;
     end;
     hold off;
end
% Return the pairs of detected peaks
points = [points; sym_points];
```

./AppendixB/src/fft_points3.m

```
function [img_PVE,img_points] = fft_pve(image,fact,viz,res)
%   Input variable is M by N image and a box win that is a scalar factor
%   of the image. The peaks in the Fourier domain are isolated and used to
%   reconstruct a new image showing the base pattern. Thus the Proportion
%   of Variancec Explained, or PVE, is the ratio variance in the
%   reconstructed versus original image.

% Check input arguments and set defaults if not given by user.
if nargin < 4, res = 0; end;
if nargin < 3, viz = 0; elseif nargin < 2, viz = 0; fact = 1; end

[M N] = size(image);
```

```
% Variance of original image
img_var = var(double(image(:)));

% Take 2D FFT of image
img_fft = fftshift(fft2(double(image)));

% Scale according to input resolution if given
if(res), img_points = fft_points3(img_fft,fact,viz, res);
else img_points = fft_points3(img_fft,fact,viz); end;

% If points are detected, ensure they survive rotation and
    calculate the
% PVE (proportion of variance explained)
if(img_points)
     % Random rotations to ensure peaks are not aberrations due to
     % orientation
     temp_reduced = fft_rotate(image,fact,length(image),res,0);

     [r c] = find(temp_reduced);
     temp_reduced_points = [r c];
     img_points = img_points(ismember(img_points,
    temp_reduced_points,'rows'),:);

     % Reconstruct image from FFT with only detected peaks
     img_rec = ifft2(ifftshift(fft_reconstruct(img_fft,img_points))
    );

     % Compute variance in the reconstructed image
     img_rec_var = var(double(img_rec(:)));

     % PVE - ratio of variance in reconstructed image vs original
    image
     img_PVE = img_rec_var/img_var;
else
     img_PVE = 0;
end
```

./AppendixB/src/fft_pve.m

```
function reduced_fft = fft_reconstruct(fourier,fftpoints)
%   Input variable is M by N image transfrom (FFT) and the highest
     peaks
%   (thresholded by fft_points3() function)
%   An ouptut minimal FFT is returned that contains only the
    central peak
%   (which is the image intensity) and the detected peaks.
K = size(fftpoints,1);

[M N] = size(fourier);
reduced_fft = zeros(M,N);

if fftpoints
```

```
    for k = 1:K
    reduced_fft(fftpoints(k,1),fftpoints(k,2)) = fourier(fftpoints
   (k,1),fftpoints(k,2));
    end,
end;

% Normalize the peaks
reduced_fft(round((M+1)/2),round((N+1)/2)) = max(fourier(:));
```

./AppendixB/src/fft_reconstruct.m

```
function [reduced_fft] = fft_rotate(img, thresh, dim, resolution,
    show)
% fft_rotate takes as input an image, the threshold scale (
    multiplier of
% variance), dimension of non-overlapping square to be used,
    picture
% resolution, and an optional display toggle.
% The image is run through the detection algorithm but also using
    random
% rotation of cropped, circular region of the image to test which
    peaks are
% preserved under this transformation. This helps reduec erroneous
     peaks
% detected due simply to the orientation of the original image.

% Check input arguments and set defaults
if nargin < 5, resolution = 0; show=0; end;

% Generate random angles for rotation.
for k = 1:2, angles(k) = (k-1)*90+randi(90,1); end;

% Create circular crop filter.
[rr cc] = meshgrid(1:dim);
rad = ceil(length(img)/2+(1-mod(length(img)/2,2)));
C = sqrt((rr-rad).^2+(cc-rad).^2)<=rad;

clear theta2 theta3 theta5 theta6
clear radius2 radius3 radius5 radius6

temp_pts2 = 0;
temp_pts3 = 0;
temp_pts5 = 0;

% Crop image into circular region and detect peaks. This is
    attempted
% iteratively while decreasing the threshold each time as
% rotational/cropping distortion might obscure the peaks.
while(~exist('theta2') | isempty(temp_pts2) | mean(temp_pts2)==0)
    % Set the threshold minimum boundary
    thresh = max([thresh,1.9])*.85;
    % Crop image into circular region
    temp_img2 = double(img).*C;
```

```
    % Detect points
    temp_pts2 = fft_points3(fftshift(fft2(temp_img2)),thresh,show,resolution);
    % Determine cartesian orgin (center of image is {0,0})
    origin = [rad rad];
    length(temp_pts2)
    % Calculate radial distance from origin
    for i = 1:length(temp_pts2)
        y = temp_pts2(i,1); x = temp_pts2(i,2);
        yy = rad-y;
        if (x>=rad), xx = x-rad; theta2(i) = atan2d(yy,xx);
        else xx = rad-x; theta2(i) = atan2d(yy,xx); theta2(i) = 180-theta2(i); end
        radius2(i) = ((xx)^2+(yy)^2)^.5;
    end
    % Determine Cartesian coordinates of detected peaks based on polar
    % coordinates determined above.
    phi2 = zeros(1,length(theta2));
    temp_pts2fin = overlay_rot(radius2,theta2,phi2,origin);
    % Reconstruct image using the set of detected peaks
    reduced_2fin = fft_reconstruct(fftshift(fft2(temp_img2)),temp_pts2fin);
    thresh = thresh - .1;
end

% Random rotation #1 of cropped region. Same process as above aside from
% rotation.
while(~exist('theta3') | isempty(temp_pts3) | mean(temp_pts3)==0)
    thresh = max([thresh,1.9])*.85;
temp_img3 = imrotate(img,angles(1),'bilinear','crop');
temp_img3 = double(temp_img3).*C;
temp_pts3 = fft_points3(fftshift(fft2(temp_img3)),thresh,show,resolution);
for i = 1:length(temp_pts3)
    y = temp_pts3(i,1); x = temp_pts3(i,2);
    yy = rad-y;
    if (x>=rad), xx = x-rad; theta3(i) = atan2d(yy,xx);
    else xx = rad-x; theta3(i) = atan2d(yy,xx); theta3(i) = 180-theta3(i); end
    radius3(i) = ((xx)^2+(yy)^2)^.5;
end
phi3 = repmat(angles(1),1,length(theta3));
temp_pts3fin = overlay_rot(radius3,theta3,phi3,origin);
reduced_3fin = fft_reconstruct(fftshift(fft2(temp_img3)),temp_pts3fin);
thresh = thresh - .1;
end

% Random rotation #2 of cropped region. Same process as above aside from
% rotation.
while(~exist('theta5') | isempty(temp_pts5) | mean(temp_pts5)==0)
```

```
    thresh = max([thresh,1.9])*.85;
temp_img5 = imrotate(img,angles(2),'bilinear','crop');
temp_img5 = double(temp_img5).*C;
temp_pts5 = fft_points3(fftshift(fft2(temp_img5)),thresh,show,resolution);
for i = 1:length(temp_pts5)
    y = temp_pts5(i,1); x = temp_pts5(i,2);
    yy = rad-y;
    if (x>=rad), xx = x-rad; theta5(i) = atan2d(yy,xx);
    else xx = rad-x; theta5(i) = atan2d(yy,xx); theta5(i) = 180-theta5(i); end
    radius5(i) = ((xx)^2+(yy)^2)^.5;
end
phi5 = repmat(angles(2),1,length(theta5));
temp_pts5fin = overlay_rot(radius5,theta5,phi5,origin);
reduced_5fin = fft_reconstruct(fftshift(fft2(temp_img5)),temp_pts5fin);
thresh = thresh - .1;
end

% Determine the union of all detected peaks after rotations and keep only
% those preserved.
reduced_fft = reduced_2fin.*reduced_3fin.*reduced_5fin;

if show
figure(1);
subplot(221); fftshow(reduced_2fin)
subplot(222); fftshow(reduced_3fin)
subplot(223); fftshow(reduced_5fin)
subplot(224); fftshow(reduced_fft)
end
end
```

./AppendixB/src/fft_rotate.m

```
function [pve,pve_heatmap,points] = fft_scan(image,step,fact,res,viz,angle)
%  Input variable is M by N image and a box win that is an scalar factor
%  of the image. The image is scanned sequentially with the box and
%  returns FFT transforms. The input image is iterateively scanned with a
%  non-overallping region of diminsions step*step. fact is the scaling
%  factor determining how many standard deviations above the mean to use
%  as a cutoff. res scale the resolution. viz is the toggle to display
%  output graphs. angle is a switch for turning on/off the fft_angle call
%  to calculate the polar coordinate of detected peaks.
```

```
% Check if image size is divisible by step
[M N] = size(image);
if(mod(M*N,step))
    fprintf('Image not divisible by input box size\n');
    return;
end

% Set default inputs if not passed by user
if nargin < 6, angle=0; elseif nargin < 5, viz = 1; angle = 0;
elseif nargin < 3,  viz = 0; angle=0; res = 0;
elseif nargin < 3, viz = 0; angle=0; res = 0; fact = 1;
end

if angle, points = struct('pts',{},'theta',{}); end

xx = step;
yy = step;

pve = zeros(M/step*N/step,1);
pve_heatmap = zeros(size(image));
if viz, figure(101),imshow(image,[1 255]); end
n = 1;

% Iterate through the PVE matrix and create a heatmap of equal
    size to the
% image.
for j = 1:yy:N
    for i = 1:xx:M
        temp_segment = image(j:j+yy-1,i:i+xx-1);
        %
        [pve(n),img_points] = fft_pve(temp_segment,fact,0,res);
        % if the angle switch was passed
        if (angle),
            if pve(n)
                temp_fft = fftshift(fft2(double(temp_segment)));
                points(n).pts = img_points;
                % Record angles and radii
                [points(n).theta, points(n).radius] = ...
                    fft_angle(temp_fft,img_points);
            % If the PVE is 0 (i.e. no pattern detected) return 0
            else points(n).pts = 0;
                points(n).theta = 0;
                points(n).radius = 0;
            end
        elseif ~angle, points(n) = size(img_points,1);
        end
        % Record the PVE for that segment
        pve_heatmap(j:j+yy-1,i:i+xx-1) = pve(n);
        % Iterate ocounter
        n = n+1;
        % If user wants to visualize, display the corresponding
    region
        % being scanned
```

```
        if viz, figure(101); hold on;
            rectangle('Position',[i j xx-1 yy-1],'EdgeColor','r');
        end;
    end;
end;

% Reshape from an M*N/(step^2) x 1 vector to an M/step x N/step matrix
pve = reshape(pve,M/step,N/step);
```

./AppendixB/src/fft_scan.m

```
function fftshow(f,type)
% Usage: FFTSHOW(F,TYPE)
%
% Displays the fft matrix F using imshow, where TYPE must be one of
% 'abs' or 'log'. If TYPE='abs', then then abs(f) is displayed; if
% TYPE='log' then log(1+abs(f)) is displayed. If TYPE is omitted, then
% 'log' is chosen as a default.
%
% Example:
% c=imread('cameraman.tif');
% cf=fftshift(fft2(c));
% fftshow(cf,'abs')
%
if nargin<2,
type='log';
end
if (type=='log')
fl = log(1+abs(f));
fm = max(fl(:));
imshow(im2uint8(fl/fm))
elseif (type=='abs')
fa=abs(f);
fm=max(fa(:));
imshow(fa/fm)
else
error('TYPE must be abs or log.');
end;
```

./AppendixB/src/fftshow.m

B.9 References

[1] Z. A. Almsherqi, C. S. McLachlan, P. Mossop, K. Knoops, and Y. Deng. Direct template matching reveals a host subcellular membrane gyroid cubic structure that is associated with SARS virus. *Redox Report*, 10:167–171, 2005.

[2] Z. A. Almsherqi, S. D. Kohlwein, and Y. Deng. Cubic membranes: a legend beyond the Flatland of cell membrane organization. *Journal of Cell Biology*, 173:839–844, 2006.

[3] R. G. W. Anderson, L. Orci, M. S. Brown, L. M. Garciasegura, and J. L. Goldstein. Ultrastructural Analysis of Crystalloid Endo-plasmic-Reticulum in Ut-1 Cells and Its Disappearance in Response to Cholesterol. *Journal of Cell Science*, 63:1–20, 1983.

[4] J. Baik and G. R. Rosania. Molecular Imaging of Intracellular Drug-Membrane Aggregate Formation. *Molecular Pharmaceutics*, 8:1742–1749, 2011.

[5] J. Baik and G. R. Rosania. Macrophages sequester clofazimine in an intracellular liquid crystal-like supramolecular organization. *PLoS One*, 7, 2012.

[6] U. Bangert, M. H. Gass, A. L. Bleloch, R. R. Nair, and A. K. Geim. Manifestation of ripples in free-standing graphene in lattice im-ages obtained in an aberration-corrected scanning transmission electron microscope. *Physica Status Solidi a-Applications and Materials Science*, 206:1117–112, 2009.

[7] Chin, Luskey D. J., K. L., R. G. W. Anderson, J. R. Faust, J. L. Goldstein, and M. S. Brown. Appearance of Crystalloid Endoplas-mic-Reticulum in Compactin-Resistant Chinese-Hamster Cells with a 500-Fold Increase in 3-Hydroxy-3-Methylglutaryl-Coenzyme-a Reductase. *Proceedings of the National Academy of Sciences of the United States of America-Biological Sciences*, 79:1185–1189, 1982.

[8] Y. R. Deng and M. Mieczkowski. Three-dimensional periodic cubic membrane structure in the mitochondria of amoebae Chaos car-olinensis. *Protoplasma*, 203:16–25, 1988.

[9] T. Fujita and M. W. Chen. Characteristic length scale of bicontinuous nanoporous structure by fast Fourier transform. *Japanese Journal of Applied Physics*, 47:1161–1163, 2008.

[10] F. C. Gong, T. H. Giddings, J. B. Meehl, L. A. Staehelin, and D. W. Galbraith. Z-membranes: Artificial organelles for overex-pressing recombinant integral membrane proteins. *Proceedings of the National Academy of Sciences of the United States of America*, 93:2219–2223, 1996.

[11] C. Knupp, C. Pinali, P. M. Munro, H. E. Gruber, M. J. Sherratt, C. Baldock, and J. M. Squire. Structural correlation between collagen VI microfibrils and collagen VI banded aggregates. *Journal of structural biology*, 154:312–326, 2006.

[12] Tomas Landh. From entangled membranes to eclectic morphologies: cubic membranes as subcellular space organizers. *FEBS Lett.*, 369:13–17, 1995.

[13] V. Lucic, F. Forster, and W Baumeister. Structural studies by electron tomography: from cells to molecules. *Annu. Rev. Bio-chem.*, 74:833–865, 2005.

[14] A. V. Pais, S. Pereira, I. Garg, J. Stephen, M. Antony, and Y. K. Inchara. Intra-abdominal, crystal-storing histiocytosis due to clofazimine in a patient with lepromatous leprosy and concurrent carcinoma of the colon. *Leprosy Review*, 75: 171–176, 2004.

[15] G. Q. Ren and Y. C. Xing. Temperature-induced restructuring of self-assembled PtPd nanoparticle superlattices. *Nanotechnology*, 20:46, 2006.

[16] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:623–656, 1948.

[17] S. Smith and G. Blobel. Colocalization of Vertebrate-Lamin-B and Lamin-B-Receptor (Lbr) in Nuclear Envelopes and in Lbr-Induced Membrane Stacks of the Yeast Saccharomyces-Cerevisiae. *Proceedings of the National Academy of Sciences of the United States of America*, 94:10124–10128, 1991.

[18] E. L. Snapp, R. S. Hegde, M. Francolini, F. Lombardo, S. Colombo, E. Pedrazzini, N. Borgese, and J. Lippincott-Schwartz. Formation of stacked ER cisternae by low affinity protein interactions. *Journal of Cell Biology*, 163: 257–269, 2003.

[19] S. Sukpanichnant, N. S. Hargrove, U. Kachintorn, S. Manatsathit, T. Chanchairujira, N. Siritanaratkul, T. Akaraviputh, and K. Thakerngpol. Clofazimine-induced crystal-storing histiocytosis producing chronic abdominal pain in a leprosy patient. *The American journal of surgical pathology*, 24:129–135, 2009.

[20] R. Wright, M. Basson, L. D'Ari, and J Rine. Increased amounts of HMG-CoA reductase induce "karmellae": a proliferation of stacked membrane pairs surrounding the yeast nucleus. *J. Cell Biol.*, 107:101–114, 1998.