# A MULTI-COMPARTMENT PHARMACOKINETIC MODEL OF DOCETAXEL

**ALISSA KUHN**
**Bachelor of Science, University of Lethbridge, 2017**

A thesis submitted
in partial fulfilment of the requirements for the degree of

**MASTER OF SCIENCE**

in

**PHYSICS**

Department of Physics and Astronomy
University of Lethbridge
LETHBRIDGE, ALBERTA, CANADA

A MULTI-COMPARTMENT PHARMACOKINETIC MODEL OF DOCETAXEL

ALISSA KUHN

Date of Defence: December 8, 2023

| | | |
|---|---|---|
| Dr. Kenneth Vos<br>Thesis Supervisor | Associate Professor | Ph.D. |
| Dr. Locke D. Spencer<br>Thesis Examination Committee Member | Associate Professor | Ph.D. |
| Dr. Arundhati Dasgupta<br>Thesis Examination Committee Member | Associate Professor | Ph.D. |
| Dr. Mark Walton<br>Chair, Thesis Examination Committee | Professor | Ph.D. |

# Dedication

To the Swedish Fish which
offered endless support to me
through this journey.

# Abstract

Docetaxel is a clinically active chemotherapeutic agent commonly used in the treatment of solid tumors. It is administered within a micelle encapsulation, polysorbate 80, via intravenous infusion. Docetaxel is known to have a high degree of interpatient variability in its pharmacokinetic behaviour. In this work, intensive, quantitative, compartmental pharmacokinetic models of docetaxel and its vehicle polysorbate 80 are developed. These models introduce both saturable kinetics and power-law relationships to the kinetic behaviour of these molecules. When fit to clinical data available in the literature by minimizing the weighted percentage variance these models out perform traditional linear models. A three-compartment model of docetaxel with both saturable and fractal effects is shown to accurately describe docetaxel pharmacokinetics. From this model pharmacokinetic metrics such as the maximum concentration, the area under the curve, and the half-life are derived. The sensitivity of this model's parameters to interpatient variability is also investigated.

# Acknowledgements

I would like to express deep gratitude to the support of my supervisor Dr. Ken Vos. From my undergrad to my masters Ken has provided unwavering and enthusiastic support to my pursuits in physics. His love of teaching is infectious and I hope that I was able to bring some of that enthusiasm to the students I taught while completing this thesis.

I would also like to thank the members of my committee, Dr. Locke Spencer and Dr. Arundhati Dasgupta for their feedback and support as well as Dr. David Kaminski who retired and left my committee before I could finish but provided valuable feedback on my initial drafts.

Thank you as well to my parents who provided me a great deal of support during my undergrad and to my sisters who kept my spirits high with photos of Hank and Loki.

A special shout out to Chris Perlette who helped make sure all my data was backed up, Geneviève May who ensured I was a proper local after my move to the Crowsnest Pass, Dakota Duffy for his excellent company and employment opportunities, and all the other friends who helped me with assignments, went climbing, hiking, and camping with me, or otherwise ensured that I found time for fun.

Most importantly I would like to thank my partner, Kyle, for his endless patience through this journey. I look forward having more time to spend with you out in the Slide.

Thank you all.

# Contents

# List of Tables

# List of Figures

# List of Abbreviations

**DOC** docetaxel

**P80** polysorbate 80

**AUC** Area Under the Curve

**PK** pharmacokinetic

**PD** pharmacodynamic

**MAP** microtubule associated protein

**IV** intravenous

**AIC** Akaike Information Criterion

**RK** Runge-Kutta

**BSA** Body Surface Area

# List of Symbols

| | |
|---|---|
| $t$ | Time |
| $X_{d,c}$ | Total amount of a molecule $d$ in compartment $c$ |
| $C_{d,c}$ | Concentration of a molecule $d$ in compartment $c$ |
| $V_d$ | Volume of Distribution |
| $C_{max}$ | Maximal concentration achieved after a drug has been administered |
| $T_{max}$ | The time corresponding to the maximal concentration achieved after a drug has been administered |
| AUC | Area under curve |
| $C_\ell$ | Clearance constant |
| $V$ | Volume |
| $\nu$ | Michaelis-Menten reaction rate |
| $K_M$ | Michaelis constant |
| $t_{1/2}$ | Half-life |
| $t_{1/2,\alpha}$ | Distribution half-life |
| $t_{1/2,\beta}$ | Elimination half-life |
| $t_{1/2,\gamma}$ | Terminal half-life |
| $I$ | Infusion |
| $D$ | Dose |
| $S_p$ | Weighted percentage variance |
| $N_p$ | Normalization factor |
| $n_p$ | Number of patients |
| $n_d$ | Number of data points |

$\mathbf{P}$ Parameter space

$\mathbf{e}$ Search vector

$AIC$ Akaike information criterion

$\eta$ Number of parameters associated with a model

$L$ Maximum likelihood function

$\Delta$ $AIC$ difference

$\bar{k}$ Rate constant

$\Gamma$ Saturation constant

$A$ Fractal/process order (kinetic rate)

$B$ Fractal/process order (saturable rate)

$\delta$ Conservation of mass

# Notes on Format

The following style conventions were selected:

**Reference** Parenthetic citation was utilized for references to external works by other authors (e.g., [1-3,7] refers to citations 1,2,3 and 7). Internal references to different chapters/appendices and sections are denoted by *chapter.appendix.section.subsection*. Figure, table and equation numbers are denoted by *chapter.appendix.value*, where *value* is the incremental numerical value of the figure, table or equation.

**Clinical Data** In some of the clinical studies used for this research, multiple courses of treatment are considered within a single study. Where applicable, a letter is included after the reference (e.g. Author et al. [1]a) to differentiate between each dosing regimen. The reference and corresponding doses are provided in §3.

**Equations** Units are provided for every relevant value. The units of values presented within tables can typically be found below the table. If the units are not provided, the equation is unitless. Units are provided in the standard International System of Units (SI).

**Text** The `typewriter` font was used to denote the C++ functions used in this research. The code of each function is available in §C.

# Chapter 1

# Introduction

*"The greatest challenge to any thinker is stating the problem in a way that will allow a solution"*

– Bertrand Russell

This thesis aims to develop an intensive pharmacokinetic (PK) model of the chemotherapeutic agent docetaxel (DOC). In this chapter the fundamental principles of PKs and it's applications in drug modelling are outlined. Then, information regarding the classification and impacts of cancers which are relevant to PKs are touched on. Finally, the behaviour of the principal molecule of interest, DOC is discussed. The content of this chapter aims to provide the necessary background to understand the complex, multi-disciplinary field of PKs and provide motivation for the development of a novel model of DOC.

## 1.1 Pharmacokinetics

One of the most clinically important aspects of a drug will be information about its PK properties. PK refers literally to the movement (*kinetics*), of a drug (*pharmaco*). It is the description of what happens to a drug once it enters the body. This is in contrast to pharmacodynamic (PD) which describes the action of the drug on the body. An understanding of both PK and PD is critical for achieving positive clinical outcomes. The study of both PK and PD in combination is often referred to as pharmacometrics.

The core principles of PK are summarized by the acronym **ADME** which stands for:

- **A**bsorption
- **D**istribution
- **M**etabolism
- **E**limination

These four steps encompass how a drug will be administered, arrive at its site of action, and ultimately be removed from the body [1].

### 1.1.1 Absorption

The drug first enters the body through the process of absorption. There are two routes by which a drug can be administered, parenterally (intravenous, intramuscular, etc.), and enterally (oral, topical, etc.). The fraction of the dose which enters the circulatory system is known as a drug's bioavailability. The larger the bioavailability is, the larger the active effect the drug may have. Drugs given intravenously will have a bioavailability of 100%. This includes encapsulated formulations of a drug.

### 1.1.2 Distribution

Once a drug is absorbed it will be distributed to the body's tissues by blood flow. The physicochemical properties of a drug such as size, solubility, etc. play a large role in determining how it will bind to tissues. The apparent volume of distribution, $V_d$, reflects the extent to which this distribution occurs and is given by

$$V_d = \frac{X}{C} \tag{1.1}$$

where $X$ is the total amount of drug in the body and $C$ is the concentration within the blood. A large $V_d$ indicates that the drug has been highly distributed through the body's tissues. Conversely, a small $V_d$ indicates that the drug remains primarily within the blood plasma.

### 1.1.3 Metabolism

Metabolism is the process by which a drug is transformed through a chemical reaction. The liver is responsible for metabolising the vast majority of drugs, though it can occur in other cells as well. Hepatic metabolism is achieved through two key processes. Phase I reactions prepare the drug to undergo a phase II reaction and the phase II reaction produces a metabolite which is more easily soluble in water and therefore easier for the kidneys to remove. These metabolic reactions are catalysed by enzymes from the cytochrome P450 family. The metabolites produced by phase II processes are typically inactive while phase I reactions may produce metabolites that are themselves clinically active molecules. Pro-drugs are a special class of drug which has an inactive parent drug that becomes active once metabolised.

### 1.1.4 Elimination

Clearance is the rate at which a drug is irreversibly eliminated from the body. Most drugs will be eliminated through either the kidneys (renal clearance) or the liver (biliary clearance). Drugs may be eliminated directly via faeces, urine, sweat, etc., or be metabolised within the liver. Drug metabolites will then be eliminated via a direct route.

The total clearance of a drug, i.e., the clearance due to all routes of elimination, will determine the dosage required to maintain a given concentration within the blood. This can be highly dependent on factors such as patient health and drug interactions. A strong understanding of a drug's clearance is therefore necessary for creating dosage plans that will yield appropriate concentrations for achieving desired treatment outcomes.

### 1.1.5 PK Metrics

In addition to the principles of ADME, there are a collection of metrics that are of particular importance to those studying the concentration-time profiles of a drug. Once obtained, the metrics can be used to inform the optimal concentrations that will achieve desired clinical effects.

$C_{Max}$ is the maximal concentration achieved after the administration of the drug. It is the point where the rate of elimination overtakes the rate of absorption. The time it occurs at is known as $T_{Max}$. The $C_{Max}$ of a given drug will be limited by it's toxicity where the maximal concentration may not be so large as to produce severely negative effects [2].

The Area Under the Curve (AUC) is the total integrated area underneath a plasma drug concentration-time curve. It is a measurement of the extent of absorption and reflects the exposure of the patient to a drug, and in turn the drugs efficacy. The $AUC$ is typically calculated directly from concentration-time data using the trapezoidal rule though some other methods of increased complexity exist [2]. Often, the $AUC$ will not be extrapolated beyond the final concentration data point. Instead it will be calculated over a specific time range. For $DOC$ it is most frequently calculated for 24 hours post infusion.

The remaining metrics are more complex than those outlined above because their definitions depend on the specific PKs of the drug. The majority of drugs studied clinically are thought to follow linear PKs. In these models, the concentration $C$ of an intravenously infused drug may be described as,

$$\frac{dC}{dt} = -\frac{C_\ell}{V_d}C, \tag{1.2}$$

where $C_\ell$ is the drug clearance, i.e. the rate at which the drug is removed from the body. The solution is

$$C(t) = \frac{dose}{V_d}e^{-\frac{C_\ell}{V_d}t}. \tag{1.3}$$

where we have used $C_{max} = dose/V_d$. These kinetics imply an important relationship between the dose, clearance and $AUC$,

$$dose = C_\ell \, AUC. \tag{1.4}$$

The linear relationship between dose and $AUC$ is why drugs, which behave according to

Eqn. 1.2, are said to exhibit linear PKs [3].

The final metric we will consider is half-life $t_{1/2}$. Half-life is used to inform the frequency of dosing schedules as it indicates when the drug concentration will have dipped below clinically active levels. In the case on linear PKs, Eqn. 1.2 implies a half-life of

$$t_{1/2} = \frac{\ln(2)V_d}{C_\ell}. \tag{1.5}$$

Linear approaches to studying drug distribution as described above are limited but can be useful in many contexts. However, these models fail to account for the full range of effects which physiological interactions may have on the kinetics of a drug. For example, when considering the binding of a drug to a receptor, non-linear models related to the Michaelis-Menten equation, should be used [4]. Saturation kinetics which include capacity-limited elimination can be more accurate in cases where drugs decline multiphasically. This typically occurs with a maximum of three distinguishable phases which are each associated with their own distinct half-life. These are denoted as the $\alpha$-half-life which is associated with the initial distribution of the drug into peripheral tissues, the $\beta$-half-life, and the $\gamma$-half-life which is associated with the terminal elimination phase. A more complete treatment of how these half-lives, as well as the other metrics, are determined for nonlinear compartmental models will be given in §2.3 after the relevant concepts have been appropriately developed.

## 1.2 PK Modelling

Prior to 1937 the aspects of pharmacology focused on by physicians and physiologists was intensely practical, with aspects such as drug administration, mechanism of action, and effect being the main concerns [5]. In 1937 Torsten Teorell, now known as the father of PK, recognised that very little work had been done on the time relation of drug action, the kinetics. In his two seminal papers [6, 7] he aimed to lay out a framework of mathematical

relations which could be used to describe the kinetics of distribution for substances within the body. To do so he developed a simplified model of the human body which considered the blood, elimination organs (kidneys, liver, etc.), and other tissues as separate compartments of a specific volume. The drug, administered subcutaneously, could then flow between or out of the compartments as determined by a rate constant $k$ which was specific to that compartment. Teorell used this model to derive differential equations and compute the time course of the amount of drug in each compartment as a percentage of the total dose. Most importantly, Teorell recognised that the rate of absorption, or changes to such, could heavily influence the magnitude and duration of the concentration-time curves within the blood and tissues which would have a strong bearing on the efficacy of treatments.

Unfortunately, Teorell's work was largely forgotten until the 1960s when the development of bioanalytic techniques improved measurements of concentration in plasma and tissues. PK models became most useful once computational power was advanced enough to support the complex nature of obtaining more precise numerical solutions required by these complex models. Since the field of study is so young standardized methods for model selection have not yet been developed and many of the techniques used for model evaluation are borrowed from related fields [8]. Some of the considerations which are commonly applied when developing PK models are outlined below.

To build any model requires inputs and outputs where the input perturbs the system in a measurable way. In the case of a PK model the input is very often an administered dose of a drug and the output is the blood sample. The concentration data obtained from blood samples form the foundation against which the model is tested.

The development of an effective PK model requires an understanding of when a model is appropriate to use, how to select a model, and how to evaluate a model's goodness-of-fit. When done successfully, this process allows for better comparison, evaluation, and prediction of pharmaceutical compounds by parametrizing the concentration-time profile of drugs. Combining an accurate PK model with PD modelling leads to a better understand-

ing of the dose-exposure-response relationship, and improved treatment efficacy. When both PK and PD principles are accounted for in a model it is pharmacometric. Fundamentally, the goal of PK models is to relate a dose to a drugs concentration and predict it's concentration-time profile as a curve.

There are two types of PK models:

(i) Models of data (Empirical models): Used when the underlying physical process is not well understood.

(ii) Models of systems (Mechanistic models): Built off of physical/physiological principles.

Empirical models are developed based on experimentally derived data and thus will be subject not just to the system being analysed but also to the error in the measurements used to build them. They will, therefore, generally include both a structural model paired with a statistical model that describes the associated error of the former. Modelling extends beyond simply characterizing data though. A good model will allow predictions to be made and will be capable of simulating relevant scenarios. This has obvious benefits in pharmacology as it allows a variety of changes to the way a drug is administered to be considered without needing to test those changes on real patients. One of the most important advantages of mathematical models in this field is reducing the impact on living animals and humans.

PK models can be categorized based on (i) having time dependent or independent parameters (time variant v. time invariant), (ii) being non-probabilistic or probabilistic in nature (deterministic v. stochastic), (iii) following steady-state or time dependent solutions (static v. dynamic), (iv) being compartmentalized or spatially diffuse (lumped v. distributed), (v) it's linearity (linear v. nonlinear), and (vi) it's continuity (continuous v. discrete) [9]. These categories can all be combined within a model. Our model includes time invariant parameters (stationary), is deterministic, dynamic, lumped (homogeneous compartments), nonlinear, and continuous.

The general process taken to develop a PK model will include the following steps:

- Look at the data
- Formulate a sensible model
- Fit the model to the data
- Diagnostically check the adequacy of the fit
- Present the results and conclusions

A useful model should promote retrodiction (recall of experiment), be predictive, and increase understanding. It should also be useful, logically consistent, and validated. Occam's razor should always be considered: is the increase in predictability worth the increase in complexity? This question will be explored further in §2.6.

Modelling biological systems is a hugely complex undertaking and subject to many difficult to track variables that can impact the data collection process. It is impractical to set out with hopes to develop a 'true' model of any biological system. Instead the goal is to produce a reasonable approximation, a simplification, that can be wielded usefully for practical purposes.

### 1.2.1 Rate Laws

The rate at which PK processes occur are dependent on the amount of molecule $X$, and given by zeroth, first, second or some mixed order rate laws. Each rate and its solution are given separately below where $k^{(n)}$ is the $n$th-order rate constant, and $\Delta t = t - t_0$ is the elapsed time.

**Zeroth Order**

The zeroth order rate equation is given by,

$$\frac{dX}{dt} = -k^{(0)}, \tag{1.6}$$

which, independent of the amount of molecule, has the solution

$$X(t) = -k^{(0)}\Delta t + X(t_0). \tag{1.7}$$

This reaction rate occurs when enzyme saturation is reached. An IV infusion can also be approximated as a zeroth order process.

**First Order**

The first order rate equation is linear in $X$,

$$\frac{dX}{dt} = -k^{(1)}X, \tag{1.8}$$

with solution,

$$X(t) = X(t_0)e^{-k^{(1)}\Delta t}. \tag{1.9}$$

This reaction rate is obeyed by most compounds at low concentrations.

**Second Order**

The second order rate equation is nonlinear in $X$,

$$\frac{dX}{dt} = -k^{(2)}X^2, \tag{1.10}$$

with solution

$$X(t) = \frac{X(t_0)}{1 + k^{(2)}\Delta t X(t_0)}. \tag{1.11}$$

### 1.2.2 Michaelis-Menten Kinetics

At their simplest, biochemical reactions take the form

$$[S] + [E] \underset{k_r}{\overset{k_f}{\rightleftharpoons}} [ES] \overset{k_{cat}}{\Longrightarrow} [E] + [P] + [BYPRODUCT], \tag{1.12}$$

where a substrate $[S]$ forms a complex $[ES]$ with an enzyme $[E]$ which then breaks down, releasing the product $[P]$, the enzyme, and the reactions byproduct. Each process is associ-

ated with is own rate where $k_f$ is the forward rate constant, $k_r$ is the reverse rate constant and $k_{cat}$ is the catalytic rate constant.

The metabolism of DOC may be characterized by this reaction:

$$\text{DOC} + \text{CYP3A4} \leftrightharpoons \text{CYP} - \text{DOC} \Rightarrow \text{CYP3A4} + \text{M4} + \text{BYPRODUCT}. \tag{1.13}$$

The reaction rate or velocity $v$ of this process was first derived by Leonor Michaelis and Maud Menten in 1913 [10]. If the rate limiting step of this process is assumed to be the enzyme-substrate breakdown the reaction rate is given by a first-order process,

$$v = \frac{d[P]}{dt} = k_{cat}[ES]. \tag{1.14}$$

As the reaction unfolds the concentration of the enzyme-substrate complex will increase quickly until a steady state is reached when the rate of formation and dissociation of the complex reaches an equilibrium. This is known as the law of mass action and implies that

$$\frac{d[ES]}{dt} = k_f[E][S] - (k_{cat} + k_r)[ES] = 0, \tag{1.15}$$

and therefore,

$$k_f[E][S] = (k_{cat} + k_r)[ES]. \tag{1.16}$$

Rearranging this term we find

$$K_M = \frac{[E][S]}{[ES]} = \frac{k_{cat} + k_r}{k_f}, \tag{1.17}$$

where $K_M$ is the Michaelis constant.

We may also note that the total amount of enzyme $[E]_0$ is conserved through the reaction

and will therefore remain constant such that

$$[E]_0 = [E] + [ES]. \tag{1.18}$$

Combining these relations we find the Michaelis-Menten equation which describes the reaction rate of simple enzymatic processes

$$v = \frac{k_{cat}[S][E]_0}{K_M + [S]}. \tag{1.19}$$

In this context the Michaelis constant $K_M$ corresponds to the concentration of the substrate when the reaction rate is at half of its maximum value. It is also inversely proportional to the binding affinity between [S] and [E].

At high concentrations, $[S] \gg K_M$, a maximum reaction rate is met,

$$v_{MAX} = k_{cat}[E]_0. \tag{1.20}$$

The reaction follows zeroth-order kinetics as discussed in §1.2.1.

For low concentrations, $[S] \ll K_M$, the behaviour is roughly linear, and the reaction rate, approximately, becomes,

$$v = \frac{k_{cat}[S][E]_0}{K_M} = \frac{v_{MAX}}{K_M}[S]. \tag{1.21}$$

The reaction rate obeys first-order kinetics. The Michaelis-Menten model is a mixed-order rate equation with the kinetics switching between first and zeroth order processes. It can be modified to describe the rate at which the amount of molecule $X$ changes within a compartment

$$\frac{dX}{dt} = \frac{\nu_{MAX}X}{K_M + X}. \tag{1.22}$$

### 1.2.3 Fractal Kinetics

Heterogeneous media further complicates the kinetics which describe biomolecular compounds. Within such media the rate constant becomes time dependant such that

$$k(t) = k_0 t^{\mu} \tag{1.23}$$

where $\mu$ is a fractal exponent related to the fractal dimension of the media [11]. This phenomena is directly relevant to PK modelling [12]. Another taxane (see §1.4), paclitaxel, has previously been show to display fractal kinetic behaviour [13, 14].

Under steady-state conditions the reaction rate of [S] is necessarily time independent and consequently becomes dependent on an anomalous reaction order $W$,

$$\frac{d[S]}{dt} = k[S]^W \tag{1.24}$$

where $W = 1 + (1-\mu)^{-1}$. Thus, Eqn. 1.15 becomes,

$$\frac{d[ES]}{dt} = k_f[E][S]^W - (k_{cat} + k_r)[ES] = 0. \tag{1.25}$$

Completing a derivation similar to the one shown above in §1.2.2 Eqn. 1.25 can be rearranged to find,

$$v = \frac{k_{cat}[S]^W[E]_0}{K_M + [S]^W}. \tag{1.26}$$

This result can be slightly modified to describe the rate at which the amount of molecule $X$ changes within a compartment

$$\frac{dX}{dt} = \frac{\nu_{MAX} X^a}{K_M + X^b} \tag{1.27}$$

where $a$ and $b$ are unique reaction orders. The implicit solution of Eqn. 1.27 is

$$K_M \frac{X^{1-a} - X_0^{1-a}}{1-a} + \frac{X^{1+b-a} - X_0^{1+b-a}}{1+b-a} = -\nu_{max}(t - t_0), \tag{1.28}$$

when $a \neq 1$ and $a \neq 1+b$. When $a = 1$ the solution becomes

$$K_M \ln(\frac{X}{X_0}) + \frac{X^{1+b-a} - X_0^{1+b-a}}{1+b-a} = -\nu_{max}(t - t_0), \tag{1.29}$$

and when $a = 1+b$ the solution becomes

$$K_M \frac{X^{1-a} - X_0^{1-a}}{1-a} + \ln(\frac{X}{X_0}) = -\nu_{max}(t - t_0), \tag{1.30}$$

## 1.3 Cancer

### 1.3.1 The Burden of Cancer

19.3 million people worldwide were diagnosed with cancer in 2020 and 10.0 million died from cancer related illness [15]. Nearly half of all Canadians (43%) will be diagnosed with cancer in their life time and, although prognosis has improved, cancer remains the leading cause of death in Canada [16]. As the population continues to grow and age, the number of new cases and deaths has increased as well. Treatment is costly and imposes a large burden on the Canadian health care system. The cost of treating cancer in Canada rose from CAD$ 2.9 billion in 2005 to CAD$ 7.5 billion in 2012 [17]. More recent studies have attempted to include the personal and societal costs faced by Canadians in these figures. Accounting for these costs along with the direct health systems costs increases Canada's economic burden to an estimated CAN$ 26.2 billion in 2021 with 30% of the cost borne by patients and their families [18]. Furthermore, the research and development of the pharmaceuticals used in cancer treatment requires huge financial and time investments. It can take upwards of a decade for a drug to reach the market [19] and cost over USD$ 1.8 billion [20].

PK modelling of antineoplastic agents can serve as an economically efficient way to improve the quality of treatment for patients receiving chemotherapy.

The primary drug used in this study, DOC, has been shown to be effective against a

wide range of cancers including breast cancer, non-small cell lung cancer, ovarian cancer, prostate cancer, and squamous cell carcinoma of the head and neck [21, 22].

### 1.3.2 Causes of Cancer

Cancers are a result of mutations to the mechanisms that prevent genetic errors from being copied into new cells [23]. The exact cause of these mutations is not known, however, lifestyle choices such as tobacco use, alcohol consumption, diet/obesity as well as other factors such as occupational exposure to carcinogens, exposure to radiation or pollution, and some viral infections (hepatitis B virus, hepatitis C virus, human papillomavirus) among others have been linked to an increase in risk of cancer development [24].

Cells have multiple ways of resisting these errors including checks that DNA has been correctly replicated, mechanisms to prevent cell replication if DNA is damaged, and apoptosis [25]. The genes that produce the proteins that regulate apoptosis are known as proto-oncogenes. Mutations to these genes can cause them to stop functioning correctly at which point they become oncogenes. Oncogenes can be activated via mutations, gene amplification, or chromosome rearrangements [1]. Mutations of the tumour suppressor gene *TP53* is one of the most frequently seen in cancers. *TP53* encodes the protein p53 which plays a vital role in maintaining the integrity of the cell by repressing proliferative signalling enhancing the effects of growth suppressors, sensitizing cells to apoptosis, promoting genomic stability, and other similar activities. Each of these functions actively obstruct tumour-promoting cellular activities [24]. Mutations causing inactivation of p53 promotes cell growth and predisposes the cell to cancer. This mutation is the most frequent cause of cancer and has been found in all types. Most commonly, it is the source of ovary and rectal cancers [24].

### 1.3.3 Classification of Cancer

Cancers are classified in multiple ways based on:

- Histology: The type of affected tissue.

- Grading: The abnormality of the cells.
- Staging: The scope of the disease.

There are six major categories that cancer can be categorized by based on histology. These are listed, along with the type of affected tissue, in Tbl. 1.1. Each of these categories can be further broken down into the specific affected cell types.

Table 1.1: Histological classification of cancers.

| Types of Cancer | Affected Tissue/Systems |
|---|---|
| Carcinoma | Epitheial tissue |
| Sarcoma | Supportive and connective tissues |
| Myeloma | Bone marrow plasma |
| Leukaemia | Haemopoietic cells |
| Lymphoma | Glands/nodes of the lymphatic system |
| Mixed | Multiple tissue systems |

There are four grades of cancer which are defined by how differentiated i.e., normal, the cells are. These are:

- Grade 1: Well-differentiated
- Grade 2: Moderately differentiated
- Grade 3: Poorly differentiated
- Grade 4: Undifferentiated

The appearance and behaviour of the cells are compared to normal tissues to determine the grade. Low grade cancers are less dangerous and have abnormal cells that still behave similar to the affected tissues. Higher grade cancers are more aggressive and will look and be arranged differently than normal. The grade of a cancer is determined by biopsy of the affected tissues.

The final way to classify cancer is staging. Staging is defined by the extent to which the cancer has spread. These are:

- Stage 0: Cancer in situ
- Stage I: Cancer limited to the tissue of origin, evidence of tumour growth
- Stage II: Limited local spread of cancerous cells
- Stage III: Extensive local and regional spread
- Stage IV: Distant metastasis

Staging and grading are most useful in determining the prognosis of a patient where a higher grade/stage indicates poorer prognosis. The category that a given cancer falls under will also influence the choice of treatment. Benign tumours are typically well differentiated (low grade), grow slowly and are unable to metastasize (stage 0-I) while malignant tumours grow rapidly and can develop into higher grades and stages [1].

### 1.3.4 PK Resistance

Cancer cells may develop resistance to chemotherapy treatment. This resistance can be either intrinsic or acquired but both cases result in reduced efficacy of treatment [26]. The most common mechanisms leading to PK drug resistance are outlined below.

A large fraction (95%) of anti-cancer drugs given intravenously will accumulate in other organs such as the liver or be cleared by the kidneys [27]. The remaining drug can be taken up by the tumour via the Enhanced Permeability and Retention effect. However, distribution for a drug within the tumour may be limited by pressure gradients in the interstitial fluid which promotes flow to the peripheries of the tumour, preventing penetration of the chemotherapeutic agent [28]. This change in fluid dynamics is counterintuitive as increased vascularization (angiogenesis) might suggest increased permeability of the tumour [1]. However, the core of tumours becomes avascular as rapidly dividing cells compress blood and lymphatic vessels. This raises the interstitial fluid pressure compared to healthy tissues and prevents deeper penetration of the drug [29].

Drug efflux pumps can also develop within cancer cells. Transporters on the cell membrane target xenobiotic compounds and push the drug out of the cell. This prevents the drug from accumulating within the target cells, reducing the active effect [1]. The amplification of drug efflux pumps may lead to multi-drug resistance. Multi-drug resistance can be mitigated by using drug encapsulation, including polysorbate 80 (P80) which is used with DOC [30].

To mitigate these effects most cancers are now treated using combination chemotherapies. This can be more effective so long as the drugs do not share the same toxicities (which would increase side effects), have different mechanisms of action, and are relevant to the patients condition [1]. DOC is used in combination with doxorubicin, cyclophosphamide, fluorouracil, capecitabine, platinum compounds/derivatives, and prednisone [21].

## 1.4 DOC

The drug under investigation in this study is DOC, an antineoplastic agent which disrupts the microtubular network in cells. DOC is a member of the taxane class of chemotherapeutic agents. It's molecular formula is $C_{43}H_{53}NO_{14}$ and it has a molecular weight of 807.88 g/mol [21]. The chemical structure of DOC is shown on the right in Fig. 1.1 along with the structure of the other major taxane, paclitaxel.

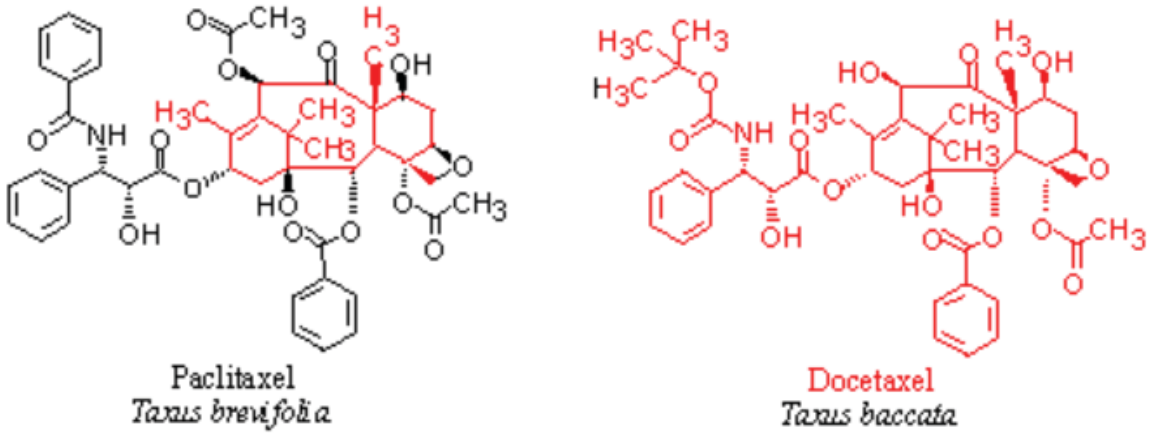



Figure 1.1: The chemical structure of taxanes.

The first taxane to be discovered was paclitaxel, in the 1970s. It was found as part of a National Cancer Institute program which tested thousands of plant-based compounds for anti-tumour activity [31]. It is derived from the bark of the Pacific yew tree (*Taxus brevifolia*). DOC is a semi-synthetic derivative of paclitaxel discovered in 1981 during a collaborative research project between Rhône-Poulenc and the Institut de Chimie des Substances Naturelles in France [32]. It was developed due to difficult production and limited supply of paclitaxel. "Docetaxel is produced semisynthetically from 10-deacetyl baccatin III, an inactive precursor which is then esterified with a synthetic side chain. An advantage of this procedure is that 10-deacetyl baccatin III can be extracted from the needles of the European yew (*Taxus baccata*), a renewable resource" [32]. Other taxanes have also shown anti-tumour activity. Cabazitaxel was approved by the FDA in 2010 and has been used to treat prostate cancer when DOC was ineffective [33]. Milataxel, tessetaxel, and ortataxel are recently discovered, novel, and orally bioavailable formulations currently being investigated for effectiveness against a variety of cancers [34].

DOC is sold under the brand name Taxotere ®, with other available variants being functionally the same. Pure DOC is extremely hydrophobic so it is given intravenously within an encapulation of P80 along with ethanol anhydrous. P80 is a nonionic surfactant composed of hydrophobic and hydrophilic moieties. This structure promotes the formation of micelles which solubilize DOC and allow it to be used clinically [35]. Oral corticosteroids are given alongside DOC to reduce symptoms caused by the encapsulation which include fluid retention and hypersensitive reactions [21].

### 1.4.1 Mechanism of Action

Microtubules are a class of macromolecular assembly and a major component in eukaryotic cytoskeletons [36]. They are built from protein units called tubulin which come in two forms, $\alpha$-tubulin and $\beta$-tubulin. These pair to form the heterodimeres that assemble into long cylindrical structures with diameters on the order of 10 nm [37]. These filamen-

tous structures play roles in a variety of cellular activities. They maintain the cell's shape and movement, facilitate cell signalling, and most importantly allow the cell to undertake division and mitosis [38, 39].

There are two main drug classes that target tubulin. Vinca alkaloids and taxanes. In contrast to the vinca alkaloids, which work by preventing microtubule assembly, the taxanes work by shifting the dynamic equilibrium between tubulin dimers and microtubules toward polymerisation [22]. The polymerisation of microtubules is controlled by two types of non-equilibrium dynamics. The first is known as dynamic instability. This process sees the ends of the tubules cycling through phases of growth and shortening. "Dynamic instability is characterized by four main variables: the rate of microtubule growth; the rate of shortening; the frequency of transition from the growth or paused state to shortening (this transition is called a 'catastrophe'); and the frequency of transition from shortening to growth or pause (called a 'rescue')" [39]. The second is known as treadmilling and is depicted in Fig. 1.2. Under this process the microtubules maintain their length by growing and shortening at opposite ends at the same rate. Tubulin subunits flow from the shortening end to the growing end, driven by a difference in the concentration of free tubulin along the microtubule. This behaviour is thought to be particularly important for mitosis [39].

These two behaviours are regulated by microtubule associated protein (MAP)s including MAPT, MAP2 and MAP4 [38]. These proteins are structural MAPs which cross-link, stabilize and suppress dynamics of microtubules at rates related to the MAPs concentration [40]. Any modifications to the rate constants of these processes will have large effects on the polymerisation dynamics and, in turn, the microtubule function. The suppression of spindle-microtubule dynamics brought on by the stabilization of tubulin in particular is the most important action of these drugs as it blocks mitosis at the metaphase-anaphase transition which triggers apoptosis [39].

The taxanes have a low affinity to free tubulin and will instead bind to the β-tubulin on the interior surface of the microtubules. This produces a stronger affect within the cell than

if free tubulin were the main target because the microtubules are affected directly. Each β-tubulin acts as an active binding site for the taxane molecule. For paxitaxel increased polymerisation of the microtubule is triggered at a 1:1 binding ratio to the β-tubulin sites along the microtubule while DOC binds at a 1.9-fold higher affinity and requires a 2.1-fold lower tubulin concentration [22]. Therefore, a greater amount of polymer will be formed by DOC compared to paxitaxel. The stronger action of DOC is thought to be in part due to higher cellular uptake and stronger action on tubulin. However, this enhanced action comes at the price of increased toxicity, and therefore DOC does not, by default, have a greater therapeutic index than paclitaxel. It is also important to note that only a small number of bound taxane molecules are required to stabilize the dynamics of a microtubule enough to cause cellular disruption and taxanes can therefore still be effective at low concentrations and without inducing polymerisation [41].

### 1.4.2 Metabolism

Cytochrome P450 is a superfamily of enzymes that are responsible for metabolising the majority of drug molecules and toxins, including DOC. DOC is primarily metabolized by CYP3A4 [21, 42], but is also metabolised by CYP3A5 and CYP1B1 [43]. This activity takes place primarily within the liver and secondly the small bowel, the principal locations of CYP3A4 production [44]. "Metabolism is principally oxidative and at the tert-butylpropionate side chain, resulting first in an alcohol docetaxel (M2), which is then cyclized to three further metabolites (M1, M3, and M4). M1 and M3 are two diastereomeric hydroxyoxazolidinones and M4 is an oxazolidinedione" [45]. None of DOC's metabolites show significant cytotoxicity [42]. As DOC is mainly metabolised by CYP3A4, caution should be used when administering DOC in combination with compounds that induce, inhibit, or are metabolized by CYP3A4 [21] .

The PK and PD pathways of DOC through a hepatic cell are depicted in Fig. 1.3. DOC enters the cell via an organic anion transporter encoded by the pharmacogene SLCO1B3.

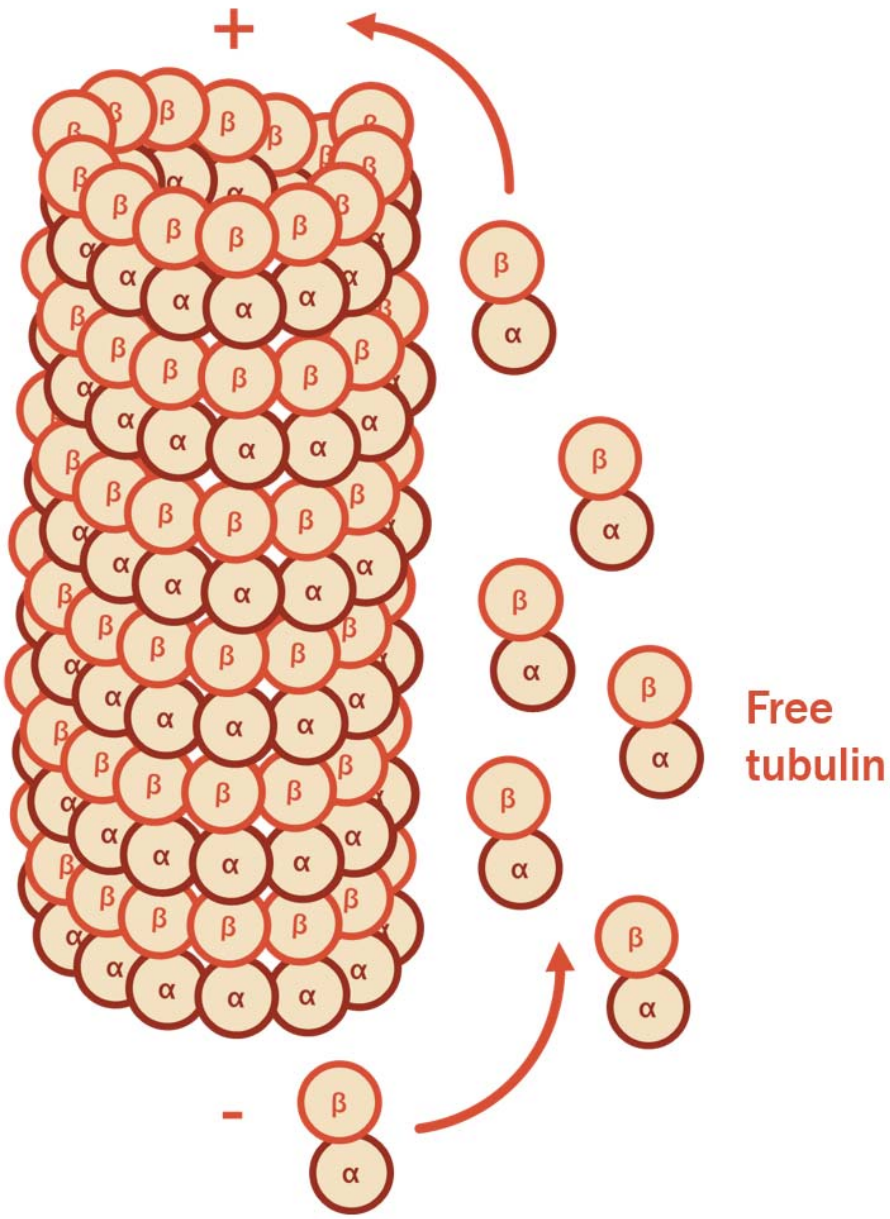



Figure 1.2: The structure of a microtubule.

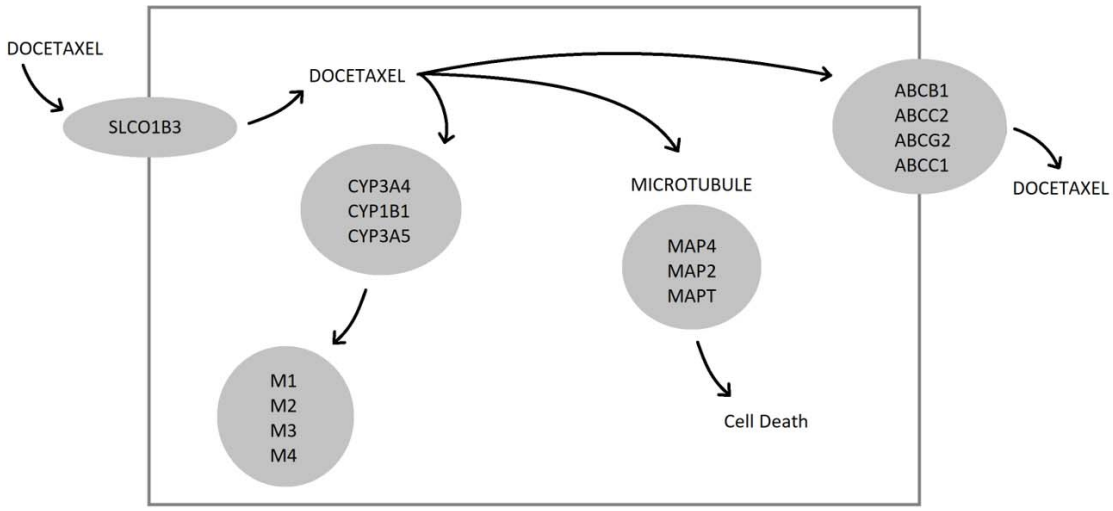



Figure 1.3: The pharmacogenes involved in the PK and PD of DOC. Figure adapted from [43].

These transporters are associated with the uptake of xenobiotic compounds in the liver [46]. It may be removed from the cell via ABCB1, ABCC2, ABCG2, and ABCC1 multidrug transporters. Neither of these actions are thought to have significant affects on the PK of DOC. Once it has entered the cell, DOC will either destroy the cell by binding with tubulin proteins as described in §1.4.1, or be metabolised as described above.

### 1.4.3 PK Metrics of DOC

Previous studies of DOC have made estimates of the PK metrics of DOC. At the recommended clinical doses of 70-115 mg/m$^2$ DOC's kinetic profile is considered to be linear and dose independent and the profile is consistent with a three-compartment model [21].

The monograph for DOC suggests that the total body clearance of DOC is 21 L/h/m$^2$, and it has a tri-phase elimination with half lives of $t_{1/2,\alpha}$ = 4 min, $t_{1/2,\beta}$ = 36 min, and $t_{1/2,\gamma}$ = 11.1 h [21]. DOC is also highly distributed through tissues. The whole body volume of distribution is roughly 74 L/m$^2$ [22].

Renal clearance is minimal (<5-10%), with the clearance being mostly hepatic (70-80%). The nature of hepatic clearance is through inactivation via oxidation by cytochrome

P450 (primarily CYP3A4) [22]. This means that the PKs of DOC is affected mainly by liver hepatic impairment. Patients with elevated serum levels of alkaline phosphatase and hepatocellular enzymes had 25% reduced clearance of DOC [22]. In another study clearance was decreased by an average of 47% in patients with hepatic impairment [21].

DOC has been observed to bind strongly with plasma proteins (>90%). It has a high affinity to $\alpha$1-acid glycoprotein as well as albumin and lipoproteins. It is this protein binding that is thought to lead to the high inter-patient variability in DOC concentrations [22]. No strong link between age or sex has been noticed to affect PK metrics [21].

### 1.4.4 Treatment Emergent Adverse Effect (TEAE)

The frequency and severity of adverse reactions will depend largely on the specific dosage regimen. A rough overview of common complications that arise with DOC treatment is given in Tbl. 1.2. The safety profile is generally similar in all patient types and does not depend on the type of cancer being treated.

The principle toxicity of DOC is myelosupression with neutrophil counts dropping as dosage increases [22]. The development of neutropenia is typically the dose limiting toxicity. Hypersensitivity is less common than in pacitaxel (which develops mainly due to reactions to it's encapsulation, Cremophor El) but can arise and is more common in patients who have not receive premedication.

Fluid retention characterized by edema and third-space fluid collections may occur as a result of increased capillary permeability. Though it is typically not significant at doses <400 mg/m$^2$ and can be mitigated with the use of corticosteroids. Fluid retention is self resolving once treatment is stopped [22].

Persistent TEAEs (those that arose during treatment and remained during follow up) will mainly resolve after treatment with DOC is stopped. Adverse effects for grade 3-4 that started or worsened during follow up were experienced by >10% of patients, the most common and severe being cardiac failure [21].

Table 1.2: Adverse effects of DOC.

| Class | Reaction |
|---|---|
| Carcinogenesis and Mutagenesis | Secondary malignancies may develop, including acute myeloid leukemia, myelodysplastic syndrome, non-Hodgkin lymphoma, and renal cancer. Risk is increased with treatments using combination regimes. |
| Cardiovascular | Venous thromboembolic events, myocardial infaraction. Ventricular arrhythmia observed with treatments using combination regimens. |
| Fluid retention | Severe fluid retention requires premedication with corticosteroids. |
| Gastrointestinal | Neutropenia associated with high risk of complications developing. Enterocolitis may develop rapidly. |
| Hematologic | Neutropenia. |
| Hepatic | Liver failure. |
| Hypersensitivity Reactions | Hypotension, bronchospasm, rash, fatal anaphylaxis. |
| Ophthalomologic | Cystoid macular edema. |
| Respiratory | Acute Respiratory distress syndrome, interstitial pneumonia, pulmonary fibrosis, respiratory failure. |
| Skin | Localized erythema of the extremities with edema, and severse cutaneous adverse reactions. |

## 1.5 Specific Aims

Our aim is to develop an intensive quantitative model of DOC and it's encapsulation, P80, that is able to accurately predict the time course concentration using PK principles. This is achieved by analysing numerical solutions to a set of nonlinear differential equations (NDEs) which are informed by the known pathways of the relevant molecules. Development of the model will permit PK metrics to be measured with a higher degree of accuracy and with more flexibility than standard PK packages that are used in the industry currently such as WinNonlin or NONMEM. The ability to individualize our model may be able to be used to further increase the accuracy for patients.

Within the next chapter the model itself, as well as the mathematical techniques used to analyse the model will be covered. Model solutions are found using the $4^{th}$ Order Runge-Kutta (RK) method. The parameter space of the solutions are searched using Powell's

Method in Multidimensions to find values that minimize the percentage variance between the model and clinical data. The fitness of the model is evaluated using the Akaike Information Criterion (AIC). The results of the parameter development are presented in §2. The PK metrics predicted by the best model are presented in §3.

# Chapter 2

# Methods

*"In mathematics you don't understand things. You just get used to them."*

– Johann von Neumann

## 2.1 The Model

Compartmental models are frequently used to predict the PK behaviours of drugs and other relevant molecules. This method models the body as a set of distinct parts within which ADME processes can play out. Each compartment is assumed to be more or less homogeneous and permits a specific set of behaviours to take place as tailored to the specific problem at hand.

The $M$-compartment $N$-molecule model outlined below is the most general form of the model. The simplest case is the one-compartment model. The one-compartment model represents all interactions within the body as a single compartment. This compartment can be further divided to increase the physiological accuracy of the model. This is the case in the two-compartment model where the first compartment represents the transport compartment, i.e. the blood plasma, and the second compartment represents all other activities in either the liver or in the healthy cells and tumour cells. Similarly, the three-compartment model improves the model accuracy by increasing the number of compartments with the first compartment representing the blood plasma, the second compartment representing the liver, and the third compartment representing healthy cells and tumour cells. In the four-

compartment model the third compartment represents healthy cells and the fourth compartment represents tumour cells. Each of these models also includes a zeroth compartment which represents the environment external to the body.

The rate of change of the molecule $d$ within compartment $c$ is described by the non-linear differential equations

$$\frac{dX_{d,c}}{dt} = I_{d,c}(t) + \frac{dX_{d,c}^{(1)}}{dt} + \frac{dX_{d,c}^{(2)}}{dt} + \frac{dX_{d,c}^{(3)}}{dt} + \frac{dX_{d,c}^{(4)}}{dt}. \tag{2.1}$$

The amount of molecule $d$ in compartment $c$ is $X_{d,c}$. The concentration of the molecule is

$$C_{d,c} = \frac{X_{d,c}}{V_{D(d,c)}}, \tag{2.2}$$

where $V_{D(d,c)}$ is the volume of distribution of molecule $d$ within compartment $c$. When multiple molecules interacting in a number of compartments are considered a set of coupled nonlinear DEs are formed.

The right-hand side of Eqn. 2.1 has been split into five terms where each term represents a specific behaviour which are described below. These equations are able to account for the behaviours which were outlined in §1.2.1.

### 2.1.1 Infusion

The first term on the right-hand side of Eqn. 2.1, $I_{d,c}(t)$, represents the infusion of molecule $d$ into compartment $c$. This process is depicted in Fig. 2.1. This can be the intravenous (IV) infusion of a free drug into the blood plasma, or the intake of a prepared drug such as with oral drugs, micelle encapsulated drugs or liposome encapsulated drugs. For drugs given intravenously, as DOC is,

$$I_{d,c}(t) = \begin{cases} \frac{D}{T_{inf}}, & \text{if } t_i \leq t \leq t_f \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

where $D$ is the dosage, and $T_{inf} = t_f - t_i$ is the duration of the infusion that begins at $t_i$ and ends at $t_f$. This form assumes a constant homogeneous infusion.

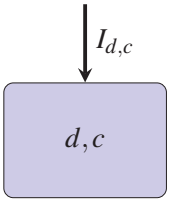



Figure 2.1: The infusion of molecule $d$ into compartment $c$.

Dosing is frequently determined based on a patients Body Surface Area (BSA). BSA values are estimated using a variety of formulas that utilize the patients weight and height. The use of BSA is thought to improve the efficacy of treatment by controlling for interpatient variability [47]. As the BSA of individual patients is not given explicitly the quantity and concentrations of molecules used in this study are modified by BSA based on the dosage.

### 2.1.2 Flow

The second term on the right-hand side of the differential equation in Eqn. 2.1 describes the flow of molecule $d$ from compartment $c$ into another compartment $c'$ and vice versa. This process is depicted in Fig. 2.2. The differential equation is;

$$\frac{dX^{(1)}_{d,c}}{dt} = \sum_{p=0}^{2}\sum_{c'=0}^{M}\left\{\frac{\overline{k}^{(1)}_{s1'}X^{A^{(1)}_{s1'}}_{d,c'}}{1+\Gamma^{(1)}_{s1'}X^{B^{(1)}_{s1'}}_{d,c'}} - \frac{\overline{k}^{(1)}_{s1}X^{A^{(1)}_{s1}}_{d,c}}{1+\Gamma^{(1)}_{s1}X^{B^{(1)}_{s1}}_{d,c}}\right\}, \tag{2.4}$$

where the subscript notation has been condensed for readability. For each parameter $s1 = p,c,c'$ and $s1' = p,c',c$, i.e. $\overline{k}^{(1)}_{s1} = \overline{k}^{(1)}_{p,c,c'}$. $p$ represents the process, which determines the exponents, $A$ and $B$. That is, the zeroth order process, $p = 0$, is $A^{(1)}_{0,c,c'} = B^{(1)}_{0,c,c'} = 0$; the first order process, $p = 1$, is $A^{(1)}_{1,c,c'} = B^{(1)}_{1,c,c'} = 1$; and the second order process, $p = 2$, is $A^{(1)}_{2,c,c'} = B^{(1)}_{2,c,c'} = 2$. The exponents $A^{(1)}_{p,c,c'}$ and $B^{(1)}_{p,c,c'}$ will be allowed to vary from these values when the model is fitted to the clinical data in some circumstances. Besides the exponents, the

parameters are the saturation parameter $\Gamma^{(1)}_{p,c,c'}$ and the effective rate constant $\overline{k}^{(1)}_{p,c,c'}$ where the bar indicates that this parameter can change in the presence of other molecules.

The compartment $c' = 0$ corresponds to the molecule being external to the body. The special case of a molecule flowing into $c' = 0$ is depicted in Fig. 2.3. This case covers the direct elimination of a molecule and permits the molecule to flow in only one direction. The rest of the compartments $c' = 1$ to $M$ represent the body with the number $M$ varying from one for the one-compartment model to four in the four-compartment model.

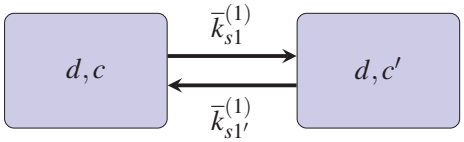



Figure 2.2: The flow of molecule $d$ from compartment $c$ into compartment $c'$ and vice versa.

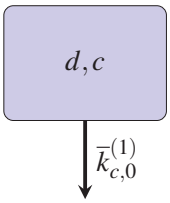



Figure 2.3: The elimination of molecule $d$ from compartment $c$ into the external environment. This process is irreversible.

### 2.1.3 Transformation

The third term on the right-hand side of Eqn. 2.1 describes the transformation of molecule $d$ in compartment $c$ into another molecule, $d'$ in compartment $c'$. This process is depicted in Fig. 2.4. The differential equation is;

$$\frac{dX^{(2)}_{d,c}}{dt} = \sum_{d'=1}^{N}\sum_{c'=0}^{M}\sum_{p=0}^{2}\left\{\frac{\delta^{(2)}_{s2'}\overline{k}^{(2)}_{s2'}X^{A^{(2)}_{s2'}}_{d',c'}}{1+\Gamma^{(2)}_{s2'}X^{B^{(2)}_{s2'}}_{d',c'}} - \frac{\delta^{(2)}_{s2}\overline{k}^{(2)}_{s2}X^{A^{(2)}_{s2}}_{d,c}}{1+\Gamma^{(2)}_{s2}X^{B^{(2)}_{s2}}_{d,c}}\right\}, \tag{2.5}$$

where the subscript notation has been condensed for readability. For each parameter $s2 = p,d,c,d',c'$ and $s2' = p,d',c',d,c$ i.e. $\overline{k}^{(2)}_{s2} = \overline{k}^{(2)}_{p,d,c,d',c'}$. $p$ is the process order and behaves

as described in §2.1.2. The δ's are stoichiometric coefficients which serve to enforce conservation of mass.

This term is relevant for the conversion of P80 encapsulated DOC into free DOC.

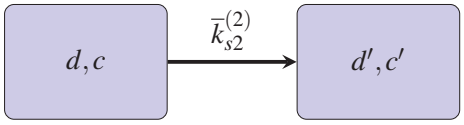



Figure 2.4: The transformation of molecule $d$ in compartment $c$ into molecule $d'$ in compartment $c'$.

### 2.1.4 Metabolization

The fourth term on the right-hand side of the differential equation in Eqn. 2.1 describes the conversion of one type of molecule into two new molecules. For example, the breakup of a complex molecule $d$ into an enzyme $d'$ and a metabolite $d''$. In principle this interaction can produce molecules outside of the compartment but for simplicity we will consider the case where metabolization is constrained within a single compartment $c$. This process is depicted in Fig. 2.5. The differential equation is;

$$\frac{dX^{(3)}_{d,c}}{dt} = \sum_{d',d''=1}^{N} \sum_{p=0}^{2} \left\{ \frac{-\delta^{(3)}_{s3} \overline{k}^{(3)}_{s3} X^{A^{(3)}_{s3}}_{d,c}}{1+\Gamma^{(3)}_{s3} X^{B^{(3)}_{s3}}_{d,c}} + \frac{\delta^{(3)}_{s3'} \overline{k}^{(3)}_{s3'} X^{A^{(3)}_{s3'}}_{d',c}}{1+\Gamma^{(3)}_{s3'} X^{B^{(3)}_{s3'}}_{d',c}} + \frac{\delta^{(3)}_{s3''} \overline{k}^{(3)}_{s3''} X^{A^{(3)}_{s3''}}_{d'',c}}{1+\Gamma^{(3)}_{s3''} X^{B^{(3)}_{s3''}}_{d'',c}} \right\}, \tag{2.6}$$

where the subscript notation has been condensed for readability. For each parameter $s3 = p, d|d', d''$, $s3' = p, d'|d, d''$, and $s3'' = p, d''|d, d'$, i.e. $\overline{k}^{(3)}_{s3} = \overline{k}^{(3)}_{p,d|d',d''}$. $p$ is the process order and behaves as described in §2.1.2.

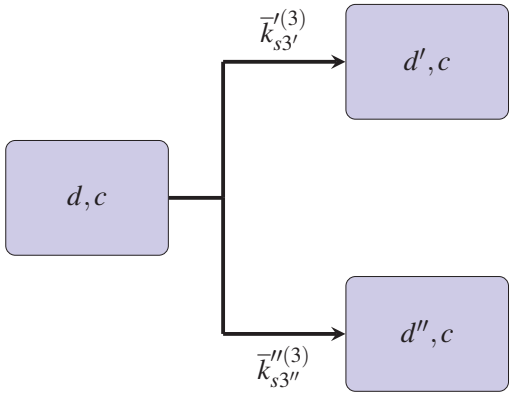



Figure 2.5: The conversion of molecule $d$ into molecules $d'$ and $d''$ within compartment $c$. Typically, the molecules involved will be a complex, an enzyme, and a metabolite, respectively.

### 2.1.5 Binding

The fifth term in Eqn. 2.1 is the reverse of the previous case and describes two types of molecules combining together to form a single molecule. An example of this is the formation of a complex $d$ from a substrate $d''$ binding to a ligand $d'$. As with the metabolization of a compound this interaction is not required to take place within a single compartment but, for simplicity, we will constrain binding to just one compartment $c$. This process is depicted in Fig. 2.6. The differential equation is;

$$\frac{dX^{(4)}_{d,c}}{dt} = \sum_{d',d''=1}^{N}\sum_{p,p'=0}^{2}\left\{\frac{\overline{k}^{(4)}_{s4}X^{A^{(4)}_{S4}}_{d',c}X^{A^{(5)}_{S5}}_{d'',c}}{1+\Gamma^{(4)}_{s4}X^{B^{(4)}_{S4}}_{d',c}X^{B^{(5)}_{S5}}_{d'',c}} - \frac{\overline{k}^{(4)}_{s4''}X^{A^{(4)}_{S4''}}_{d,c}X^{A^{(5)}_{S5''}}_{d',c}}{1+\Gamma^{(4)}_{s4''}X^{B^{(4)}_{S4''}}_{d,c}X^{B^{(5)}_{S5''}}_{d',c}} - \frac{\overline{k}^{(4)}_{s4'}X^{A^{(4)}_{S4'}}_{d,c}X^{A^{(5)}_{S5'}}_{d'',c}}{1+\Gamma^{(4)}_{s4'}X^{B^{(4)}_{S4'}}_{d,c}X^{B^{(5)}_{S5'}}_{d'',c}}\right\}, \tag{2.7}$$

where the subscript notation has been condensed for readability. For each parameter $s4 = p,p',d',d''|d$, $s4'' = p,p',d,d'|d''$, and $s4' = p,p',d,d''|d'$. The subscripts associated with the exponents are $S4 = p,d',d''|d$, $S4'' = p,d,d'|d''$, $S4' = p,d,d''|d'$, $S5 = p',d',d''|d$, $S5'' = p',d,d'|d''$, and $S5' = p',d,d''|d'$.

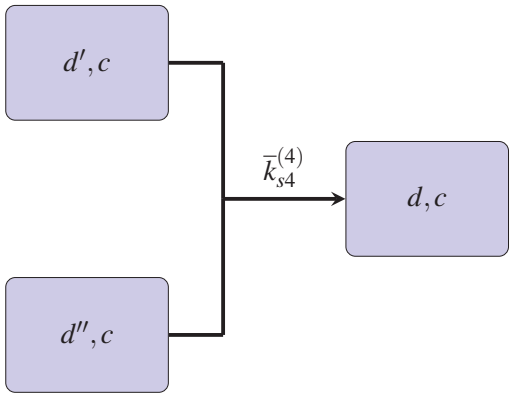



Figure 2.6: The binding of molecule $d'$ with molecule $d''$ to form the molecule $d$ within compartment $c$. Typically, the molecules involved will be an enzyme binding with a ligand to form a complex, respectively.

The processes of binding and metabolization are important for pro-drugs as well as drugs which have active metabolites. None of the metabolites of DOC are active[48]. As they are not clinically relevant, concentration-time data of the metabolites is not available in the literature and thus these terms were not included in the models of DOC.

## 2.2 Pharmacokinetic Parameters

Determination of the pharmacokinetic parameters described by Eqn. 2.3-2.7, as well as the volume of distribution was achieved through minimization of the weighted percentage variance

$$S_p = \frac{4}{N_p} \sum_{i,j,l,m} n_{i,j,l,m} \left( \frac{X_{i,j}^{(m)}(t_l) - V_{d,i,j} C_{i,j}^{(m)}(t_l)}{X_{i,j}^{(m)}(t_l) + V_{d,i,j} C_{i,j}^{(m)}(t_l)} \right)^2 \tag{2.8}$$

where $X_{i,j}^{(m)}(t_l)$ is the theoretical amount of molecule $i$ in compartment $j$ at time $t_l$ corresponding to the dosage from clinical study $m$ and $C_{i,j}^{(m)}(t_l)$ is the clinical concentration of molecule $i$ in compartment $j$ at time $t_l$ from clinical study $m$. $n_{i,j,l,m}$ is the weighting coefficient corresponding to the number of patients in study $m$ at time $t_l$. The normalization factor is

$$N_p = \sum_{i,j,l,m} n_{i,j,l,m}. \tag{2.9}$$

The clinical concentrations that are being compared against the theoretically calculated values were obtained from published clinical data studies which used P80/DOC alone. The concentration-time graphs presented as results were digitized and converted to micromoles per litre ($\mu$mol/L). All studies used in the determination of parameters use units of milligram per metre squared (mg/m$^2$) for the dosage.

## 2.3 Computing PK Metrics

Clinically relevant metrics can be determined using the best performing model. The clinical relevance of these terms was outline in §1.1.5.

The maximum concentration $C_{Max}$ is found from the models post infusion peak. It is simply the largest concentration reached by the concentration-time curve. $T_{Max}$ is then the time which corresponds to $C_{Max}$.

The $AUC$ is defined by

$$AUC = \lim_{t\to\infty} \int_{t=0}^{t} C_{i,j}(t)dt, \tag{2.10}$$

however, in practice, we are typically unable to find analytic solutions for $C_{i,j}(t)$. In these cases we may still determine the $AUC$ numerically using the linear trapezoidal method [3],

$$AUC = \frac{1}{2}\sum_{i}^{R}(C_{i,j}(t_i) + C_{i,j}(t_{i+1}))(t_{i+1} - t_i). \tag{2.11}$$

The concentrations are computed at small intervals in order to approximate a continuous curve and improve the accuracy of the calculation. $\Delta t$ ranges from 0.25-10 min, increasing with $t$. The $AUC$ is generally computed across a finite interval which depends on either the time corresponding to the final blood sample collected or a pre-determined interval such at 24-hours post infusion. $R$ is determined based on the specific interval under consideration.

The half-lives are the most difficult to precisely determine from the model, particularly when the behaviour is non-linear. However, they can be roughly estimated from a semi-log plot of the concentration-time curve. If the behaviour at long time intervals is roughly

linear, the slope $\lambda_\gamma$ of that section is related to the terminal half-life by,

$$t_{1/2,\gamma} = \frac{\ln(2)}{-\lambda_\gamma}. \tag{2.12}$$

Similarly, this can be done for any part of the semi-log plot which is approximately linear. The distributive half-life can be determined from the slope $\gamma_\alpha$ of the semi-log plot immediately following the end of infusion using,

$$t_{1/2,\alpha} = \frac{\ln(2)}{-\lambda_\alpha}. \tag{2.13}$$

The specific ranges used will depend heavily on the specific model under review.

## 2.4 Fourth-Order Runge-Kutta Method

The fourth-order Runge-Kutta Method was used to solve the differential equations described in section 2.1. The function `RK4()` was coded to produce a solution for the model. The code used is presented in §C. The Runge-Kutta methods are a family of widely used numerical methods, useful for efficiently producing stable solutions to ODEs. This is achieved in a step-by-step fashion, by evaluating the ODE along a trial step $h$. More complex evaluations will produce more precise solutions with an $n$th-order RK method having error terms of order $O(h^{n+1})$ [49].

From a known initial position $(t, X_0, X_1, \ldots, X_i)$ the fourth-order method evaluates the set of $i$ coupled ODE's $dX_i/dt = f^{(i)}(t, X_0, X_1, \ldots, X_i)$ at the points

$$\begin{aligned}
k_1^{(i)} &= hf(t, X_0, X_1, \ldots, X_i) \\
k_2^{(i)} &= hf(t + \frac{h}{2}, X_0 + \frac{k_1^{(i)}}{2}, X_1 + \frac{k_1^{(i)}}{2}, \ldots, X_i + \frac{k_1^{(i)}}{2}) \\
k_3^{(i)} &= hf(t + \frac{h}{2}, X_0 + \frac{k_2^{(i)}}{2}, X_1 + \frac{k_2^{(i)}}{2}, \ldots, X_i + \frac{k_2^{(i)}}{2}) \\
k_4^{(i)} &= hf(t + h, X_0 + k_3^{(i)}, X_1 + k_3^{(i)}, \ldots, X_i + k_3^{(i)})
\end{aligned}$$

and projects the solution as

$$X^{(i)}(t+h) = X^{(i)}(t) + \frac{k_1^{(i)}}{6} + \frac{k_2^{(i)}}{3} + \frac{k_3^{(i)}}{3} + \frac{k_4^{(i)}}{6} + O(h^5). \tag{2.14}$$

This process is repeated, with care taken to ensure proper convergence, until the desired endpoint is reached [49].

## 2.5 Powell's Method in Multidimension

Finding the global extremum of a function is a famously difficult problem. The standard methods of searching involve either finding local extrema from many unique starting positions, then taking the most extreme as the solution or probing the space around local extrema to see if either a better solution can be found or if the original extrema is returned to. Due to the nature of these problems no one optimization method is perfect and the method selected must conform to the nature of the problem being solved.

The constraints of our model suggest use of Powell's method in multidimensions which has the appealing property of not requiring the computation of derivatives. This allows it to be applied to numerical solutions without much hassle.

Our problem requires varying the model parameters until minimization of the weighted percentage variance between the numerical solution of the model $X$ and the corresponding clinical drug concentrations $C$ was achieved. The percentage variance is given by Eqn. 2.8.

For a function of $M$ variables we begin by calculating $S_p(\mathbf{P})$, where $\mathbf{P}_i$ is an arbitrary start point, and defining a set of $M$ search vectors $\mathbf{e_0}, \mathbf{e_1}, \ldots \mathbf{e_M}$. $S_p$ is then minimized along each direction successively to reach $S_p(\mathbf{P}_{i+1})$. The search vectors are then updated such that $\mathbf{e}_i \rightarrow \mathbf{e}_{i+1}$ and $\mathbf{e}_M = \mathbf{P}_{i+1} - \mathbf{P}_i$. $S_p(\mathbf{P}_{i+1})$ is taken as the new start point and the process is repeated. This procedure continues until a minimum is located.

The introduction of the new search vector serves to speed up the process, allowing the program to minimize the function more directly. The disadvantage of replacing $\mathbf{e}_M$ with

each pass is that the search vectors tend to become aligned, resulting in a set of search vectors that no longer span the entire parameter space. This was mitigated, somewhat counter intuitively, by replacing the most successful search vector with $\mathbf{P}_{i+1} - \mathbf{P}_i$ with each cycle. By replacing the vector corresponding to the direction of largest decrease with a vector that already points more or less in the same direction we prevent the search vectors from building up a linear dependence. This, in combination with periodically reinitializing the search vectors to their original values, allows the minimum to be efficiently located [49].

The only restriction placed on the varying parameters is that each element of $\mathbf{P} \geq 0$.

## 2.6 Akaike Information Criterion

As discussed in §1.2 it is important to strike a balance between simplicity and complexity while building a model. Each model is at risk of either over or underfitting and their respective consequences of over-complicating the system or information loss. A quantitative test, which removes the subjective judgement from the evaluation of each models utility, is necessary.

To evaluate the relative quality of each model the Akaike information criterion (AIC) will be employed. It has been shown to be particularly well suited for time series analysis [50] which lends itself well to the problem at hand. This criterion is defined as

$$AIC = 2\eta - 2\ln L \tag{2.15}$$

where $\eta$ is the number of parameters associated with the model and $L$ is the maximum value of the likelihood function [51]. For problems where the errors associated with the data are independent identical normal distributions [50], and tailoring the expression for our models, we may calculate the criterion as

$$AIC = 2(\eta + 1) + N_p \ln(S_p) \tag{2.16}$$

where $N_p$ is the total number of data points, Eqn. 2.9, and $S_p$ is the percentage variance given by Eqn. 2.8.

It is necessary to include a corrective term when calculating AIC when the sample size is small, $N_p/\eta < 40$. For these cases

$$AIC_c = AIC + \frac{2(\eta+1)(\eta+2)}{N_p - \eta} \tag{2.17}$$

is used [52].

It is the model with the lowest AIC (or $AIC_c$) value that gives the best fit. It is important to note that an AIC value on its own provides no information about the quality of a fit. It is only useful when compared against the performance of other models on a given data set. The difference between the AIC of a model and the AIC of the best performing model

$$\Delta = AIC - AIC_{min} \tag{2.18}$$

can be used to evaluate the level of empirical evidence in support of a given model with larger values of $\Delta$ indicating less evidence. As a rule of thumb, $\Delta > 10$ suggests that there is essentially no support for the model, $7 > \Delta > 4$ implies a weak level of evidence, and $\Delta < 2$ suggests a substantial level of evidence for a model [52]. The best performing model will always have $\Delta = \Delta_{min} = 0$.

## 2.7 Procedure Summary

In order to minimize the model outlined above against clinical concentration-time data we first start with an arbitrary point in the parameter space. This point may be informed by curve stripping or by previous fits in some cases. The model is solved using the fourth-order Runge-Kutta method, and the weighted percentage variance is computed from that result. The parameter space is then searched using Powell's method in multi-dimensions until a minimum is found. Selected models are compared using the Akaike information criterion.

# Chapter 3

# Parameter Development

*" A complex system that works is invariably found to have evolved from a simple system that works."*

– John Gaule

To build a model of DOC and its related molecules we will apply the techniques described in §2. The clinical data used in this research was obtained by digitizing DOC and P80 concentration-time curves available in the literature using WebPlotDigitizer v4.6 [53].

Parameters are obtained for a number of specific cases of the model described in §2.1 by minimizing $S_p$ (see Eqn. 2.8) against a set of calibration data. The calibration data is comprised of concentration-time curve studies with a large number of patient data. The one exception to this is Baker et al. [54] which is a single patient study but has been included with the calibration data because concentration-time data for P80 was also included for the patient. Collectively, this data provides sufficient variation in the number, age, gender and type of cancer between patients to provide a baseline for the determination of DOC PKs. As discussed in §2.5, fitting of the calibration data is achieved using `POWELL()` which is available in §C. The number in brackets following each parameter is the deviation which results in a 1% increase in $S_p$. This value serves as the error in each parameter or, alternatively, the sensitivity of the model to that parameter. Any value presented without a corresponding error was fixed and not permitted to vary.

Each model is validated against the other models. The performance of each model is compared using $AIC_c$ (see Eqn. 2.17) values computed from the calibration data. A variety of models are examined with the complexity increasing until additional parameters provide diminishing returns as determined by the $AIC_c$. The robustness of the selected model is then validated by computing $S_p$ for a set of evaluation data. The evaluation data is comprised of smaller clinical studies as well as studies which include factors that may influence the PKs of DOC such as liver impairment or concurrent drug treatments.

The complexity of the model is built up slowly. First, we will consider one-compartment one-molecule models of P80 and DOC treated separately. The P80 and DOC models with the best performance are then combined in a two-molecule model and evaluated. The results of the one-compartment models then inform the development of the two-compartment models and, in turn, the three-compartment models.

All models were solved numerically with C++. The models that could be easily solved analytically served as a measure for testing the accuracy of the code. These simple models were solved both analytically and numerically in Wolfram Mathematica 12.0 [55] and showed agreement with `RK4()` which can be found in §C. For consistency and clarity, only the numerical results are presented.

## 3.1 Model Types

The type of behaviour captured by the model will be dependent on which of its parameters are allowed to vary. Each model may either permit only one of the behaviour types outlined below, or a combination of the types as a series. For example the dominant process of a model may be combined with a correction term of a different type. The naming conventions used to distinguish between each type of model are outlined below.

### 3.1.1 Fixed Order Process

Single, fixed-order processes are denoted using the order of the process, i.e. the exponent, $A$, is fixed. A zeroth order model with $A = 0$ is denoted 'zeroth' , a first order with $A = 1$, 'first', and a second order with $A = 2$, 'second'. These models permit only $\overline{k}$ to vary and have $\Gamma = 0$.

### 3.1.2 Fractal Process

When the order of the process $A$ is permitted to vary, these models are denoted as 'fractal'. These models permit $\overline{k}$ and $A$ to vary and have $\Gamma = 0$. Multiple fractal processes may be included within a model. The architecture of biological structures important to PKs, such as the liver, can be self-similar. Diffusion across such structures is characterized by fractional power law time dependence [13].

### 3.1.3 Saturable Kinetic Process

When all parameters in each term are allowed to vary the model type is denoted as 'saturable'. These models permit $k$, $\Gamma$, $A$, and $B$ to vary. This is the most general process of the model. Multiple saturable processes may be included within a model. A special case of this model type is the Michaelis-Menten model [10] which has a fixed first order process with $A = B = 1$, and $\overline{k}$ and $\Gamma$ allowed to vary. These models are denoted with 'MM' as their model type. Another special case has $B = 1$, and $\overline{k}$, $\Gamma$, and $A$ allowed to vary. These models are denoted 'fractal MM'.

### 3.1.4 Mixed

When interaction terms are included, i.e. when the processes described above are combined, the model type is denoted as 'mixed'. To differentiate between mixed models the dominant process will be used. For example, a fractal model with a first order correction would be denoted 'mixed fractal'. Models which permit multiple fixed order processes are denoted 'mixed order'.

## 3.2 One-Compartment One-Molecule Models

The simplest possible model considers a single molecule which enters a compartment via infusion and then undergoes direct elimination. These models were developed for both P80 and DOC independently.

### 3.2.1 Determining P80 Parameters

The clinical papers used as the calibration and evaluation data in the determination of the P80 parameters are given in Tbl. 3.1 and Tbl. 3.2 respectively. The tables include the dosage, infusion time $T_{inf}$, and weighting coefficient corresponding to the number of patients $n_p$ and data points $n_d$ in that study. As P80 is thought to have little affect on the PKs and efficacy of DOC, relatively little concentration-time data has been published. Baker et al. [54] was used to fit the model and the remaining papers were used for verification of the model's accuracy. Baker et al. [54] was selected because it has the largest amount of corresponding DOC patient data.

Table 3.1: Clinical paper used in the calibration of P80 model parameters.

| Reference | Dosage (mg/m$^2$) | $T_{inf}$ (min) | $n_p$ | $n_d$ |
|---|---|---|---|---|
| Baker et al. [54] | 75 | 60 | 1 | 6 |

Table 3.2: Clinical papers used in the evaluation of P80 model parameters.

| Reference | Dosage (mg/m$^2$) | $T_{inf}$ (min) | $n_p$ | $n_d$ |
|---|---|---|---|---|
| Baker et al. [56]a | 30 | 60 | 1 | 6 |
| Baker et al. [56]b | 75 | 60 | 1 | 6 |

The general one-compartment one-molecule model used in the determination of P80

parameters is depicted in Fig. 3.1 and given by

$$\frac{dX_{P80,1}}{dt} = I_{P80,1} - \sum_{p=0}^{2} \left\{ \frac{\overline{k}^{(1)}_{p,1,0} X^{A^{(1)}_{p,1,0}}_{P80,1}}{1 + \Gamma^{(1)}_{p,1,0} X^{B^{(1)}_{p,1,0}}_{P80,1}} \right\}. \tag{3.1}$$

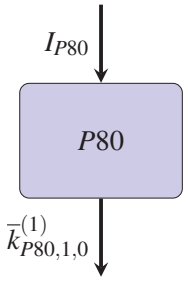



Figure 3.1: Single compartment model of P80.

The methods discussed in §2 were used in the development of the parameters for the one-compartment P80 model. Specifically, the calibration data is curve stripped using semi-log plots of the concentration-time data [57] and the results were used to initialize the parameter search (see §2.5).

The fit results for the best performing one-compartment P80 models can be found in Tbl. 3.3 and seen in Fig. 3.2. The results of the statistical tests for each model are given in Tbl. 3.4, including the variance $S_p$ and the $\text{AIC}_c$ computed between the model and Baker et al. [54]. $\Delta$ has also been provided for the convenience of the reader.

By considering the $\text{AIC}_c$ differences, $\Delta$, we see that the fractal model shows the greatest level of empirical evidence for the behaviour of P80. While the behaviour may appear linear, it is much better described by parametrizing the process order. Further parameterization improves the variance but the gains in $S_p$ are not enough to balance the cost of the additional parameters as determined by the $\text{AIC}_c$. Other models were also tested but gave significantly worse results. The statistics for these can be found in §B.1.1. The $\text{AIC}_c$ differences show that the level of evidence for the fractal model is significant. We also note the relatively small $V_{D,(P80,1)}$, which indicates that the distribution of P80 into the surrounding tissues is limited. This motivates the decision to use the fractal P80 model to describe the

release of DOC from the encapsulation as a single process in §3.3.

Table 3.3: One-compartment P80 model results.

| Parameter* | First | Fractal | Mixed Order |
|---|---|---|---|
| $\overline{k}^{(1)}_{0,1,0}$ | 0 | 0 | 0.0000000(5) |
| $\overline{k}^{(1)}_{1,1,0}$ | 0.02663(19) | 0.014956(47) | 0.22574(13) |
| $\overline{k}^{(1)}_{2,1,0}$ | 0 | 0 | 0.0000399(10) |
| $A^{(1)}_{0,1,0}$ | - | - | 0 |
| $A^{(1)}_{1,1,0}$ | 1 | 1.14623(58) | 1 |
| $A^{(1)}_{2,1,0}$ | - | - | 2 |
| $\Gamma^{(1)}_{1,1,0}$ | 0 | 0 | 0 |
| $V_{D,(P80,1)}$ | 3.329(67) | 2.369(12) | 2.202(20) |

* The units for $\overline{k}_{p,c,c'}$ are $[(\frac{1}{\mathrm{s}})(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{1-A_{p,c,c'}}]$ , the units for $\Gamma_{p,1}$ are $[(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{1-B_{p,c,c'}}]$, and the units for $V_{D,(P80,1)}$ are $[\frac{\mathrm{L}}{\mathrm{m}^2}]$. All other parameters are dimensionless.

Table 3.4: One-compartment P80 model statistics.

| Statistic | First | Fractal | Mixed Order |
|---|---|---|---|
| $S_p$ | 0.0396 | 0.0037 | 0.0096 |
| $\mathrm{AIC}_c$ | -7.377 | -12.34 | -6.520 |
| $\Delta$ | 4.97 | 0.00 | 5.82 |

The fractal model is compared to the evaluation data in Fig. 3.3. The predictive power of the model varies for each data set. Baker et al. [56]a is well predicted by the model with a variance of $S_p = 0.0592$ while Baker et al. [56]b is poorly described, particularly at high concentrations, with a variance of $S_p = 0.4215$.

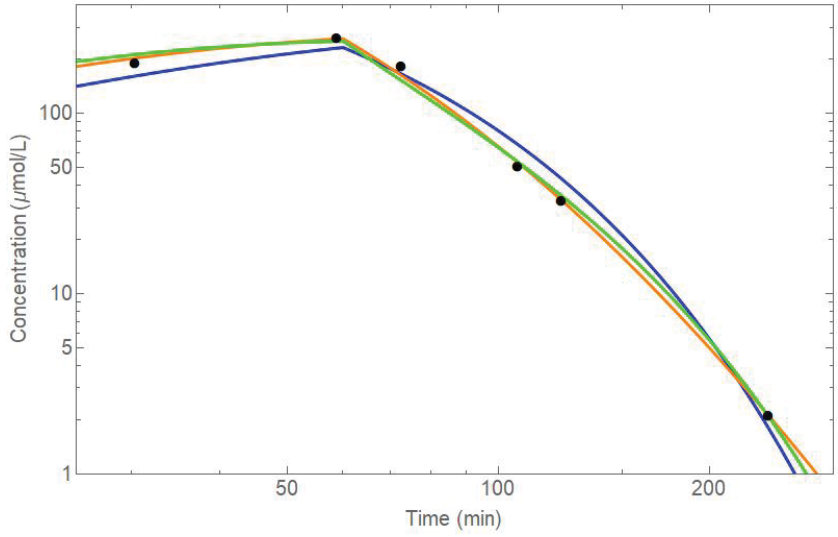



Figure 3.2: Select models of P80 shown with the calibration data. The calibration clinical data is given in Tbl. 3.1 and the models of P80 are given in Tbl. 3.3. The paper is represented as ● Baker et al. [54], and the models as — First, — Fractal, and — Mixed Order.

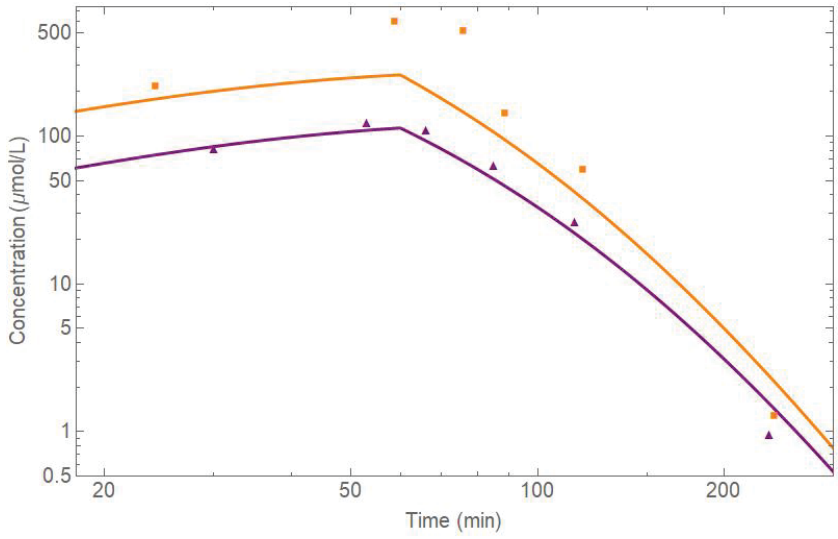



Figure 3.3: The fractal model of P80 shown with the evaluation data. The evaluation clinical data is given in Tbl. 3.2 and the fractal order elimination model parameters are given in Tbl. 3.3. The papers are represented as ▲ Baker et al. [56]a, ■ Baker et al. [56]b, and the model, with the corresponding curves for infusions, are represented as — 0.619 µmol $\cdot m^{-2} \cdot min^{-1}$ over 60 min, and — 1.548 µmol $\cdot m^{-2} \cdot min^{-1}$ over 60 min.

### 3.2.2 Determining DOC Parameters

The clinical papers used in the determination of DOC parameters are given in Tbl. 3.5. The accuracy of the best performing model can then be checked against the clinical data from the papers given in Tbl. 3.6 in order to evaluate the predictive power of the model.

To determine the parameters for the one-compartment DOC model, the calibration data was curve stripped and the results were used to initialize the fits for the individual data sets to the model independently. The calibration data was then fit collectively, using the result of each independent fit as the initialization. By first fitting the data individually, a variety of potential parameters can be captured before being tested against all of the evaluation data. This allows the parameter space to be more thoroughly searched for minima and gives a greater chance at finding the global minimum. The parameters which best minimized the variance for all of the calibration data were then selected as the final set of parameters for each model. These parameters are presented in Tbl. 3.7. The results for the statistical tests which are computed between the model and the calibration data can be found in Tbl. 3.8.

Table 3.5: Clinical papers used in the calibration of DOC model parameters.

| Reference | Dosage (mg/m$^2$) | $T_{inf}$ (min) | $n_p$ | $n_d$ |
|---|---|---|---|---|
| Lim et al. [58] | 75 | 60 | 51 | 5 |
| Robert et al. [59] | 75 | 60 | 19 | 10 |
| Brunsvig et al. [60] | 20 | 60 | 15 | 11 |
| Witta et al. [61]a | 30 | 30 | 13 | 10 |
| Cox et al. [62]a | 30 | 60 | 10 | 11 |
| Baker et al. [54] | 75 | 60 | 1 | 10 |

The general one-compartment one-molecule model used in the determination of DOC

parameters is depicted in Fig. 3.4 and given by

$$\frac{dX_{DOC,1}}{dt} = I_{DOC} - \sum_{p=0}^{2} \left\{ \frac{\overline{k}^{(1)}_{p,1,0} X^{A^{(1)}_{p,1,0}}_{DOC,1}}{1 + \Gamma^{(1)}_{p,1,0} X^{B^{(1)}_{p,1,0}}_{DOC,1}} \right\}. \tag{3.2}$$

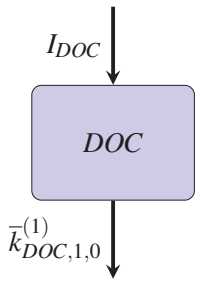



Figure 3.4: Single compartment model of DOC.

Table 3.6: Clinical papers used in the evaluation of DOC model accuracy.

| Reference | Dosage (mg/m$^2$) | $T_{inf}$ (min) | $n_p$ | $n_d$ |
|---|---|---|---|---|
| Awada et al. [63] | 75 | 60 | 6 | 11 |
| Baker et al. [56]a | 30 | 60 | 1 | 10 |
| Baker et al. [56]b | 75 | 60 | 1 | 12 |
| Baker et al. [64]a | 35 | 30 | 8 | 9 |
| Baker et al. [64]b | 75 | 60 | 9 | 11 |
| Blagden et al. [65]a | 60 | 60 | 15 | 4 |
| Blagden et al. [65]b | 75 | 60 | 6 | 4 |
| Clarke et al. [32]a | 100 | 60 | 1 | 14 |
| Clarke et al. [32]b | 100 | 60 | 1 | 13 |
| Clarke et al. [32]c | 100 | 90 | 1 | 16 |
| Clarke et al. [32]d | 100 | 60 | 1 | 17 |
| Clarke et al. [32]e | 100 | 60 | 1 | 16 |

Table 3.6: (continued) Clinical papers used in the evaluation of DOC model accuracy.

| Reference | Dosage (mg/m$^2$) | $T_{inf}$ (min) | $n_p$ | $n_d$ |
|---|---|---|---|---|
| Cox et al. [62]b | 30 | 60 | 10 | 11 |
| Cox et al. [62]c | 30 | 60 | 10 | 11 |
| El-Rayes et al. [66] | 60 | 60 | 6 | 5 |
| Witta et al. [61]b | 36 | 30 | 6 | 10 |

Table 3.7: One-compartment DOC model numerical results.

| Parameters* | Fractal | Fractal + First | Saturable |
|---|---|---|---|
| $\bar{k}^{(1)}_{0,1,0}$ | 0 | 0 | 0 |
| $\bar{k}^{(1)}_{1,1,0}$ | 3.53(45) | 85(13) | 0.2593(14) |
| $\bar{k}^{(1)}_{2,1,0}$ | 0 | 0.00045(10) | 0 |
| $A^{(1)}_{1,1,0}$ | 2.605(32) | 2.977(30) | 2.6244(40) |
| $A^{(1)}_{2,1,0}$ | - | 1 | - |
| $\Gamma^{(1)}_{1,1,0}$ | 0 | 0 | 0.06989(45) |
| $\Gamma^{(1)}_{2,1,0}$ | - | 0 | - |
| $B^{(1)}_{1,1,0}$ | - | - | 3.1479(47) |
| $V_{D,(DOC,1)}$ | 0.301(22) | 0.1084(75) | 1.59(11) |

* The units for $\bar{k}_{p,c,c'}$ are $[(\frac{1}{\mathrm{s}})(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{1-A_{p,c,c'}}]$ , the units for $\Gamma_{p,c,c'}$ are $[(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{1-B_{p,c,c'}}]$, and the units for $V_{D,(DOC,1)}$ are $[\frac{\mathrm{L}}{\mathrm{m}^2}]$. All other parameters are dimensionless.

The highest performing one-compartment models of DOC are the fractal model, a fractal model with a first order correction, and the saturable model. Of these, the best performing model is the saturable model. Further parameterization of the one-compartment one-molecule models of DOC resulted in larger AIC$_c$ values. The statistics for additional

models which were tested but gave poor results can be found in §B.1.2. Each of the models given in Tbl. 3.7 are compared against the most commonly administered dosage in the available data in Fig. 3.5. Note that this figure does not include all of the data used for fitting and serves only to show the behaviour of the models in comparison to a commonly administered dosage. Additional figures showing the remaining data sets compared to the evaluation and calibration data can be found in §A.1.2 and §A.1.1 respectively. At a dose of 75 mg/m$^2$ for 60 min these models follow the general trend of the data with the saturable model doing the best job of capturing $C_{max}$. However at other doses the saturable model is far less accurate at predicting DOC concentrations. This is particularly true at the largest dose of 100 mg/m$^2$ over 60 min as shown in Fig. A.9.

Table 3.8: One-compartment DOC model statistics.

| Statistic | Fractal | Fractal + First | Saturable |
|---|---|---|---|
| $S_p$ | 0.3442 | 0.3275 | 0.3192 |
| AIC$_c$ | -921.0 | -962.3 | -982.5 |
| $\Delta^*$ | 611 | 570 | 550 |

* Relative to the three-compartment one molecule fractal MM model in Tbl. 3.13.

The saturable model is compared to all of the calibration data within Fig. 3.6. The fit of this model follows the general trend of the data far better than most of the one-compartment models but grossly overestimates the concentration for Witta et al. [61] and underestimates the $C_{max}$ of Cox et al. [62]a. This model is unable to capture DOC PKs across the range of administered doses.

## 3.3 One-Compartment Two-Molecule Models

DOC is encapsulated in P80 when administered. The influence of P80 on the modelling of DOC must be considered by introducing DOC to the one-compartment P80 model de-

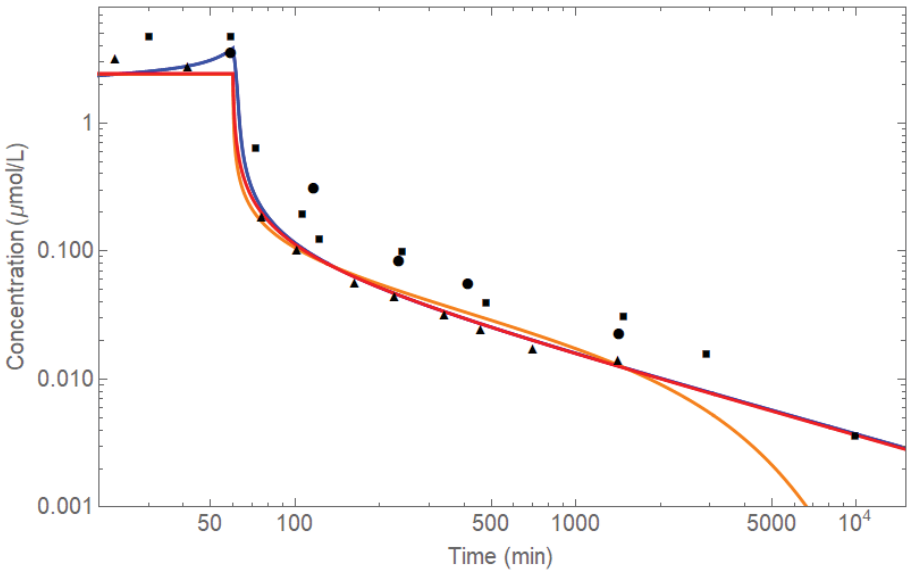



Figure 3.5: Comparison between the one-compartment one-molecule models of DOC and a sample of calibration data. The calibration data used is the data given in Tbl. 3.5 with a dosage of 75 mg/m$^2$ over 60 min. The one-compartment one-molecule model parameters are given in Tbl. 3.7. ● Lim et al. [58], ▲ Robert et al. [59], ■ Baker et al. [54], — Fractal, — Fractal+First, and — Saturable.

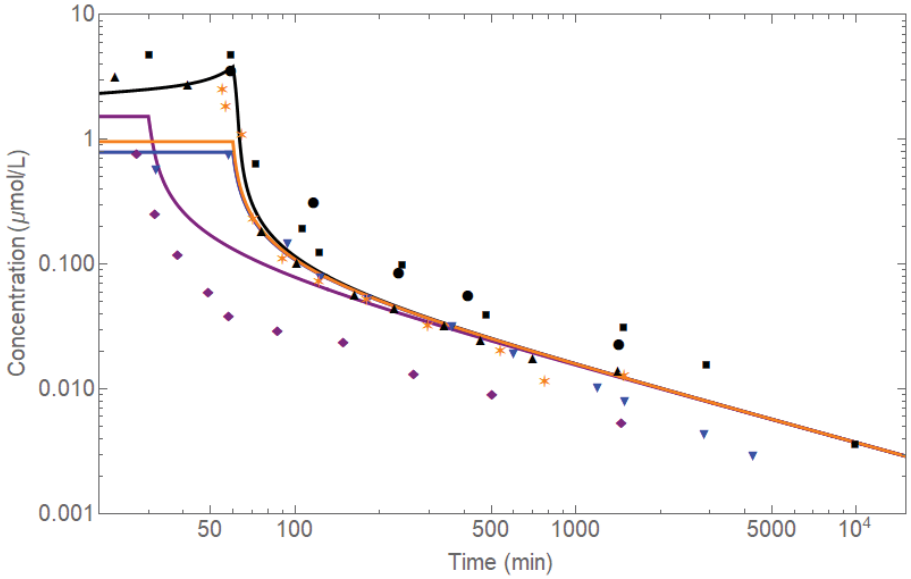



Figure 3.6: Comparison between the one-compartment one-molecule saturable model of DOC and the calibration data. The calibration data is given in Tbl. 3.5 and the one-compartment one-molecule saturable model parameters are given in Tbl. 3.7. ● Lim et al. [58], ▲ Robert et al. [59], ■ Baker et al. [54], ◆ Witta et al. [61]a, ▼ Brunsvig et al. [60], ✶ Cox et al. [62]a. — 30 mg/m$^2$ over 30 min, — 20 mg/m$^2$ over 60 min, — 60 mg/m$^2$ over 60 min, and — 75 mg/m$^2$ over 60 min.

veloped in §3.2.1. In this model, an infusion of P80 is delivered to the main compartment. DOC is then released by P80, after which it can be eliminated. These processes all take place within the same compartment. The general one-compartment two-molecule model used in the determination of DOC parameters is depicted in Fig. 3.7 and given by

$$\frac{dX_{P80,1}}{dt} = I_{P80} - \frac{\bar{k}^{(2)}_{1,s} X^{A^{(2)}_{1,s}}_{P80,1}}{1+\Gamma^{(2)}_{1,s} X^{B^{(2)}_{1,s}}_{P80,1}}, \tag{3.3}$$

$$\frac{dX_{DOC,1}}{dt} = \frac{\delta_{1,s}\bar{k}^{(2)}_{1,s} X^{A^{(2)}_{1,s}}_{P80,1}}{1+\Gamma^{(2)}_{1,s} X^{B^{(2)}_{1,s}}_{P80,1}} - \sum_{p=0}^{2}\left\{\frac{\bar{k}^{(1)}_{p,1,0} X^{A^{(1)}_{p,1,0}}_{DOC,1}}{1+\Gamma^{(1)}_{p,1,0} X^{B^{(1)}_{p,1,0}}_{DOC,1}}\right\}. \tag{3.4}$$

where the subscript $s = DOC, 1, P80, 1$.

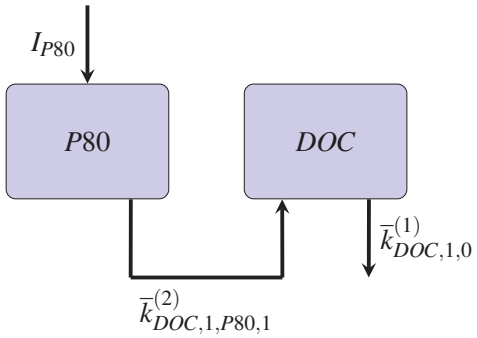



Figure 3.7: The one-compartment two-molecule model of P80 and DOC. The parameters associated with $\bar{k}^{(2)}_{DOC,1,P80,1}$ are fixed and given in Tbl. 3.3, while the parameters associated with $\bar{k}^{(1)}_{DOC,1,0}$ are permitted to vary.

The fractal model developed for P80 as given in Tbl. 3.3 was used to describe the PKs of P80. Thus, the transformation process is fixed with:

$\bar{k}^{(2)}_{1,s} = 0.014956(47)$ $((\frac{1}{\mathrm{s}})(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{1-A_{1,s}})$, and

$A^{(2)}_{1,s} = 1.14623(58)$.

The remaining parameters of Eqn. 3.4 are then fit using the results of Tbl. 3.7 as the initialization. The results of the best fits are presented in Tbl. 3.9. The statistical tests which are computed between the model and the calibration data in Tbl. 3.5 can be found in Tbl. 3.10. It is very clear however, that the inclusion of P80 dramatically affects the

parameters which describe the elimination of DOC. The quality of all of the fits has been reduced. Comparing the saturable models from both the one-compartment one-molecule and the one-compartment two-molecule cases we find a 44.9% increase in the variance. We also note that adding a first order correction to the fractal model no longer improves the fit in any way. The saturable model also has $\delta > 1$ which is unphysical as it implies that more DOC than P80 is produced by the transformation. The statistics for additional models which were tested can be found in §B.2.

Table 3.9: One-compartment two-molecule DOC model numerical results.

| Parameters* | Fractal | Fractal + First | Saturable |
|---|---|---|---|
| $\delta^{(2)}_{1,s}$ | 0.0086(49) | 0.0086(49) | 124(16) |
| $\bar{k}^{(1)}_{0,1,0}$ | 0 | 0 | 0 |
| $\bar{k}^{(1)}_{1,1,0}$ | 0.66(16) | 0.66(16) | $2.57(35)\times10^5$ |
| $\bar{k}^{(1)}_{2,1,0}$ | 0 | 0.00000(18) | 0 |
| $A^{(1)}_{1,1,0}$ | 2.618(73) | 2.618(73) | 6.331(82) |
| $A^{(1)}_{2,1,0}$ | - | 1 | - |
| $\Gamma^{(1)}_{1,1,0}$ | 0 | 0 | 530(68) |
| $\Gamma^{(1)}_{2,1,0}$ | - | 0 | - |
| $B^{(1)}_{1,1,0}$ | - | - | 5.583(76) |
| $V_{D,DOC,1}$ | 1.30(15) | 1.30(15) | 3.75(31) |

* The units for $\bar{k}_{p,c,c'}$ are $[(\frac{1}{\mathrm{s}})(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{1-A_{p,c,c'}}]$ , the units for $\Gamma_{p,1}$ are $[(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{1-B_{p,c,c'}}]$, and the units for $V_{D,(DOC,1)}$ are $[\frac{\mathrm{L}}{\mathrm{m}^2}]$. All other parameters are dimensionless.

As with the one-compartment one-molecule models of DOC, each of the models given in Tbl. 3.9 are compared against the most commonly administered dosage in the available data in Fig. 3.8. The fractal model significantly underestimates $C_{max}$ and its poor fit is reflected in its high variance. The saturable model better captures the behaviour of DOC at

high concentrations but, as explained above, this improvement comes at the cost of physical consistency in the model.

Table 3.10: One-compartment two-molecule DOC model statistics.

| Statistic | Fractal | Fractal + First | Saturable |
|---|---|---|---|
| $S_p$ | 0.6657 | 0.6657 | 0.4489 |
| $AIC_c$ | -344.4 | -342.4 | -683.6 |
| $\Delta^*$ | 1188 | 1190 | 849 |

* Relative to the three-compartment one molecule fractal MM model in Tbl. 3.13.

The saturable model is compared to all of the calibration data in Fig. 3.9. The fits have problems similar to, but more pronounced than, those shown in Fig. 3.6. Witta et al. [61]a is very poorly described and the $C_{max}$ is not well captured for the majority of the infusions. Additional figures showing the remaining data sets compared to the evaluation and calibration data can be found in §A.2.2 and §A.2.1 respectively.

Collectively, the results of these models implies that DOC dissociates with P80 nearly instantaneously after infusion and the presence of P80 does not influence the PKs of DOC. This result is consistent with the current understanding of DOC PKs [21]. Because of the poor quality of the one-compartment two-molecule model fits, P80 will not be included in the development of models with more compartments.

## 3.4 Two-Compartment One-Molecule Models

The two-compartment model introduces a peripheral compartment which a molecule may flow into and back out of. The molecule can also be eliminated from either compartment. This model was developed for DOC alone. The general two-compartment one-molecule model used in the determination of DOC parameters is depicted in Fig. 3.10 and

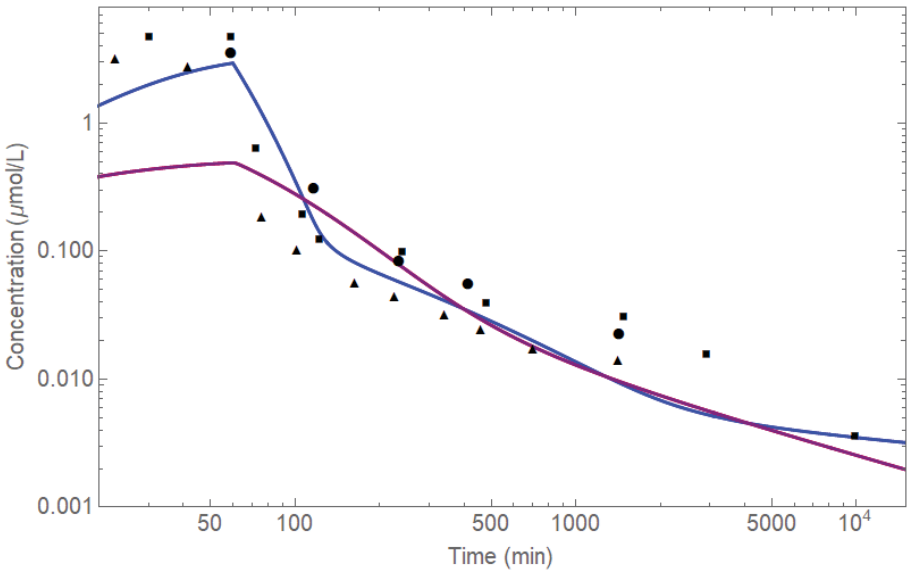



Figure 3.8: Comparison of one-compartment two-molecule models of DOC and a sample of calibration data. The calibration data used is the data given in Tbl. 3.5 with a dosage of 75 mg/m$^2$ over 60 min. The one-compartment two-molecule model parameters are given in Tbl. 3.7. ● Lim et al. [58], ▲ Robert et al. [59], ■ Baker et al. [54], — Fractal, — Fractal+First, and — Saturable.

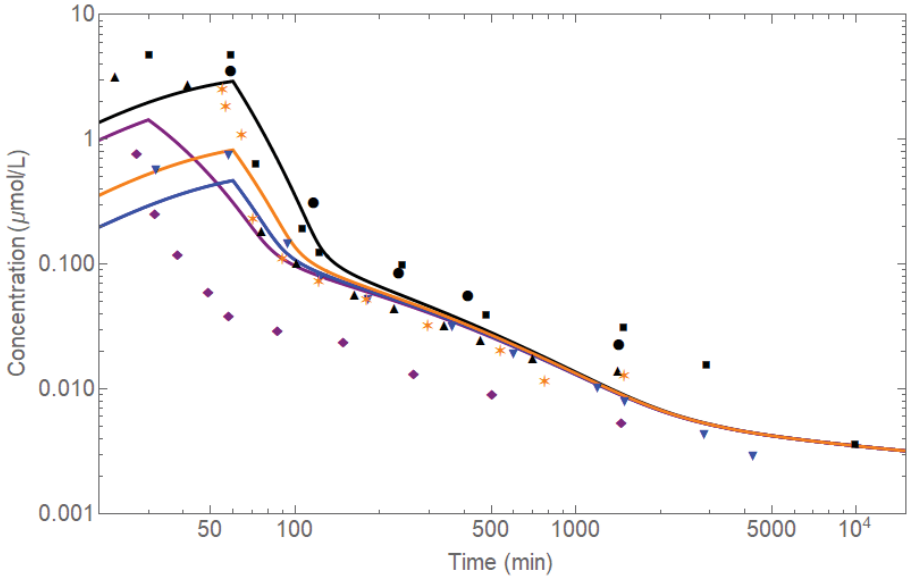



Figure 3.9: Comparison between the one-compartment two-molecule saturable model of DOC and the calibration data. The calibration data given in Tbl. 3.5 and the one-compartment two-molecule saturable model parameters are given in Tbl. 3.7. ● Lim et al. [58], ▲ Robert et al. [59], ■ Baker et al. [54], ◆ Witta et al. [61]a, ▼ Brunsvig et al. [60], ✶ Cox et al. [62]a. — 30 mg/m$^2$ over 30 min, — 20 mg/m$^2$ over 60 min, — 60 mg/m$^2$ over 60 min, and — 75 mg/m$^2$ over 60 min.

given by

$$\frac{dX_{DOC,1}}{dt} = I_{DOC} - \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,1,2}X^{A^{(1)}_{p,1,2}}_{DOC,1}}{1+\Gamma^{(1)}_{p,1,2}X^{B^{(1)}_{p,1,2}}_{DOC,1}}\right\} + \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,2,1}X^{A^{(1)}_{p,2,1}}_{DOC,2}}{1+\Gamma^{(1)}_{p,2,1}X^{B^{(1)}_{p,2,1}}_{DOC,2}}\right\} - \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,1,0}X^{A^{(1)}_{p,1,0}}_{DOC,1}}{1+\Gamma^{(1)}_{p,1,0}X^{B^{(1)}_{p,1,0}}_{DOC,1}}\right\}, \tag{3.5}$$

$$\frac{dX_{DOC,2}}{dt} = \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,1,2}X^{A^{(1)}_{p,1,2}}_{DOC,1}}{1+\Gamma^{(2)}_{p,1,2}X^{B^{(1)}_{p,1,2}}_{DOC,1}}\right\} - \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,2,1}X^{A^{(1)}_{p,2,1}}_{DOC,2}}{1+\Gamma^{(1)}_{p,2,1}X^{B^{(1)}_{p,2,1}}_{DOC,2}}\right\} - \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,2,0}X^{A^{(1)}_{p,2,0}}_{DOC,2}}{1+\Gamma^{(1)}_{p,2,0}X^{B^{(1)}_{p,2,0}}_{DOC,2}}\right\}. \tag{3.6}$$

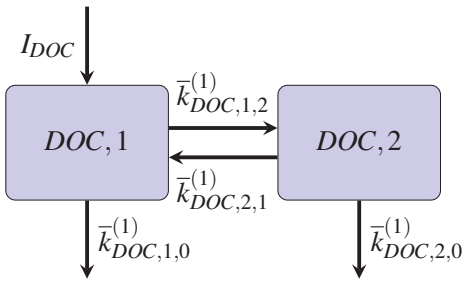



Figure 3.10: The two-compartment model of DOC.

To determine the parameters of the two-compartment DOC model the results of the corresponding one-compartment DOC was used to initialize the parameters which control the flow of DOC from compartment one into compartment two. This process is labelled as $\overline{k}^{(1)}_{DOC,1,2}$ in Fig. 3.10. Then, as with the one-compartment DOC model, the individual calibration data sets are fit and those results are used to initialize the collective fits. The final parameters are given in Tbl. 3.11. The statistical tests which are computed between the models and the calibration data in Tbl. 3.5 can be found in Tbl. 3.12.

All tested two-compartment models performed significantly better than any of the one-compartment models. The highest performing two-compartment models of DOC are the

fractal model, and the fractal MM model. Of these, the best performing model is the fractal MM model. The statistics for additional models which were also tested can be found in §B.3.

By including an additional compartment there is a 1.7× reduction in the variance when compared to the best one-compartment model of DOC. Moreover, the additional parameterization introduced by the second compartment does not lead to increased $\mathrm{AIC}_c$ values relative to the one-compartment DOC models.

Table 3.11: Two-compartment one-molecule DOC model numerical results.

| Parameters* | Fractal | Fractal MM |
|---|---|---|
| $\overline{k}^{(1)}_{0,1,2}$ | 0 | 0 |
| $\overline{k}^{(1)}_{1,1,2}$ | 0.437(31) | 24.7(20) |
| $\overline{k}^{(1)}_{2,1,2}$ | 0 | 0 |
| $A^{(1)}_{1,1,2}$ | 1.208(23) | 1.929(26) |
| $\Gamma^{(1)}_{1,1,2}$ | 0 | 16.2(19) |
| $B^{(1)}_{1,1,2}$ | - | 1 |
| $\overline{k}^{(1)}_{0,2,1}$ | 0 | 0 |
| $\overline{k}^{(1)}_{1,2,1}$ | 0.000258(22) | 0.0000303(36) |
| $\overline{k}^{(1)}_{2,2,1}$ | 0 | 0 |
| $A^{(1)}_{1,2,1}$ | 1.307(29) | 1.945(37) |
| $\Gamma^{(1)}_{1,2,1}$ | 0 | 0.0000(40) |
| $B^{(1)}_{1,2,1}$ | - | 1 |
| $\overline{k}^{(1)}_{0,2,0}$ | 0 | 0 |
| $\overline{k}^{(1)}_{1,2,0}$ | 0.000139(13) | 0.0000906(80) |
| $\overline{k}^{(1)}_{2,2,0}$ | 0 | 0 |
| $A^{(1)}_{1,2,0}$ | 1.989(31) | 2.329(28) |

Table 3.11: (continued) Two-compartment one-molecule DOC model numerical results.

| Parameters* | Fractal | Fractal MM |
|---|---|---|
| $\Gamma^{(1)}_{1,2,0}$ | 0 | 0.0251(56) |
| $B^{(1)}_{1,2,0}$ | - | 1 |
| $V_{D,(DOC,1)}$ | 1.065(66) | 0.374(22) |

* The units for $\overline{k}_{p,c,c'}$ are $[(\frac{1}{\mathrm{s}})(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{1-A_{p,c,c'}}]$ , the units for $\Gamma_{p,c,c'}$ are $[(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{1-B_{p,c,c'}}]$, and the units for $V_{D,(DOC,1)}$ are $[\frac{\mathrm{L}}{\mathrm{m}^2}]$. All other parameters are dimensionless.

Table 3.12: Two-compartment one-molecule DOC model statistics.

| Statistic | Fractal | Fractal MM |
|---|---|---|
| $S_p$ | 0.2734 | 0.2593 |
| $\mathrm{AIC}_c$ | -1113 | -1154 |
| $\Delta^*$ | 419 | 379 |

* Relative to the three-compartment one molecule fractal MM model in Tbl. 3.13.

The models from Tbl. 3.11 are compared against the most commonly administered dosage in the available data in Fig. 3.11. There is not a significant difference between the behaviour of these models which is reflected in their variances. Note that this figure does not include all of the data used for fitting and serves only to show the behaviour of the models in comparison to the most commonly administered dosage.

The fractal MM model is compared to all of the calibration data in Fig. 3.13. While the fit has been improved we see similar discrepancies to the fits for Witta et al. [61]a and Cox et al. [62] as we did in the one-compartment one-drug saturable model. Additional figures showing the remaining data sets compared to the evaluation and calibration data can be found in §A.3.2 and §A.3.1 respectively.

The behaviour of these models is distinctly biphasic which is expected for a two-compartment model. Fig. 3.13 shows that the sharp phase transition causes the model

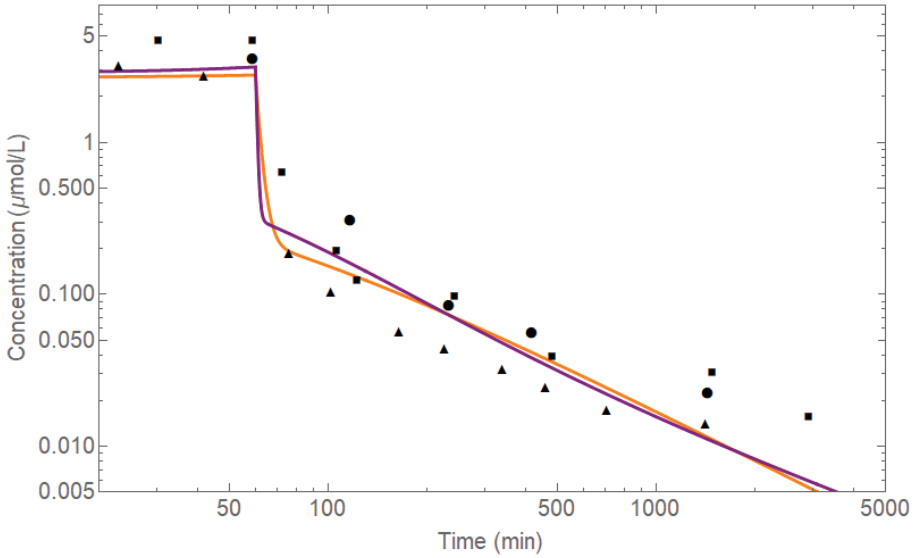



Figure 3.11: Comparison between the two-compartment one-molecule models of DOC and a sample of calibration data. The calibration data used is the data given in Tbl. 3.5 with a dosage of 75 mg/m$^2$ over 60 min. The two-compartment one-molecule model parameters are given in Tbl. 3.11. ● Lim et al. [58], ▲ Robert et al. [59], ■ Baker et al. [54], — Fractal, and — Fractal MM.

to inaccurately predict the concentration just after the end of the infusion. This suggests that further compartmentalization will further improve the fit as the additional degrees of freedom smooth out the transition.

## 3.5 Three-Compartment One-Molecule Models

Previously developed compartmental models of DOC have most commonly used three compartments to describe the PKs of DOC. These models will traditionally permit only first-order processes to describe the flow between the compartments.

The general three-compartment one-molecule model used in the determination of DOC

parameters is depicted in Fig. 3.12 and given by

$$\frac{dX_{DOC,1}}{dt} = I_{DOC} - \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,1,2}X^{A^{(1)}_{p,1,2}}_{DOC,1}}{1+\Gamma^{(1)}_{p,1,2}X^{B^{(1)}_{p,1,2}}_{DOC,1}}\right\} + \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,2,1}X^{A^{(1)}_{p,2,1}}_{DOC,2}}{1+\Gamma^{(1)}_{p,2,1}X^{B^{(1)}_{p,2,1}}_{DOC,2}}\right\} - \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,1,3}X^{A^{(1)}_{p,1,3}}_{DOC,1}}{1+\Gamma^{(1)}_{p,1,3}X^{B^{(1)}_{p,1,3}}_{DOC,1}}\right\} + \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,3,1}X^{A^{(1)}_{p,3,1}}_{DOC,3}}{1+\Gamma^{(1)}_{p,3,1}X^{B^{(1)}_{p,3,1}}_{DOC,3}}\right\}, \tag{3.7}$$

$$\frac{dX_{DOC,2}}{dt} = \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,1,2}X^{A^{(1)}_{p,1,2}}_{DOC,1}}{1+\Gamma^{(1)}_{p,1,2}X^{B^{(1)}_{p,1,2}}_{DOC,1}}\right\} - \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,2,1}X^{A^{(1)}_{p,2,1}}_{DOC,2}}{1+\Gamma^{(1)}_{p,2,1}X^{B^{(1)}_{p,2,1}}_{DOC,2}}\right\}, \tag{3.8}$$

$$\frac{dX_{DOC,3}}{dt} = \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,1,3}X^{A^{(1)}_{p,1,3}}_{DOC,1}}{1+\Gamma^{(1)}_{p,1,3}X^{B^{(1)}_{p,1,3}}_{DOC,1}}\right\} - \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,3,1}X^{A^{(1)}_{p,3,1}}_{DOC,3}}{1+\Gamma^{(1)}_{p,3,1}X^{B^{(1)}_{p,3,1}}_{DOC,3}}\right\} - \sum_{p=0}^{2}\left\{\frac{\overline{k}^{(1)}_{p,3,0}X^{A^{(1)}_{p,3,0}}_{DOC,3}}{1+\Gamma^{(1)}_{p,3,0}X^{B^{(1)}_{p,3,0}}_{DOC,3}}\right\}. \tag{3.9}$$

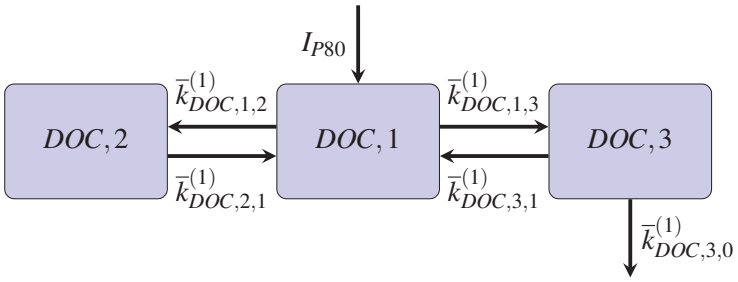



Figure 3.12: The three-compartment model of DOC.

To determine the parameters of the three-compartment DOC model the results of the corresponding two-compartment DOC are used to initialize the parameters which control the flow of DOC between compartment one and three as well as the elimination of DOC from compartment three. That is, the processes labelled $\overline{k}^{(1)}_{DOC,1,3}$, $\overline{k}^{(1)}_{DOC,3,1}$ and $\overline{k}^{(1)}_{DOC,3,0}$ in Fig. 3.12 were initialized using two-compartment DOC results. The additional parametrization within the three-compartment modelling means that most models will over fit the individual calibration data. Individual fits were used to generate initializations for the collective

fits where possible but otherwise the two-compartment results were used. The final parameters are given in Tbl. 3.13.

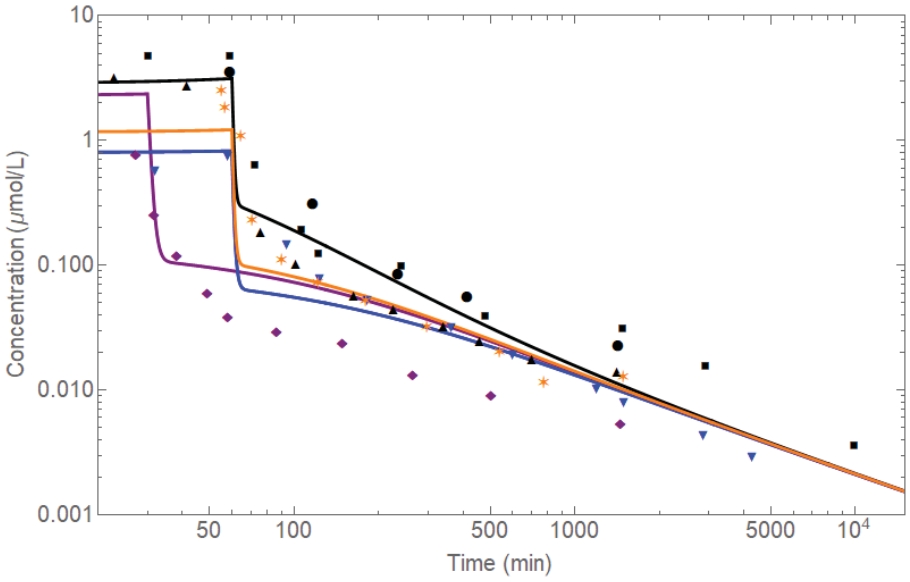



Figure 3.13: Comparison between the two-compartment one-molecule fractal MM model of DOC and the calibration data. The calibration data is given in Tbl. 3.5 and the two-compartment one-molecule fractal MM model parameters are given in Tbl. 3.11. ● Lim et al. [58], ▲ Robert et al. [59], ■ Baker et al. [54], ◆ Witta et al. [61]a, ▼ Brunsvig et al. [60], ✶ Cox et al. [62]a. — 30 mg/m$^2$ over 30 min, — 20 mg/m$^2$ over 60 min, — 60 mg/m$^2$ over 60 min, and — 75 mg/m$^2$ over 60 min.

Table 3.13: Three-compartment one-molecule DOC model parameter results.

| Parameters* | Mixed Saturable | Fractal | Fractal MM |
|---|---|---|---|
| $\overline{k}^{(1)}_{1,1,2}$ | 0.2090(88) | 0.140(16) | 0.636(58) |
| $A^{(1)}_{1,1,2}$ | 1 | 0.1664(91) | 2.28(12) |
| $\Gamma^{(1)}_{1,1,2}$ | 0 | 0 | 0.86(17) |
| $B^{(1)}_{1,1,2}$ | - | - | 1 |
| $\overline{k}^{(1)}_{1,2,1}$ | 0.000080(13) | 0.00470(26) | 0.00043(21) |
| $A^{(1)}_{1,2,1}$ | 1 | 1.715(29) | 1.234(11) |
| $\Gamma^{(1)}_{1,2,1}$ | 0 | 0 | 0.197(10) |

Table 3.13: (continued) Three-compartment one-molecule DOC model parameter results.

| Parameters* | Mixed Saturable | Fractal | Fractal MM |
|---|---|---|---|
| $B^{(1)}_{1,2,1}$ | - | - | 1 |
| $\overline{k}^{(1)}_{1,1,3}$ | 0.594(46) | 0.396(25) | 23.4(11) |
| $A^{(1)}_{1,1,3}$ | 0.1804(96) | 1.592(32) | 1.1398(67) |
| $\Gamma^{(1)}_{1,1,3}$ | 0.073(21) | 0 | 138(10) |
| $B^{(1)}_{1,1,3}$ | 0.60(15) | - | 1 |
| $\overline{k}^{(1)}_{1,3,1}$ | 0.0271(14) | 0.00118(23) | 0.00522(20) |
| $A^{(1)}_{1,3,1}$ | 1 | 0.959(63) | 1.699(25) |
| $\Gamma^{(1)}_{1,3,1}$ | 0 | 0 | 0.0000(80) |
| $B^{(1)}_{1,3,1}$ | - | - | 1 |
| $\overline{k}^{(1)}_{1,3,0}$ | 0.000004(29) | 0.0961(88) | 0.001532(55) |
| $A^{(1)}_{1,3,0}$ | 1 | 0.571(31) | 2.281(23) |
| $\Gamma^{(1)}_{1,3,0}$ | 0 | 0 | 1.380(59) |
| $B^{(1)}_{1,3,0}$ | - | - | 1 |
| $V_{D,(DOC,1)}$ | 1.586(77) | 0.844(42) | 0.875(38) |

* The units for $\overline{k}_{p,c,c'}$ are $[(\frac{1}{\mathrm{s}})(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{1-A_{p,c,c'}}]$ , the units for $\Gamma_{p,c,c'}$ are $[(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{1-B_{p,c,c'}}]$, and the units for $V_{D,(DOC,1)}$ are $[\frac{\mathrm{L}}{\mathrm{m}^2}]$. All other parameters are dimensionless.

Table 3.14: Three-compartment one-molecule DOC model statistics.

| Statistic | Mixed Saturable | Fractal | Fractal MM |
|---|---|---|---|
| $S_p$ | 0.1914 | 0.1901 | 0.1654 |
| $\mathrm{AIC}_c$ | -1420 | -1422 | -1533 |
| $\Delta^*$ | 113 | 110 | 0 |

The three-compartment models tested made significant improvements to the model ac-

curacy compared to the one- and two-compartment models. The highest performing three-compartment models of DOC are the mixed saturable model, the fractal model, and the fractal MM model. Of these, the best performing model is the fractal MM model. The statistics for additional models which were also tested can be found in §B.4. Each of the models listed above are given in Tbl. 3.13. The statistical tests which are computed between the models and the calibration data in Tbl. 3.5 can be found in Tbl. 3.14. The variance of the fractal MM model is $1.6\times$ lower than the best performing two-compartment model which is similar to the improvement when going from one-compartment to two-compartments. Also similar is the drop in the $\mathrm{AIC}_c$ values. The fractal MM model has the smallest $\mathrm{AIC}_c$ value of all models tested and therefore $\mathrm{AIC}_{c,min} = -1533$. We do however note that there is a strong potential that further parametrization of the model could improve the $\mathrm{AIC}_c$.

The models given in Tbl. 3.13 are compared against the most commonly administered dosage in the available data in Fig. 3.14. Note that Fig. 3.14 does not include all of the data used for fitting and serves only to show the behaviour of the models in comparison to the most commonly administered dosage. The fractal MM model is compared to all of the calibration data in Fig. 3.15. Each of these models does a much better job at predicting the concentrations of DOC at both high and low concentrations than the previously tested models with the fractal MM curve fitting the data most accurately.

The three-compartment fractal MM model does a significantly better job of capturing the behaviour for all of the different dosage levels within a single model as can be seen in Fig. 3.15. This is reflected in the $\mathrm{AIC}_c$ and the variance. The three-compartment fractal model's variance is 14.9% lower than the next best model and the $\mathrm{AIC}_c$ of every other model is $\gg 10$ suggesting a high degree of empirical evidence for this model. We may note the behaviour of Witta et al. [61]a is much better described than in the best performing one-and two-compartment models and the $C_{max}$ of Cox et al. [62] is much more closely approximated, though there are still considerable discrepancies in both cases.

The fractal MM model, is compared to a sample of the evaluation data corresponding

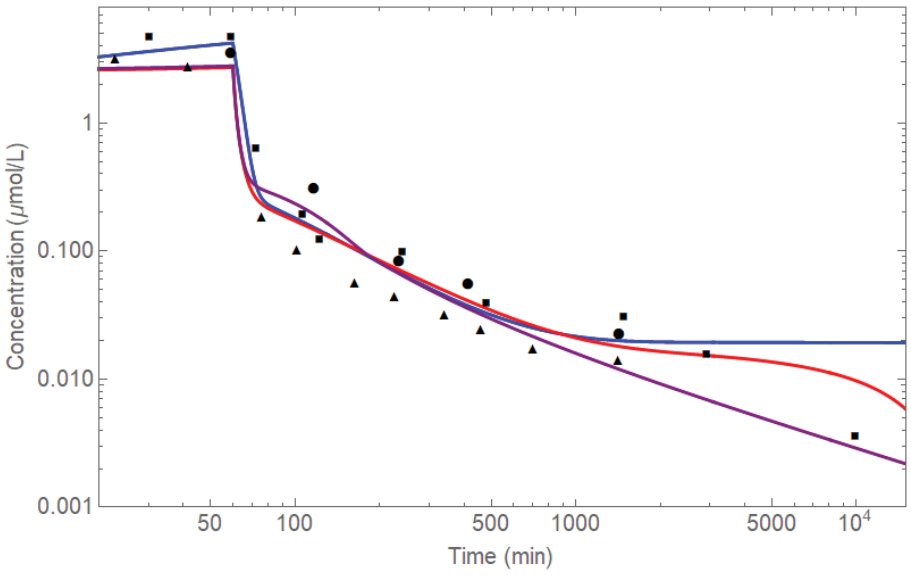



Figure 3.14: Comparison between the three-compartment one-molecule models of DOC and a sample of calibration data. The calibration data used is the data given in Tbl. 3.5 with a dosage of 75 mg/m$^2$ over 60 min. The three-compartment one-molecule model parameters are given in Tbl. 3.13. ● Lim et al. [58], ▲ Robert et al. [59], ■ Baker et al. [54], — Mixed Saturable, — Fractal, and — Fractal MM.

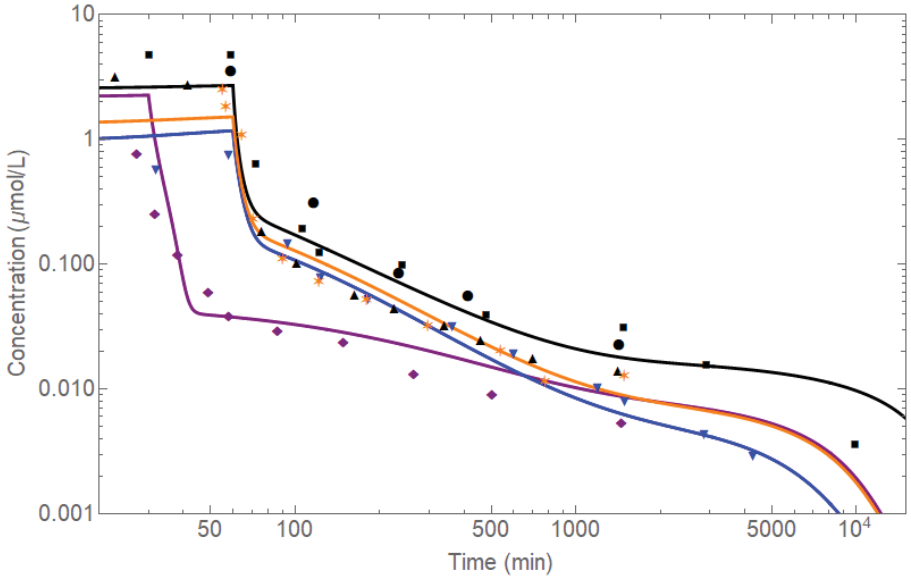



Figure 3.15: Comparison between the three-compartment one-molecule fractal MM model of DOC and the calibration data. The calibration data is given in Tbl. 3.5 and the three-compartment one-molecule fractal MM model parameters are given in Tbl. 3.13. ● Lim et al. [58], ▲ Robert et al. [59], ■ Baker et al. [54], ◆ Witta et al. [61]a, ▼ Brunsvig et al. [60], ✶ Cox et al. [62]a. — 30 mg/m$^2$ over 30 min, — 20 mg/m$^2$ over 60 min, — 60 mg/m$^2$ over 60 min, and — 75 mg/m$^2$ over 60 min.

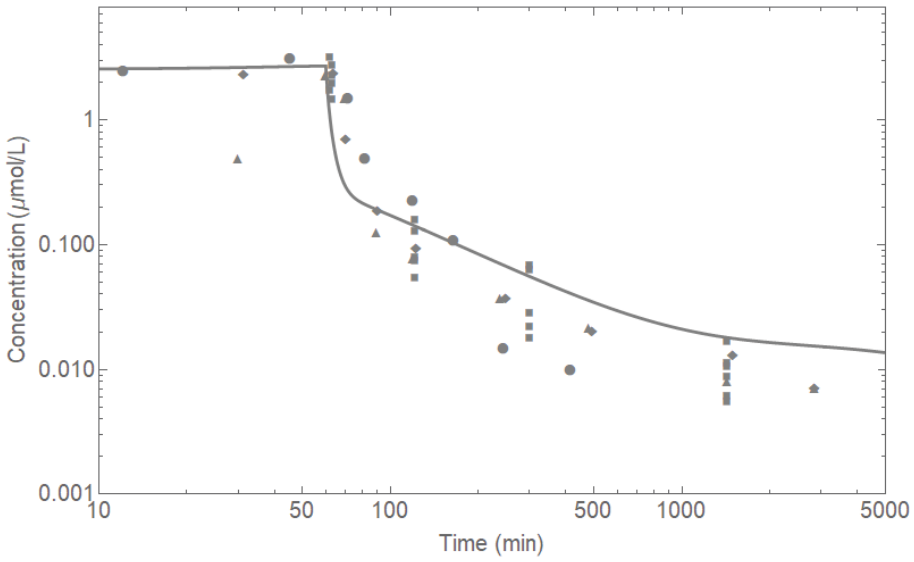



Figure 3.16: Comparison between between the three-compartment one-molecule fractal MM model of DOC and a sample of evaluation data with a dosage of 75 mg/m$^2$ over 60 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the three-compartment one-molecule fractal MM model of DOC given in Tbl. 3.13. ● Awada et al. [63], ▲ Baker et al. [64]b, ■ Blagden et al. [65]b, ◆ Baker et al. [56]b.

to the same dosage as in Fig. 3.16. This figure shows that the fractal MM model can accurately predict the concentrations of DOC though there is a tendency to over estimate the concentration at later times when the concentration is below $\approx 0.08\ \mu\text{mol}/\text{L}$.

Additional figures showing comparisons for the remaining calibration and evaluation data sets can be found in §A.4.1 and §A.4.2 respectively. Of note is Fig. A.37 which includes the evaluation data set Blagden et al. [65]a. Fig. A.37 illustrates the high degree of interpatient variability which makes accurately predicting the concentration of DOC so difficult. However, it is clear that the average patient is very well described by the fractal MM model in this case which suggests that this model can still serve as a valuable tool for evaluating DOC. The predictive power of this model will be further discussed within §4.

## 3.6 Model Selection

As discussed in §2.6, it is the model with the smallest $\text{AIC}_c$ value which should be selected to describe the time course behaviour of DOC within the blood plasma. The three-

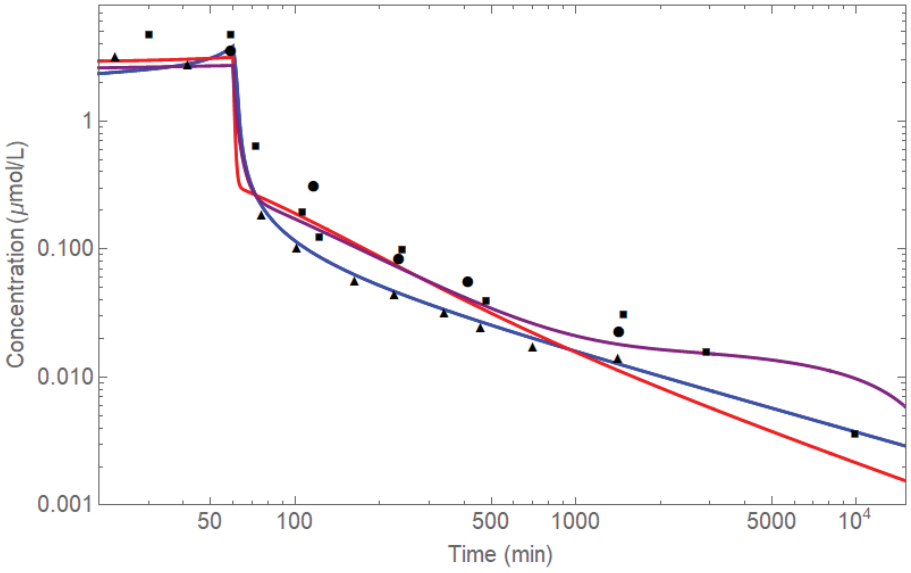



Figure 3.17: Comparison between the best performing one- two- and three-compartment models. The calibration data used is the data given in Tbl. 3.5 with a dosage of 75 mg/m$^2$ over 60 min. ● Lim et al. [58], ▲ Robert et al. [59], ■ Baker et al. [54], — One-compartment, — Two-compartment, and — Three-compartment.

compartment fractal MM model of DOC was selected by the AIC$_c$ with a high enough degree of empirical evidence that all other models of DOC should be discarded. The gradual improvement in the models as compartments are added can be seen in Fig. 3.17. It is this model that will be used to derive the clinically important PK metrics.

# Chapter 4

# Model Applications

*" All models are wrong,*

*but some are useful."*

– George E.P. Box

In the previous chapter, models which simulate the concentration of DOC within the blood plasma were developed. The PK metrics presented within this chapter are all derived from the best performing model, specifically, the three-compartment one-molecule fractal MM model which can be found in Tbl. 3.13. This was the model which showed the greatest level of empirical evidence as determined by the $AIC_c$ (see Eqn. 2.17) with the criterion suggesting all other models should be disregarded.

The PK metrics derived from this model are compared to PK metrics computed from the concentration-time data which was digitized from the literature. Some of the clinical studies used provided their own calculations of the PK metrics alongside the concentration-time data. These values have also been included here. The accuracy of the PK metrics derived from the model are evaluated using $S_p$ (see Eqn. 2.8) as computed between the model and each individual data set with a smaller variance suggesting a more accurate prediction.

We will also study the relationship between each parameter in the three-compartment one-molecule fractal MM model and its sensitivity to the interpatient variability of DOC. The results will inform how this model could be applied to improve patient outcomes by

individualizing treatment.

## 4.1 Evaluation Data

Before covering the PK metrics we will first consider how accurately the selected three-compartment one-molecule fractal MM model is able to predict the concentration of DOC. This is done by examining the variance. Collectively, the variance computed for the calibration data (Tbl. 3.5) was $S_p = 0.1654$, but we can also consider how well each individual data set is described. The variance for each calibration data set is given in Tbl. 4.1. Of the calibration data the most accurately predicted is Brunsvig et al. [60] and the least accurately predicted is Witta et al. [61]a.

Table 4.1: Metrics computed for the DOC calibration data compared to those derived from the clinical concentration-time data.

| Dose ($mg/m^2$) | $T_{inf}$ (min) | Reference | $C_{max}$ ($\frac{\mu mol}{L}$) | $AUC_{24}$ ($\frac{\mu mol \cdot min}{L}$) | $AUC_\infty$ ($\frac{\mu mol \cdot min}{L}$) | $S_p$ |
|---|---|---|---|---|---|---|
| 30 | 30 | Model | 2.27 | 90.20 | 138.3 | - |
| | | [61]a | 0.76 | 29.55 | 39.54 | 0.323 |
| 20 | 60 | Model | 1.18 | 95.48 | 120.1 | - |
| | | [60] | 0.75 | 75.34 | 97.26 | 0.091 |
| 30 | 60 | Model | 1.52 | 124.6 | 170.3 | - |
| | | [62]a | 1.69 | 106.6 | 124.5 | 0.153 |
| 75 | 60 | Model | 2.71 | 217.9 | 395.3 | - |
| | | [58] | 3.55 | 293.1 | 318.6 | 0.154 |
| | | [59] | 3.20 | 154.7 | 211.7 | 0.145 |
| | | [54] | 4.61 | 318.0 | 431.8 | 0.257 |

The variances computed for the evaluation data can be found in Tbl. 3.6. The variances

generally indicate that the concentration of DOC within the blood plasma is predicted well by the model, particularly when considering the high degree of interpatient variability associated with DOC administration. While direct comparisons of the variance are not productive because of the differences between each data set, a lower variance indicates a more accurate prediction and a higher degree of confidence in the derived metrics. Of the evaluation data the least accurately predicted is Awada et al. [63] while the most accurate is Cox et al. [62]c.

Table 4.2: Metrics computed for the DOC evaluation data compared to those derived from the clinical concentration-time data.

| Dose ($mg/m^2$) | $T_{inf}$ (min) | Reference | $C_{max}$ ($\frac{\mu mol}{L}$) | $AUC_{24}$ ($\frac{\mu mol \cdot min}{L}$) | $AUC_\infty$ ($\frac{\mu mol \cdot min}{L}$) | $S_p$ |
|---|---|---|---|---|---|---|
| 35 | 30 | Model | 2.51 | 100.1 | 160.2 | - |
| | | [64]a | 3.20 | 140.3 | 184.9 | 1.006 |
| 30 | 60 | Model | 1.52 | 124.6 | 170.3 | - |
| | | [62]b | 2.43 | 123.9 | 148.3 | 0.278 |
| | | [62]c | 2.49 | 110.2 | 146.4 | 0.081 |
| | | [56]a | 1.23 | 87.47 | 112.9 | 0.303 |
| 36 | 60 | Model | 2.56 | 102.0 | 164.5 | - |
| | | [61]b | 0.78 | 45.62 | 64.43 | 0.872 |
| 60 | 60 | Model | 2.35 | 190.5 | 318.3 | - |
| | | [65]a | 1.79 | 150.5 | 167.1 | 0.514 |
| | | [66] | 1.86 | n/a | 233.2 | 0.526 |
| 75 | 60 | Model | 2.71 | 217.9 | 395.3 | - |
| | | [63] | 3.10 | 208.2 | 212.5 | 1.634 |
| | | [64]b | 2.27 | 114.6 | 191.6 | 0.684 |

Table 4.2: (continued) Metrics computed for DOC from the three-compartment one-molecule fractal MM model compared to those derived from the clinical concentration-time data.

| Dose ($mg/m^2$) | $T_{inf}$ (min) | Reference | $C_{max}$ ($\frac{\mu mol}{L}$) | $AUC_{24}$ ($\frac{\mu mol \cdot min}{L}$) | $AUC_{\infty}$ ($\frac{\mu mol \cdot min}{L}$) | $S_p$ |
|---|---|---|---|---|---|---|
| | | [65]b | 2.03 | 156.9 | 167.1 | 0.416 |
| | | [56]b | 2.37 | 168.1 | 253.3 | 0.639 |
| 100 | 60 | Model | 3.26 | 259.4 | 529.7 | - |
| | | [32]a | 3.19 | n/a | 178.0 | 0.671 |
| | | [32]b | 2.73 | n/a | 161.3 | 0.220 |
| | | [32]d | 3.97 | 363.7 | 370.1 | 0.534 |
| | | [32]e | 5.82 | 347.7 | 364.5 | 0.559 |
| 100 | 90 | Model | 2.54 | 293.5 | 562.3 | - |
| | | [32]c | 4.69 | n/a | 378.0 | 0.336 |

Why DOC is so difficult to model can be clearly seen by considering the concentration-time data from Blagden et al. [65]a. The large range in concentrations between patients can be seen in Fig. 4.2. If the concentrations from each patient are averaged, the variance is reduced by $7.1\times$ to $S_p = 0.079$. This shows that while DOC concentrations may be hard to accurately predict for a given patient, the average patient is well described by the model.

## 4.2 Metric Comparison

Of principle concern to clinicians are the PK metrics which inform treatment as described in §1.1.5. The most important of these are the peak plasma concentration, $C_{max}$, which correlates with toxicity, the Area Under the Curve, $AUC$, which correlates with drug exposure, and the terminal half-life, $t_{1/2,\gamma}$, which informs dosing schedules. Due to the specific PKs of the model being used we are also able to derive the distribution half-life $t_{1/2,\alpha}$

from this model. To derive these metrics, the model was solved to provide a continuous curve. That is, the concentrations were generated using a small step size relative to the time. These step sizes ranged from 0.25-10 min, starting small and increasing with time. The concentrations are calculated from the beginning of the infusion to three weeks after the infusion starts. The techniques described in §2.3 can then be applied to the results of these calculations.

**Peak Plasma Concentration**

The peak plasma concentration is the largest concentration achieved by DOC within the blood plasma. It is obtained from the model as the largest calculated concentration. $C_{max}$ from the clinical data is simply the largest concentration in the data set. For DOC, $C_{max}$ corresponds to the end of the infusion. The peak concentrations are presented in Tbl. 4.1 and Tbl. 4.2 for the calibration and evaluation DOC data respectively.

It should also be noted that the techniques used to generate the model, specifically the use of a weighted percentage variance, does not bias the fit towards high concentration data in the way the use of an absolute variance would. Despite this, we still see general agreement between the model and the clinical $C_{max}$ values. For all dosage regimes with >3 data sets the theoretical $C_{max}$ lies within the range of the evaluation $C_{max}$ values given in Tbl. 4.2.

**Area Under Curve**

There are two $AUC$ values which are relevant for DOC. The first is $AUC_{24}$ and the second is $AUC_{\infty}$. $AUC_{24}$ is computed from the time the infusion starts to 24 h after the infusion is stopped. The linear trapezoidal method (see Eqn. 2.11) is applied to the model curve to compute $AUC_{24}$ values for each dosage. To derive $AUC_{24}$ from the clinical data the linear trapezoidal method is applied directly to the concentration-time data. This is done for all data sets that provide a concentration at least 24 h post infusion.

$AUC_{\infty}$ is computed from the time the infusion starts and extrapolated to infinity for the

clinical data by assuming an exponential decay for the tail end of the elimination such that,

$$AUC_{\infty} = AUC_{last} + \frac{C_{last} t_{1/2,\gamma}}{\ln(2)} \tag{4.1}$$

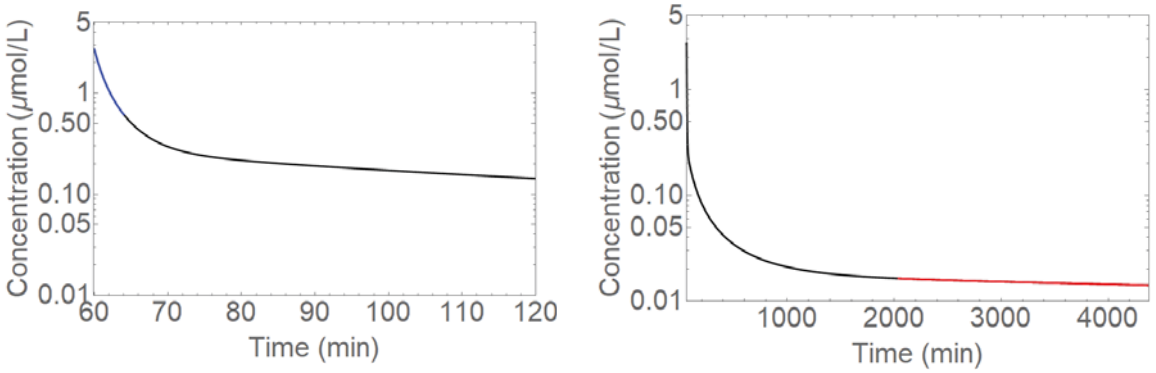
where $AUC_{last}$ is the $AUC$ from the beginning of the infusion to the final data point $C_{last}$ as computed using the linear trapezoidal method. $AUC_{\infty}$ for the model is computed by applying the linear trapezoidal method to the model curve from the time the infusion begins to three-weeks post infusion. At this point the concentrations are at least $10\times$ less than 0.0050 $\mu$mol/L which is the bottom range of the DOC concentration required to inhibit 50% of human cell growth in vitro [67]. The $AUC$ values are presented in Tbl. 4.1 and Tbl. 4.2.

When the evaluation data is considered we see that the theoretical $AUC_{24}$ values are generally larger than the clinical evaluation values with the exception of Baker et al. [64]a, Clarke et al. [32]d, and Clarke et al. [32]e while the theoretical $AUC_{\infty}$ is consistently larger, with only Baker et al. [64]a being an exception.

**Half-Life**

The half-lives of DOC can be recognized by plotting the concentration-time curve on a semi-log scale. When this is done for the modelled concentrations two distinct phases can be seen. The slopes of this plot are related to the distributive and terminal half-lives through Eqn. 2.13 and Eqn. 2.12 respectively. These phases can be seen in Fig. 4.1 which shows 72 h post infusion concentrations of DOC.

The alpha half-life is computed for the clinical data by performing a linear regression on the semi-log concentrations of two data points, the point corresponding to $C_{max}$ and the point immediately following. It is computed for the model by applying the same procedure to the model curve over the same time period used to compute the clinical value. This method gives a very rough estimate of the distributive half-life. The values derived from the calibration data are presented in Tbl. 4.3 and the values derived from the evaluation data




(a) Distribution phase.

(b) Terminal phase.

Figure 4.1: The distributive and terminal phases of the three-compartment one-molecule fractal MM model. — 75 mg/m$^2$ for 60 min. — Distribution phase, — Terminal phase.

are presented in Tbl. 4.4.

The terminal half-life is computed as above where again, the times used were adjusted to match with the available clinical data at the terminal phase of elimination. For the clinical data the terminal half-life is computed by taking the final 2 or 3 data points provided in the data set. Two points were used only in cases where the third point was too close to the phase transition. To compute the terminal half-life for the model a linear regression is performed on the model's semi-log concentrations over the same time period of the clinical half-life that was computed. The values derived from the calibration data are presented in Tbl. 4.3 and the values derived from the evaluation data are presented in Tbl. 4.4.

While the values presented in Tbl. 4.3 and Tbl. 4.4 may provide insight into how quickly DOC is removed from the blood plasma at different phases they have some major drawbacks. Primarily, the accuracy of these values relies on the frequency of sampling times used by the clinical studies. The sampling frequency in these studies is generally not high enough to clearly distinguish between DOC phases. This is particularly true of the distributive half-life as the first sample post infusion may occur up to an hour later. This is the case for Blagden et al. [65]a and Blagden et al. [65]b. The theoretical values in these cases are much larger than the other half-lives calculated at the same dose because of the time period used to derive these values. The long range behaviour should be more accurate because

many multiples of the half-life are not occurring within the time period but they are still constrained by the sampling times.

Table 4.3: Half-lives computed for the DOC calibration data compared to those derived from the clinical concentration-time data.

| Dose ($mg/m^2$) | $T_{inf}$ (min) | Ref. | Model $t_{1/2,\alpha}$ (min) | Clinical $t_{1/2,\alpha}$ (min) | Model $t_{1/2,\gamma}$ (min) | Clinical $t_{1/2,\gamma}$ (min) |
|---|---|---|---|---|---|---|
| 30 | 30 | [61]a | 2.06 | 2.78 | 1311 | 1296 |
| 20 | 60 | [60] | 14.2 | 15.6 | 3153 | 1942 |
| 30 | 60 | [62]a | 3.34 | 17.0 | 863 | 965 |
| 75 | 60 | [58] | 26.7 | 16.4 | 966 | 786 |
| | | [59] | 8.45 | 12.2 | 1151 | 1351 |
| | | [54] | 4.40 | 4.67 | 10789 | 2887 |

Table 4.4: Half-lives computed for the DOC evaluation compared to those derived from the clinical concentration-time data.

| Dose ($mg/m^2$) | $T_{inf}$ (min) | Ref. | Model $t_{1/2,\alpha}$ (min) | Clinical $t_{1/2,\alpha}$ (min) | Model $t_{1/2,\gamma}$ (min) | Clinical $t_{1/2,\gamma}$ (min) |
|---|---|---|---|---|---|---|
| 35 | 30 | [64]a | 9.79 | 4.84 | 4743 | 5195 |
| 30 | 60 | [62]b | 3.34 | 4.31 | 863 | 1328 |
| | | [62]c | 3.34 | 3.40 | 863 | 1998 |
| | | [56]a | 2.39 | 2.93 | 3924 | 2355 |
| 36 | 60 | [61]b | 5.27 | 5.96 | 1293 | 1371 |
| 60 | 60 | [65]a | 26.8 | 14.3 | 710 | 674 |
| | | [66] | 5.58 | 9.77 | 169 | 535 |
| 75 | 60 | [63] | 10.4 | 25.3 | 439 | 302 |

Table 4.4: (continued) Half-lives computed for the DOC evaluation compared to those derived from the clinical concentration-time data.

| Dose ($mg/m^2$) | $T_{inf}$ (min) | Ref. | Model $t_{1/2,\alpha}$ (min) | Clinical $t_{1/2,\alpha}$ (min) | Model $t_{1/2,\gamma}$ (min) | Clinical $t_{1/2,\gamma}$ (min) |
|---|---|---|---|---|---|---|
| | | [64]b | 3.46 | 15.9 | 6611 | 7010 |
| | | [65]b | 26.7 | 13.2 | 802 | 674 |
| | | [56]b | 1.92 | 3.71 | 3092 | 12566 |
| 100 | 60 | [32]a | 2.01 | 3.57 | 570 | 375 |
| | | [32]b | 2.25 | 5.72 | 561 | 299 |
| | | [32]d | 1.79 | 21.3 | 1318 | 419 |
| | | [32]e | 1.79 | 5.92 | 1396 | 741 |
| 100 | 90 | [32]c | 5.61 | 6.28 | 546 | 880 |

An advantage of PK modelling is that the half-lives can be precisely determined by plotting the model's semi-log concentrations against time. Linear sections of this plot represent distinct phases (see Fig. 4.1). By excluding data that does not fall within the range displaying linear behaviour the half-life is more accurately captured.

The distributive phase of DOC occurs between 4-10 min post infusion depending on the dosage. The terminal phase can be computed for one day or for three days. The one day results will be more similar to current estimates of the terminal half-life while the three day results will be more accurate to the terminal behaviour of DOC. These values are presented in Tbl. 4.5.

The DOC monograph [21] gives the half-lives for doses between 70-115 $mg/m^2$ as $t_{1/2,\alpha} = 4$ min and $t_{1/2,\gamma} = 666$ min. The distributive half-life computed from the model is roughly half that given in the monograph. The 24 h terminal half-life is expected to be similar to clinical estimates of $t_{1/2,\gamma}$ however it is typically larger than these values.

Extending the time period over which this PK metric is considered further increases the model estimate of $t_{1/2,\gamma}$. This suggests that DOC is more slowly eliminated than current clinical estimates.

Table 4.5: Three-compartment one-molecule fractal MM half-lives.

| Dose ($mg/m^2$) | $T_{inf}$ (min) | $t_{1/2,\alpha}$ (min) | 24 h $t_{1/2,\gamma}$ (min) | 72 h $t_{1/2,\gamma}$ (min) |
|---|---|---|---|---|
| 30 | 30 | 2.155 | 1722 | 5539 |
| 35 | 30 | 2.198 | 1834 | 6411 |
| 20 | 60 | 2.824 | 945 | 3439 |
| 30 | 60 | 2.618 | 1119 | 5218 |
| 36 | 60 | 2.205 | 1855 | 6581 |
| 60 | 60 | 2.140 | 1479 | 9954 |
| 75 | 60 | 1.917 | 1636 | 11820 |
| 100 | 60 | 2.014 | 1949 | 14900 |
| 100 | 90 | 2.350 | 1792 | 14350 |

The results of the clinical metrics show general agreement with the model results. It is clear that the metrics often show large variations between studies with the same doses. Despite that variation the metrics are typically reasonably close, particularly if we consider the degree of error associated with the metrics computed in the literature as given in Tbl. 4.6. The peak concentrations given in Tbl. 4.6 all agree well within error to their model counter parts, with the exception of Baker et al. [54]. All of the $AUC_{24}$ values in Tbl. 4.6 agree with the model metrics as well. Prediction of the $AUC_{\infty}$ metric in Tbl. 4.6 shows less agreement. Only Baker et al. [54] and Cox et al. [62]c agree within error. This is not unexpected, however, as the methods used to extrapolate the $AUC$ beyond the final concentration sample will vary between studies and extrapolation will inherently be less accurate. The terminal

half-lives given in Tbl. 4.6 also show less agreement with the model with only the metrics given by Cox et al. [62] agreeing within error. Again, with this metric, there is no standard method for determining $t_{1/2,\gamma}$ so a larger degree of variation is not unexpected.

Because the metrics are derived from a model fit to a significantly larger volume of data ($N_p = 860$) than any of the individual studies the PK metrics derived from the model will be accurate for the average patient receiving DOC treatment.

Table 4.6: DOC PK metrics as computed in the literature.

| Reference | $C_{max}$ ($\frac{\mu\text{mol}}{\text{L}}$) | $AUC_{24}$ ($\frac{\mu\text{mol}\cdot\text{min}}{L}$) | $AUC_{\infty}$ ($\frac{\mu\text{mol}\cdot\text{min}}{L}$) | $t_{1/2,\gamma}$ (min) |
|---|---|---|---|---|
| Awada et al. [63] | 3.20(90) | - | 228(62) | - |
| Baker et al. [64]b | 2.70(88) | 227(63) | - | 1050(440) |
| Lim et al. [58] | 2.8(14) | 280(200) | - | 735(210) |
| Baker et al. [54] | 4.8(18) | - | 427(190)* | - |
| Baker et al. [64]a | 2.29(90) | 98(31) | - | 936(720) |
| Brunsvig et al. [60] | 0.74(55) | 92(34) | - | - |
| Cox et al. [62]a | 1.9(14) | - | 100(52) | 714(490) |
| Cox et al. [62]b | 2.3(21) | - | 112(46) | 714(460) |
| Cox et al. [62]c | 2.6(28) | - | 146(100) | 936(550) |

* *AUC* computed to 8 days post infusion.

## 4.3 Individual PK

One of the main advantages to modelling pharmaceutical compounds is the ability to adjust the model to reflect interpatient variations. The underlying cause of these variations for DOC may be related to age, renal function, sex, ethnicity or other factors which can impact the level of α-acid glycoprotein and albumin in the blood plasma [68, 69, 70]. Only

unbound drug is pharmacologically active and able to move between compartments [71]. Therefore, the extent of DOC protein binding will have an effect on the efficacy of treatment and the PK metrics. Protein binding with α-acid glycoprotein is thought to be the main source of interpatient variability for DOC [72].

Most of the concentration data provided in the literature is either given as the average concentration or a single patient's data is selected as a 'representative' patient. This makes it difficult to understand the breadth of the potential patient responses to DOC treatment. Blagden et al. [65] does not average the concentration for each sampling, giving us a glimpse into the degree with which the concentration of DOC is dependent on the patient. However, the concentration-time data presented in Blagden et al. [65] does not differentiate patients between each sampling. To complete the individual PK analysis it was assumed that the largest concentration at each sample belonged to the same patient, the second largest at each sample to the same patient, and so on for all of the data. Another study which displayed the concentrations for individual patients, Clarke et al. [32], supports this assumption. We assigned 'patient 0' as the patient with the highest concentration, 'patient 1' as the second highest, etc. with 'patient 14' having the lowest concentration data.

The three-compartment one-molecule fractal MM model is a population PK model. It represents the patients collectively. To understand how well it captures the PKs of individual patients the variance can be computed separately for patients 0-14. The sum of these values is the population total variance $\sum S_{p,p} = 8.340$. The individualized data sets can then be fit to the model, allowing a single parameter from the population model to vary at a time. The variance can then be recalculated for each patient after the parameter has been allowed to vary. When this is done the parameter with the largest reduction in the total variance is determined to be the parameter with the highest degree of patient sensitivity. The total adjusted variance $\sum S_{p,i}$ for each parameter is given in Tbl. 4.7.

Table 4.7: Parameter sensitivity to individual patient PK.

| Parameter* | $\sum S_{p,i}$ |
|---|---|
| $\overline{k}_{1,1,2}$ | 1.3586 |
| $V_d$ | 2.3529 |
| $A_{1,3,1}$ | 4.1650 |
| $\overline{k}_{1,3,1}$ | 4.2281 |
| $A_{1,1,3}$ | 4.2582 |
| $\Gamma_{1,3,1}$ | 4.3145 |
| $\overline{k}_{1,1,3}$ | 4.7041 |
| $A_{1,3,0}$ | 5.0933 |
| $\Gamma_{1,3,0}$ | 5.3093 |
| $\overline{k}_{1,3,0}$ | 5.4209 |
| $\Gamma_{1,2,1}$ | 6.0010 |
| $\overline{k}_{1,2,1}$ | 6.0010 |
| $A_{1,2,1}$ | 6.0352 |
| $\Gamma_{1,1,2}$ | 6.3483 |
| $A_{1,1,2}$ | 6.5191 |
| $\Gamma_{1,1,3}$ | 8.1231 |

* The units for $\overline{k}_{p,c,c'}$ are $[(\frac{1}{\mathrm{s}})(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{1-A_{p,c,c'}}]$ , the units for $\Gamma_{p,c,c'}$ are $[(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{1-B_{p,c,c'}}]$, and the units for $V_{D,(DOC,1)}$ are $[\frac{\mathrm{L}}{\mathrm{m}^2}]$. Parameter values can be found in Tbl. 3.13. All other parameters are dimensionless.

The largest reduction in the variance is associated with $\overline{k}_{1,1,2}$ with an 83% reduction. The $V_d$ also shows a high degree of patient sensitivity with a 71% reduction. The remaining parameters all reduce the variance <50%. In Tbl. 4.8 we see that, compared to the model value of $\overline{k}_{1,1,2} = 0.636(58)\ (\frac{1}{\mathrm{s}})(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{-1.28}$, the rate constant is smaller for patients with higher blood plasma concentrations of DOC and becomes significantly larger for low concentration patients. While this parameter does the best job of accounting for the most

extraneous cases (patients 11-14) the PK of these patients is clearly still not very well captured by the model. The extreme cases are depicted in Fig. 4.2.

Table 4.8: Sensitivity of $\overline{k}_{1,1,2}$ to individual patients.

| Patient | $\overline{k}^{*}_{1,1,2}$ | $S_{p,i}$ | $S_{p,p}$ |
|---|---|---|---|
| 0 | 0.255 | 0.0057 | 0.2449 |
| 1 | 0.363 | 0.0046 | 0.0923 |
| 2 | 0.426 | 0.0052 | 0.0491 |
| 3 | 0.613 | 0.0014 | 0.0017 |
| 4 | 0.706 | 0.0055 | 0.0083 |
| 5 | 0.842 | 0.0079 | 0.0277 |
| 6 | 1.06 | 0.0114 | 0.0740 |
| 7 | 1.27 | 0.0168 | 0.1292 |
| 8 | 1.54 | 0.0198 | 0.1978 |
| 9 | 1.86 | 0.0153 | 0.2693 |
| 10 | 2.46 | 0.0502 | 0.4243 |
| 11 | 7.26 | 0.0230 | 1.0108 |
| 12 | 28.2 | 0.0474 | 1.6653 |
| 13 | 105 | 0.5127 | 2.0646 |
| 14 | 141 | 0.6129 | 1.6547 |

* The units for $\overline{k}_{1,1,2}$ are $[(\frac{1}{\mathrm{s}})(\frac{\mu\mathrm{mol}}{\mathrm{m}^2})^{-1.28}]$.

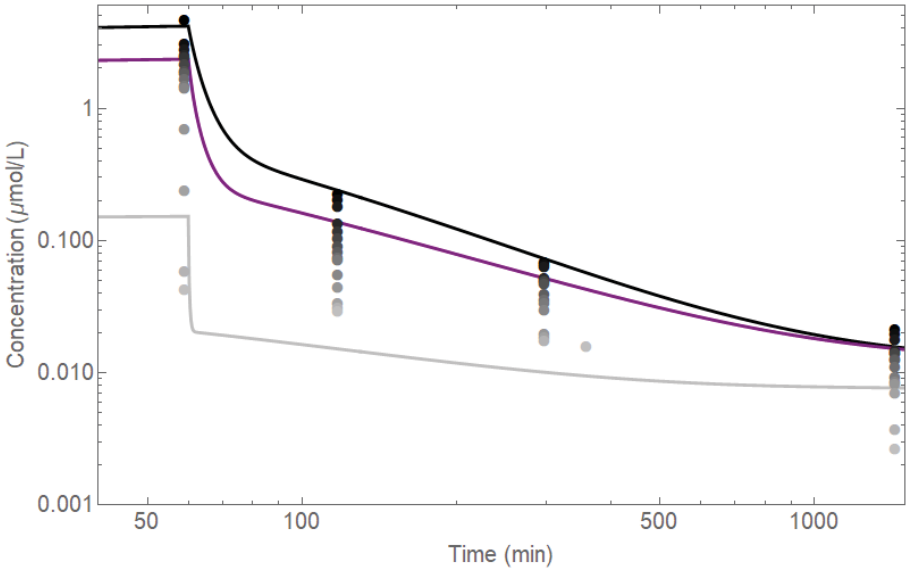



Figure 4.2: The effect of varying $\bar{k}_{1,1,2}$ for individual patients. The comparison shows Blagden et al. [65]b patients ● patient 0, ● patient 14 and the adjusted results for the extreme cases between — the patient 0 model and — the patient 14 model along with — the population model.

Both of $\bar{k}_{1,1,2}$ and $V_D$ may be useful in improving the accuracy of metric predictions. More information about the patients which would allow us to correlate the value of the parameter with a cofactor would be necessary to actually implement this result in a way that would improve patient outcomes. Unfortunately, this information is not readily available. However, these results indicate that there is potential to customize the model for individuals.

# Chapter 5

# Conclusion

*"An approximate answer to the right question is worth far more than a precise answer to the wrong one."*

– John Tukey

In this work, a novel model which introduces fractal and saturable processes is developed for DOC and it's delivery vehicle P80. Previous studies have used linear three-compartment models to describe the PK of DOC. The introduction of non-linear terms was shown to improve the accuracy of the models in all cases when compared to their previously established kinetic counterparts. The techniques applied provide insight to the PK behaviours that determine important PK metrics which, in turn, inform treatment plans.

One- two- and three-compartment models of DOC were developed along with a one-compartment model of P80 and a one-compartment two-molecule model of P80 and DOC together. These models are tested against calibration data digitized from published studies which include a variety of patients and dosing regimes. The individual studies within the calibration data set were used to develop a baseline for the parameters before fitting the model to the calibration data. The results of the modelling are verified against a set of evaluation data which was not used in fitting the model.

The model is built up slowly, with the results from the one-compartment models informing the two-compartment models, and in turn the three- compartment models. Increasing

the models complexity in turn increases the number of parameters. To ensure the data is not being overfit the Akaike Information Criterion, $AIC_c$, was utilized.

Of the one-compartment models tested for P80, the fractal model was selected with a substantial level of empirical evidence when compared to all other tested models as determined by the $AIC_c$. This indicates that the modelling of molecules with what is perceived to be simple kinetics may still see benefits to their description when more complex kinetics are introduced to their models.

Inclusion of P80 in the one-compartment model of DOC gave poor fit results indicating that P80 dissociates from DOC nearly instantaneously when entering the blood stream. The poor fits in comparison to models of DOC alone indicates that P80 does not affect the PKs of DOC. This motivates the exclusion of P80 from the DOC models developed in this work.

The one-compartment DOC models performed well for high dose DOC infusions but failed to accurately predict DOC concentrations at low doses. The best performing one-compartment model was the saturable model which indicates that non-standard kinetics should be used for DOC modelling. The two-compartment models tested further improved the accuracy of the DOC model. The results of this testing selected a fractal MM model as the best fit. These models informed the fits for the three-compartment models.

Ultimately, the model selected for DOC by the $AIC_c$ was the three-compartment fractal MM model. The main advantage this model displayed was its ability to predict DOC concentrations for both high and low doses. Previously, three-compartment models have been recommended for modelling DOC. However, these models would not have included variable exponents. We found that the inclusion of a variable exponent improved the DOC simulations enough to warrant their inclusion in DOC models.

From the three-compartment fractal MM model we are able to derive accurate, clinically relevant PK metrics. Principally, these are the peak plasma concentration $C_{max}$, the area under the concentration-time curve $AUC$, and the distributive $t_{1/2,\alpha}$ and terminal $t_{1/2,\gamma}$ half-lives of the pharmacologically active compound DOC. The PK metrics are developed for

both the calibration and the evaluation data. The evaluation data results show that DOC is well predicted by the selected model. Despite a high degree of interpatient variability the variances computed for the evaluation data sets are generally low.

To better understand the effect of individual patients PKs on the three-compartment fractal MM model the sensitivity of each model parameter was evaluated for the patient data provided in Blagden et al [65]b. We found that the rate constant controlling the flow from compartment one into compartment two had the largest impact on improving the predictive power of the model.

We note that there are a few limitations to the results presented in this work. Firstly, relatively little data is available in the literature for P80 and therefore conclusions drawn from the modelling of P80 may not apply to a typical patient. Secondly, limited information about the patients, which the concentration-time data comes from, makes it difficult to draw conclusions about individual PKs. Lastly, the three-compartment one-molecule saturable model was not properly developed. It is possible that this model could improve the accuracy of the DOC model even further.

Additional investigation into the PKs of DOC could be pursued in the future. Of particular interest would be further parameterization of the three-compartment DOC model. It is also possible that a four-compartment model could improve the description of DOC. Another point of interest is the affect of multi-drug interactions on DOC. DOC is commonly given concurrently with other chemotherapeutic drugs which may influence DOC behaviour. Lastly, study of the specific cofactors which most influence the population PK of DOC would be useful for informing the individualization of patient treatment.

In summary, the three-compartment fractal MM model is able to characterize the PK of DOC with a reasonable degree of accuracy. Multi-compartment models which introduced variable exponents to the modelling of DOC improved the accuracy of predicting the concentration of DOC within the blood plasma.

# References


[1] M. Boarder, D. Newby, and P. Navti. *Pharmacology for Pharmacy and the Health Sciences: A patient-centred approach*. Oxford University Press, 2010.

[2] P. L. Bonate. Pharmacokinetics. *WIREs Computational Statistics*, 3(4):332–342, 2011.

[3] S. E. Rosenbaum. *Basic Pharmacokinetics and Pharmacodynamics: An Integrated Textbook and Computer Simulations*. John Wiley and Sons, 1st edition, 2011.

[4] S. A. Saganuwan. Application of modified michaelis–menten equations for determination of enzyme inducing and inhibiting drugs. *BMC Pharmacology and Toxicology*, 22(1):1–15, 2021.

[5] K. L. Paalzow. Torsten Teorell, the father of pharmacokinetics. *Upsala journal of medical sciences*, 100(1):41–46, 1995.

[6] T. Teorell. Kinetics of distribution of substances administered to the body, i. the extravascular modes of administration. *Archives internationales de pharmacodynamie et de therapie*, 57:205–225, 1937.

[7] T. Teorell. Kinetics of distribution of substances administered to the body, ii. the intravascular modes of administration. *Archives internationales de pharmacodynamie et de therapie*, 57:226–240, 1937.

[8] S. Braakman, P. Pathmanathan, and H. Moore. Evaluation framework for systems models. *CPT: pharmacometrics & systems pharmacology*, 11(3):264–289, 2022.

[9] P. L. Bonate. *Pharmacokinetic-Pharmacodynamic Modeling and Simulation*. Springer, 2011.

[10] K. A. Johnson and R. S. Goody. The original michaelis constant: translation of the 1913 michaelis–menten paper. *Biochemistry*, 50(39):8264–8269, 2011.

[11] R. Kopelman. Fractal reaction kinetics. *Science*, 241(4873):1620–1626, 1988.

[12] L. M. Pereira. Fractal pharmacokinetics. *Computational and Mathematical methods in Medicine*, 11(2):161–184, 2010.

[13] R. Marsh, J. Fuite, and J. A. Tuszynski. Fractal space and time—sources of nonlinearity in drug elimination. In *2003 European Control Conference (ECC)*, pages 2842–2850. IEEE, 2003.

[14] K. J. E. Vos, A. G. Martin, M. G. Trimboli, L. Forestell, K. Barakat, and J. A.Tuszynski. A multi-compartment pharmacokinetic model of the interaction between paclitaxel and doxorubicin. *EPJ Nonlinear Biomedical Physics*, 2:1–40, 2014.

[15] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray. Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 71(3):209–249, 2021.

[16] D. R. Brenner, H. K. Weir, A. A. Demers, L. F. Ellison, C. Louzado, A. Shaw, D. Turner, R. R. Woods, and L. M. Smith. Projected estimates of cancer in Canada in 2020. *Canadian Medical Association Journal*, 192(9):E199–E205, 2020.

[17] C. de Oliveira, S. Weir, J. Rangrej, M. D. Krahn, N. Mittmann, J. S. Hoch, K. K. W. Chan S., and Peacock. The economic burden of cancer care in Canada: a population-based cost study. *Canadian Medical Association Journal Open*, 6(1):E1, 2018.

[18] R. Garaszczuk, J. H. E. Yong, Z. Sun, and C. de Oliveira. The economic burden of cancer in Canada from a societal perspective. *Current Oncology*, 29(4):2735–2748, 2022.

[19] J. A. DiMasi, R. W. Hansen, and H. G. Grabowski. The price of innovation: new estimates of drug development costs. *Journal of health economics*, 22(2):151–185, 2003.

[20] J. A. DiMasi, H. G. Grabowski, and R. W. Hansen. Innovation in the pharmaceutical industry: new estimates of R&D costs. *Journal of health economics*, 47:20–33, 2016.

[21] Pfizer Canada Inc. *Docetaxel Injection USP*, June 2018.

[22] E. K. Rowinsky. The development and clinical utility of the taxane class of antimicrotubule chemotherapy agents. *Annual review of medicine*, 48(1):353–374, 1997.

[23] National Institutes of Health (US). Understanding cancer. https://www.ncbi.nlm.nih.gov/books/NBK20362/, 2007.

[24] B. W. Stewart, C. P. Wild, et al. *World Cancer Report 2014*. International Agency for Research on Cancer, 2014.

[25] D. Hanahan. Hallmarks of cancer: new dimensions. *Cancer discovery*, 12(1):31–46, 2022.

[26] B. Izar, J. Rotow, J. Gainor, J. Clark, and B. Chabner. Pharmacokinetics, clinical indications, and resistance mechanisms in molecular targeted therapies in cancer. *Pharmacological reviews*, 65(4):1351–1395, 2013.

[27] Y. H. Bae and K. Park. Targeted drug delivery to tumors: myths, reality and possibility. *Journal of controlled release*, 153(3):198, 2011.

[28] A. Shemi, E. Z. Khvalevsky, R. M. Gabai, A. Domb, and Y. Barenholz. Multistep, effective drug distribution within solid tumors. *Oncotarget*, 6(37):39564 – 39577, 2015.

[29] V. P. Chauhan, J. D. Martin, H. Liu, D. A. Lacorre, S. R. Jain, S. V. Kozin, T. Stylianopoulos, A. S. Mousa, X. Han, P. Adstamongkonkul, et al. Angiotensin inhibition enhances drug delivery and potentiates chemotherapy by decompressing tumour blood vessels. *Nature communications*, 4(1):1–11, 2013.

[30] L. K. Webster, M. E. Linsenmeyer, D. Rischin, M. E. Urch, D. M. Woodcock, and M. J. Millward. Plasma concentrations of polysorbate 80 measured in patients following administration of docetaxel or etoposide. *Cancer chemotherapy and pharmacology*, 39(6):557–560, 1997.

[31] E. K. Rowinsky, N. Onetto, R. M. Canetta, and S. G. Arbuck. Taxol: the first of the taxanes, an important new class of antitumor agents. *Seminars in oncology*, 4(4):253–265, 1992.

[32] S. J. Clarke and L. P. Rivory. Clinical pharmacokinetics of docetaxel. *Clinical pharmacokinetics*, 36(2):99–114, 1999.

[33] J.S. De Bono, S. Oudard, M. Ozguroglu, S. Hansen, J. Machiels, I. Kocak, G. Gravis, I. Bodrogi, M. J. Mackenzie, L. Shen, et al. Prednisone plus cabazitaxel or mitoxantrone for metastatic castration-resistant prostate cancer progressing after docetaxel treatment: a randomised open-label trial. *The Lancet*, 376(9747):1147–1154, 2010.

[34] J. P. Flores and M. W. Saif. Novel oral taxane therapies: recent phase i results. *Clinical investigation*, 3(4):333, 2013.

[35] L. S. Schwartzberg and R. M. Navari. Safety of polysorbate 80 in the oncology setting. *Advances in therapy*, 35(6):754–767, 2018.

[36] P.W. Gunning, U. Ghoshdastider, S. Whitaker, D. Popp, and R. C. Robinson. The evolution of compositionally and functionally distinct actin filaments. *Journal of cell science*, 128(11):2009–2019, 2015.

[37] R. Phillips, J. Kondev, J. Theriot, and H. Garcia. *Physical biology of the cell*. Garland Science, 2012.

[38] B. T. McGrogan, B. Gilmartin, D. N. Carney, and A. McCann. Taxanes, microtubules and chemoresistant breast cancer. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, 1785(2):96–132, 2008.

[39] M. Jordan and L. WIlson. Microtubules as a target for anticancer drugs. *Nature Reviews Cancer*, 19(6):646–662, 2004.

[40] S. Honore, E. Pasquier, and D. Braguer. Understanding microtubule dynamics for improved cancer therapy. *Cellular and Molecular Life Sciences CMLS*, 62(24):3039–3056, 2005.

[41] B. W. Derry, L. Wilson, and M. A. Jordan. Substoichiometric binding of taxol suppresses microtubule dynamics. *Biochemistry*, 34(7):2203–2211, 1995.

[42] F. Marre, G. Sanderink, G. de Sousa, C. Gaillard, M. Martinet, and R. Rahmani. Hepatic biotransformation of docetaxel (Taxotere®) in vitro: Involvement of the CYP3A subfamily in humans. *Cancer Research*, 56(6):1296–1302, 1996.

[43] C. Oshiro, S. Marsh, H. McLeod, M. Carrillo, T. Klein, and R. Altman. Taxane pathway. *Pharmacogenetics and genomics*, 19(12):979, 2009.

[44] J. Hirth, P. B. Watkins, M. Strawderman, A. Schott, R. Bruno, and L. H. Baker. The effect of an individual's cytochrome CYP3A4 activity on docetaxel clearance. *Clinical cancer research*, 6(4):1255–1258, 2000.

[45] The Human Metabolome Database. Docetaxel (hmdb0015378). https://hmdb.ca/metabolites/HMDB0015378/ # %20references.

[46] Gene Cards (The Human Gene Database). Slcob13 gene. https://www.genecards.org/cgi-bin/carddisp.pl?gene=SLCO1B3.

[47] S. A. Kaestner and G. J. Sewell. Chemotherapy dosing part i: scientific basis for current practice and use of body surface area. *Clinical oncology*, 19(1):23–37, 2007.

[48] H. Rosing, V. Lustig, L. J. C. Van Warmerdam, M. T. Huizing, W. W. Bokkel Huinink, J. H. M. Schellens, S. Rodenhuis, A. Bult, and J. H. Beijnen. Pharmacokinetics and metabolism of docetaxel administered as a 1-h intravenous infusion. *Cancer chemotherapy and pharmacology*, 45:213–218, 2000.

[49] W. H. Press, A. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipies in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1999.

[50] H. Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[51] H. Akaike. Information theory and an extension of the maximum likelihood principle. *Proc. 2nd Int. Symp. Information Theory, Supp.to Problems of Control and Information Theory*, pages 123–129, 1972.

[52] D. Anderson and K. Burnham. *Model selection and multi-model inference*. NY: Springer-Verlag, 2nd edition, 2004.

[53] Ankit Rohatgi. Webplotdigitizer: Version 4.6, 2022.

[54] S. D. Baker, J. Li, A. J. ten Tije, W. D. Figg, W. Graveland, J. Verweij, and A. Sparreboom. Relationship of systemic exposure to unbound docetaxel and neutropenia. *Clinical pharmacology & therapeutics*, 77(1):43–53, 2005.

[55] Wolfram Research, Inc. Mathematica, Version 12.0. Champaign, IL, 2019.

[56] S. D. Baker, M. Zhao, P. He, M. A. Carducci, J. Verweij, and A. Sparreboom. Simultaneous analysis of docetaxel and the formulation vehicle polysorbate 80 in human plasma by liquid chromatography/tandem mass spectrometry. *Analytical biochemistry*, 324(2):276–284, 2003.

[57] A. Dunne. An iterative curve stripping technique for pharmacokinetic parameter estimation. *Journal of pharmacy and pharmacology*, 38(2):97–101, 1986.

[58] Y. W. Lim, B. C. Goh, L. Z. Wang, S. H. Tan, B. Y. S. Chuah, S. E. Lim, P. Iau, S. A. Buhari, C. W. Chan, N. B. Sukri, et al. Pharmacokinetics and pharmacodynamics of docetaxel with or without ketoconazole modulation in chemonaive breast cancer patients. *Annals of oncology*, 21(11):2175–2182, 2010.

[59] F. Robert, A. Sandler, J. H. Schiller, G. Liu, K. Harper, L. Verkh, X. Huang, J. Ilagan, L. Tye, R. Chao, et al. Sunitinib in combination with docetaxel in patients with advanced solid tumors: a phase i dose-escalation study. *Cancer chemotherapy and pharmacology*, 66(4):669–680, 2010.

[60] P. F. Brunsvig, A. Andersen, S. Aamdal, V. Kristensen, and H. Olsen. Pharmacokinetic analysis of two different docetaxel dose levels in patients with non-small cell lung cancer treated with docetaxel as monotherapy or with concurrent radiotherapy. *BMC cancer*, 7(1):197, 2007.

[61] S. E. Witta, D. L. Gustafson, S. A. Pierson, A. Menter, S. N. Holden, M. Basche, M. Persky, C. L. O'Bryant, C. Zeng, A. Baron, et al. A phase i and pharmacokinetic study of exisulind and docetaxel in patients with advanced solid tumors. *Clinical cancer research*, 10(21):7229–7237, 2004.

[62] M. C. Cox, J. Low, J. Lee, J. Walshe, N. Denduluri, A. Berman, M. G. Permenter, W. P. Petros, D. K. Price, W. D. Figg, et al. Influence of garlic (allium sativum) on the pharmacokinetics of docetaxel. *Clinical cancer research*, 12(15):4636–4640, 2006.

[63] A. Awada, S. Zhang, T. Gil, D. de Valeriola, Y. Lalami, P. De Porre, and M. J. Piccart-Gebhart. A phase i clinical and pharmacokinetic study of tipifarnib in combination with docetaxel in patients with advanced solid malignancies. *Current medical research and opinion*, 23(5):991–1003, 2007.

[64] S. D. Baker, M. Zhao, C. K. Lee, J. Verweij, Y. Zabelina, J. R. Brahmer, A. C. Wolff, A. Sparreboom, and M. A. Carducci. Comparative pharmacokinetics of weekly and every-three-weeks docetaxel. *Clinical Cancer Research*, 10(6):1976–1983, 2004.

[65] S. P. Blagden, L. R. Molife, A. Seebaran, M. Payne, A. H. M. Reid, A. S. Protheroe, L. S. Vasist, D. D. Williams, C. Bowen, S. J. Kathman, et al. A phase i trial of ispinesib, a kinesin spindle protein inhibitor, with docetaxel in patients with advanced solid tumours. *British journal of cancer*, 98(5):894–899, 2008.

[66] B. F. El-Rayes, S. Gadgeel, R. Parchment, P. Lorusso, and P. A. Philip. A phase i study of flavopiridol and docetaxel. *Investigational new drugs*, 24(4):305, 2006.

[67] F. Lavelle, M. C. Bissery, C. Combeau, J. F. Riou, P. Vrignaud, and S. Andre. Preclinical evaluation of docetaxel (taxotere). 22(2 Suppl 4):3–16, 1995.

[68] J. Alcorn and P. J. McNamara. Pharmacokinetics in the newborn. *Advanced drug delivery reviews*, 55(5):667–686, 2003.

[69] M. Gandhi, F. Aweeka, R. M. Greenblatt, and T. F. Blaschke. Sex differences in pharmacokinetics and pharmacodynamics. *Annu. Rev. Pharmacol. Toxicol.*, 44:499–523, 2004.

[70] S. A. Smith and N. J. Waters. Pharmacokinetic and pharmacodynamic considerations for drugs binding to alpha-1-acid glycoprotein. *Pharmaceutical Research*, 36(2):30, 2019.

[71] S. Grogan and C. V. Preuss. Pharmacokinetics. *StatPearls [Internet]*, 2023.

[72] H. Kenmotsu, C. K. Imamuraand A. Ono, S. Omori, K. Nakashima, K. Wakuda, T. Taira, T. Naito, H. Murakami, T. Takahashi, et al. The effects of advanced age and serum α1-acid glycoprotein on docetaxel unbound exposure and dose-limiting toxicity in cancer patients. *British journal of clinical pharmacology*, 83(11):2416–2425, 2017.

# Appendix A

# Theoretical and Clinical Concentrations of DOC

Samples of the modelling results were shown in chapter §3. The remaining comparisons can be found below.

## A.1 One-Compartment One-Molecule Models

### A.1.1 Calibration Data

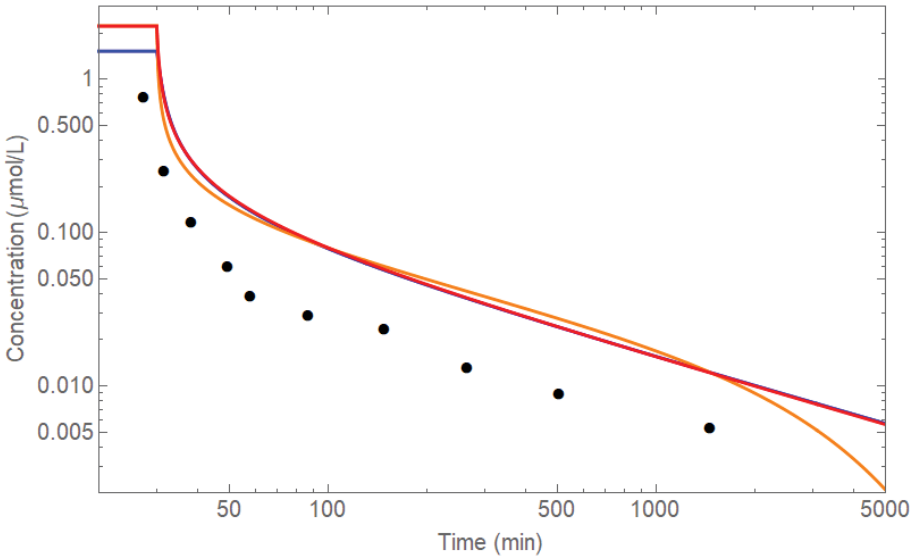



Figure A.1: Comparison between the one-compartment one-molecule models of DOC and a sample of calibration data with a dosage of 30 mg/m$^2$ over 30 min. The calibration data used is the data given in Tbl. 3.5. The one-compartment one-molecule models of DOC are given in Tbl. 3.7. ● Witta et al. [61]a, — Fractal, — Fractal+First, and — Saturable.

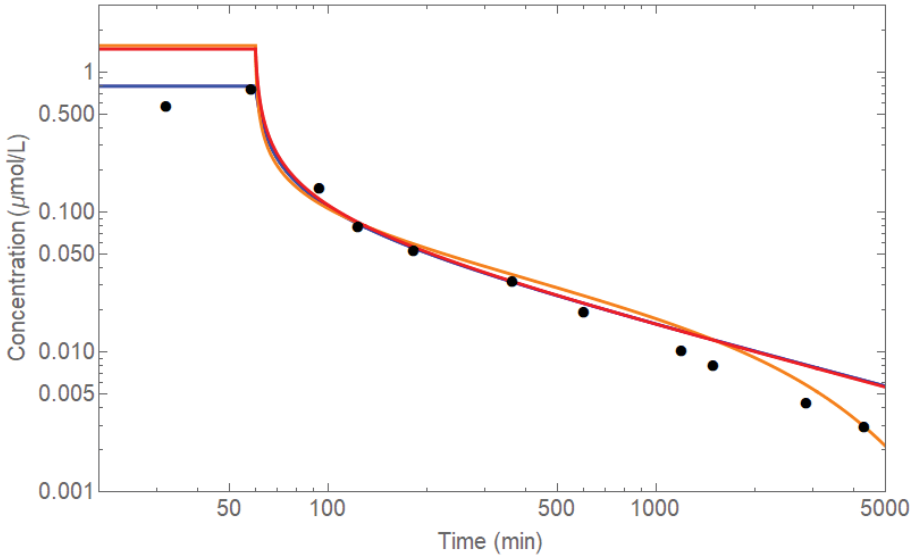



Figure A.2: Comparison between the one-compartment one-molecule models of DOC and a sample of calibration data with a dosage of 20 mg/m$^2$ over 60 min. The calibration data used is the data given in Tbl. 3.5. The one-compartment one-molecule models of DOC are given in Tbl. 3.7. ● Brunsvig et al. [60], — Fractal, — Fractal+First, and — Saturable.

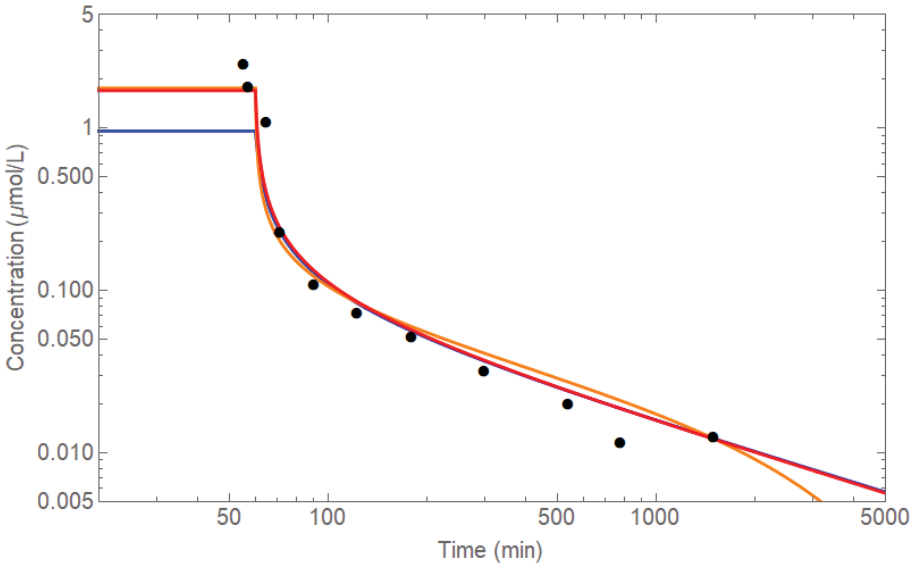



Figure A.3: Comparison between the one-compartment one-molecule models of DOC and a sample of calibration data with a dosage of 30 mg/m$^2$ over 60 min. The calibration data used is the data given in Tbl. 3.5. The one-compartment one-molecule models of DOC are given in Tbl. 3.7. ● Cox et al. [62]a, — Fractal, — Fractal+First, and — Saturable.

### A.1.2 Evaluation Data

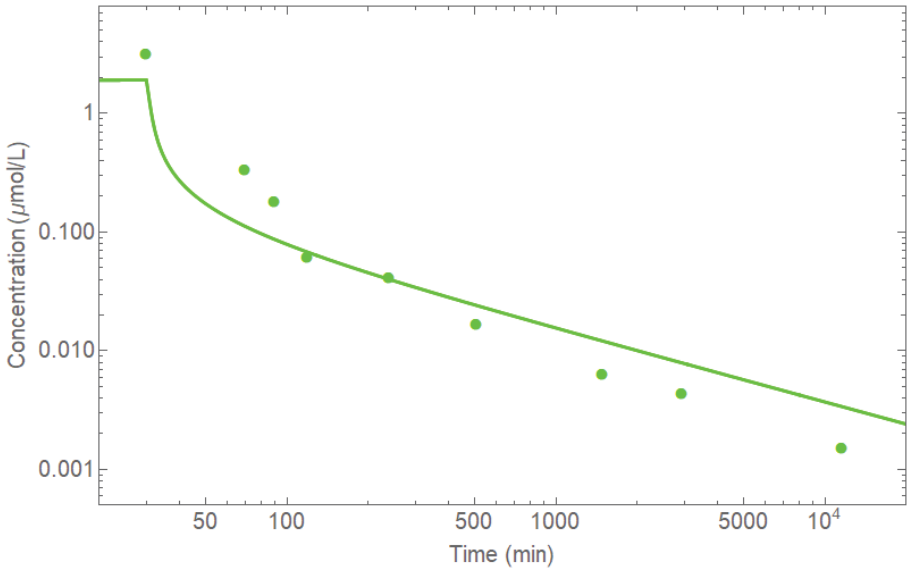



Figure A.4: Comparison between the one-compartment one-molecule saturable model of DOC and the evaluation data with a dosage of 35 mg/m$^2$ over 30 min. The calibration data used is the data given in Tbl. 3.5. The model shown is the one-compartment one-molecule saturablemodel of DOC given in Tbl. 3.7.
● Baker, et al. [64]a.

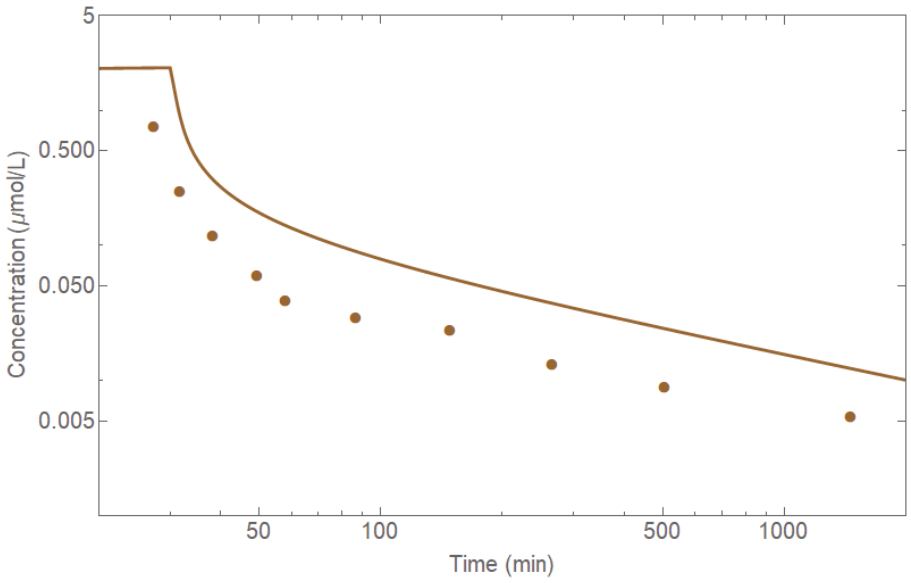



Figure A.5: Comparison between the one-compartment one-molecule saturable model of DOC and the evaluation data with a dosage of 36 mg/m$^2$ over 30 min. The calibration data used is the data given in Tbl. 3.5. The model shown is the one-compartment one-molecule saturablemodel of DOC given in Tbl. 3.7.
● Witta, et al. [61]b.

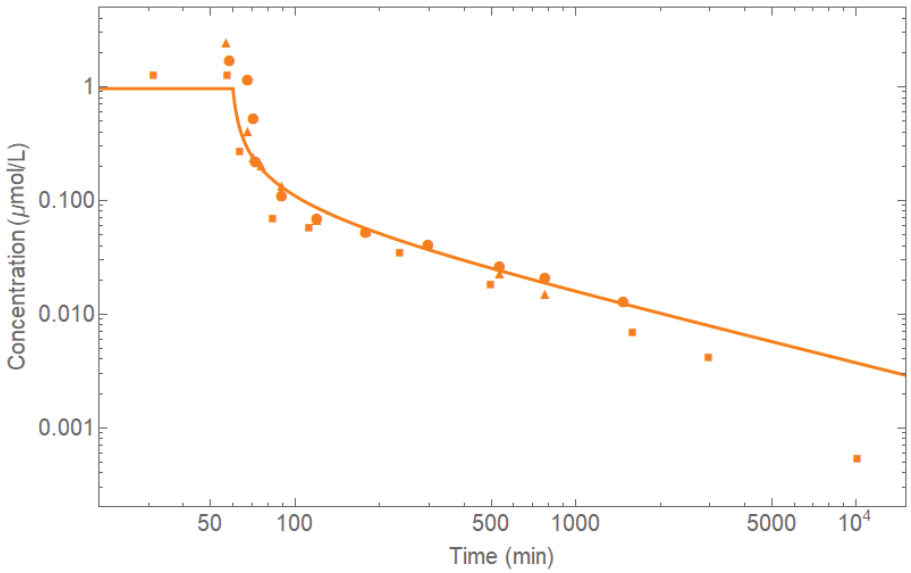



Figure A.6: Comparison between the one-compartment one-molecule saturable model of DOC and the evaluation data with a dosage of 30 mg/m$^2$ over 60 min. The calibration data used is the data given in Tbl. 3.5. The model shown is the one-compartment one-molecule saturablemodel of DOC given in Tbl. 3.7.
● Cox, et al. [62]b, ▲ Cox, et al. [62]c, ■ Baker et al. [56]a.

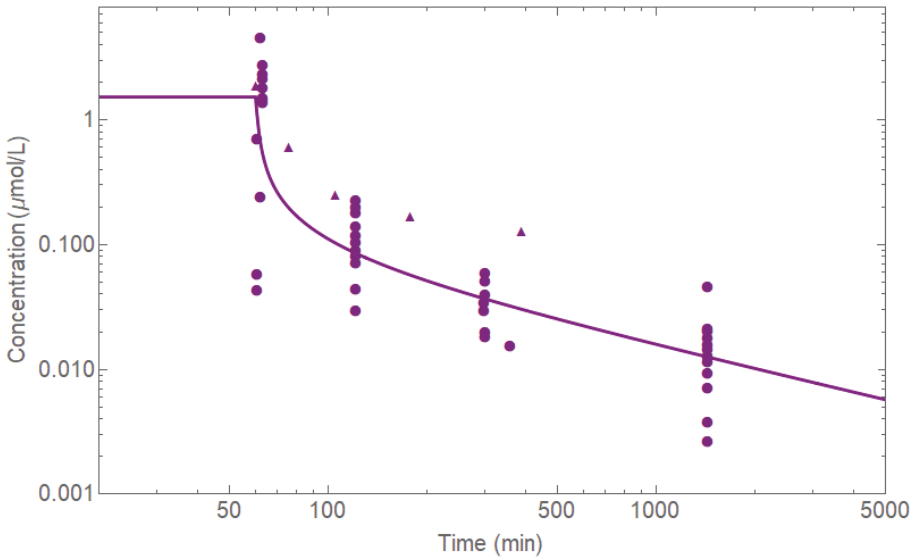



Figure A.7: Comparison between the one-compartment one-molecule saturable model of DOC and the evaluation data with a dosage of 60 mg/m$^2$ over 60 min. The calibration data used is the data given in Tbl. 3.5. The model shown is the one-compartment one-molecule saturablemodel of DOC given in Tbl. 3.11.
● Blagden, et al. [64]a, ▲ El-Rayes, et al. [66].

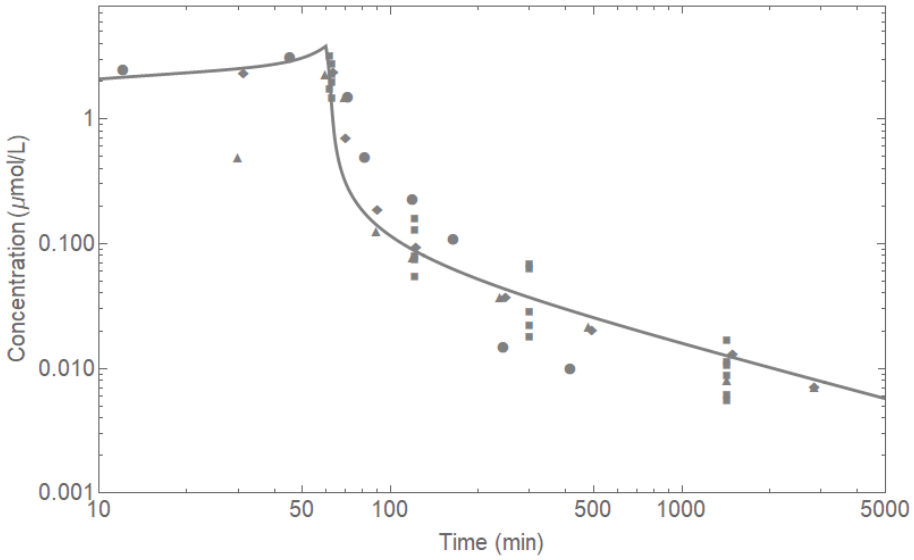



Figure A.8: Comparison between the one-compartment one-molecule saturable model of DOC and the evaluation data with a dosage of 75 mg/m$^2$ over 60 min. The calibration data used is the data given in Tbl. 3.5. The model shown is the one-compartment one-molecule saturablemodel of DOC given in Tbl. 3.7. ● Awada et al. [63], ▲ Baker et al. [64]b, ■ Blagden et al. [65]b, ◆ Baker et al. [56]b.

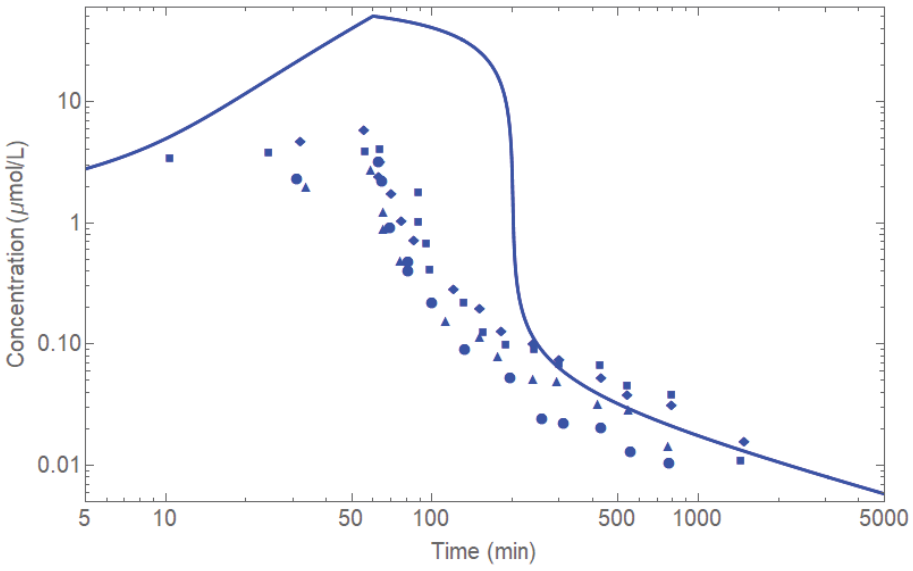



Figure A.9: Comparison between the one-compartment one-molecule saturable model of DOC and the evaluation data with a dosage of 100 mg/m$^2$ over 60 min. The calibration data used is the data given in Tbl. 3.5. The model shown is the the one-compartment one-molecule saturablemodel of DOC given in Tbl. 3.7.
● Clarke, et al. [32]a, ▲ Clarke, et al. [32]b, ■ Clarke, et al. [32]d, ◆ Clarke, et al. [32]e.

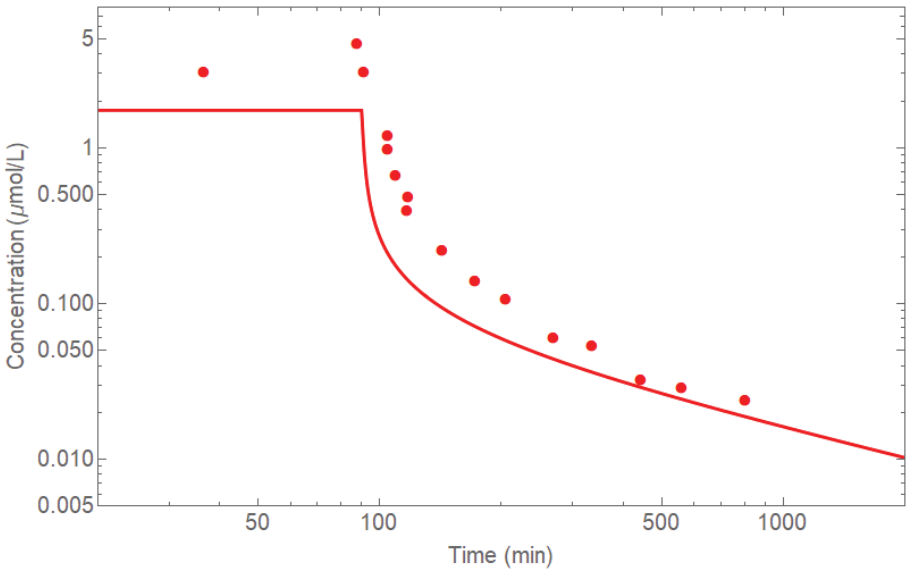



Figure A.10: Comparison between the one-compartment one-molecule saturable model of DOC and the evaluation data with a dosage of 100 mg/m$^2$ over 90 min. The calibration data used is the data given in Tbl. 3.5. The model shown is the the one-compartment one-molecule saturablemodel of DOC given in Tbl. 3.7.
● Clarke, et al. [32]c.

## A.2 One-Compartment Two-Molecule Models

### A.2.1 Calibration Data

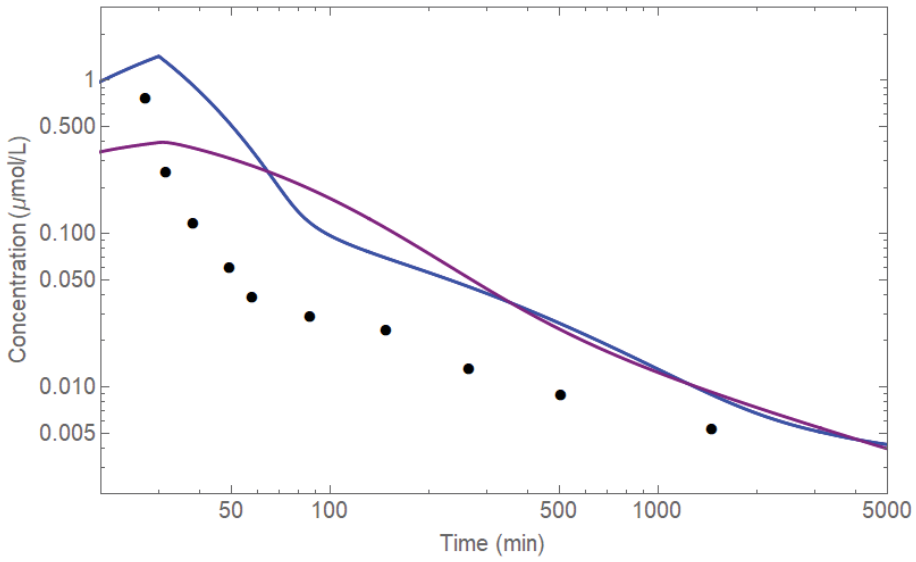



Figure A.11: Comparison between the one-compartment two-molecule models of DOC and a sample of calibration data with a dosage of 30 mg/m$^2$ over 30 min. The calibration data used is the data given in Tbl. 3.5. The one-compartment two-molecule models of DOC are given in Tbl. 3.9. ● Witta et al. [61]a, — Fractal, — Fractal+First, and — Saturable.

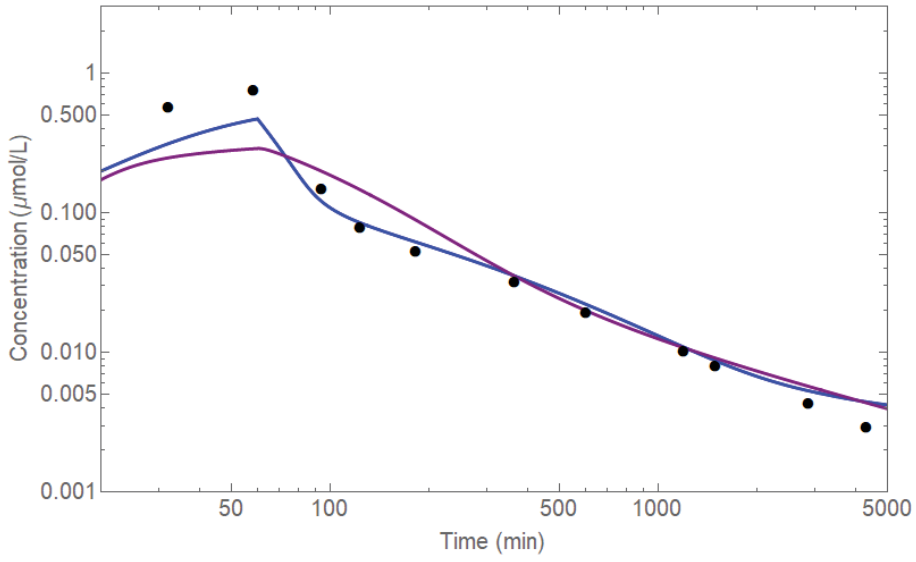



Figure A.12: Comparison between the one-compartment two-molecule models of DOC and a sample of calibration data with a dosage of 20 mg/m$^2$ over 60 min. The calibration data used is the data given in Tbl. 3.5. The one-compartment two-molecule models of DOC are given in Tbl. 3.9. ● Brunsvig et al. [60], — Fractal, — Fractal+First, and — Saturable.

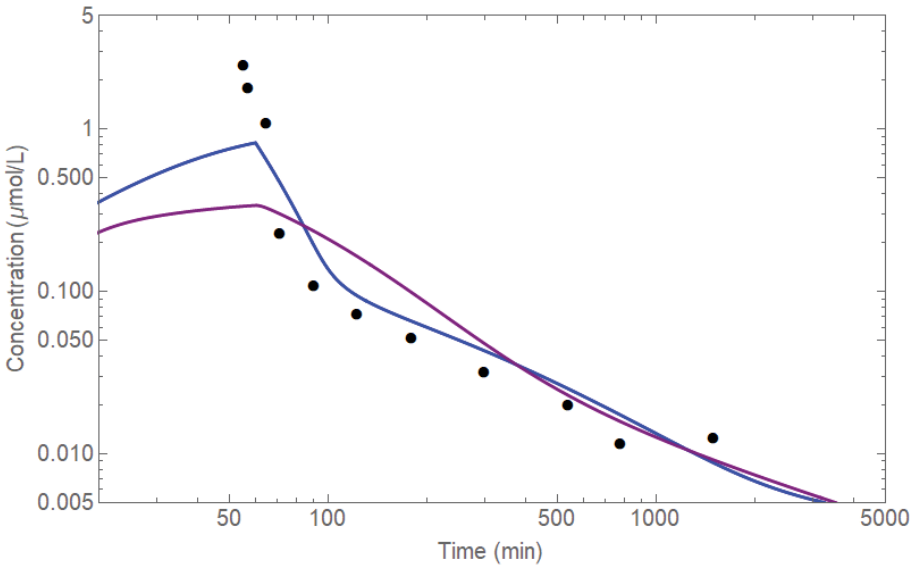



Figure A.13: Comparison between the one-compartment two-molecule models of DOC and a sample of calibration data with a dosage of 30 mg/m$^2$ over 60 min. The calibration data used is the data given in Tbl. 3.5. The one-compartment two-molecule models of DOC are given in Tbl. 3.9. ● Cox et al. [62]a, — Fractal, — Fractal+First, and — Saturable.

### A.2.2 Evaluation Data

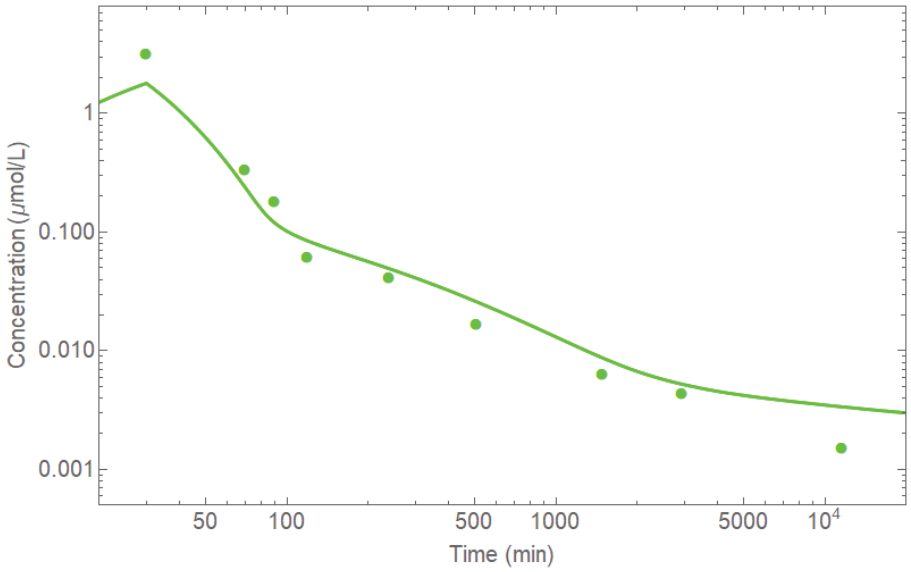



Figure A.14: Comparison between the one-compartment two-molecule saturable model of DOC and the evaluation data with a dosage of 35 mg/m$^2$ over 30 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the one-compartment two-molecule saturable model of DOC given in Tbl. 3.9.
● Baker, et al. [64]a.

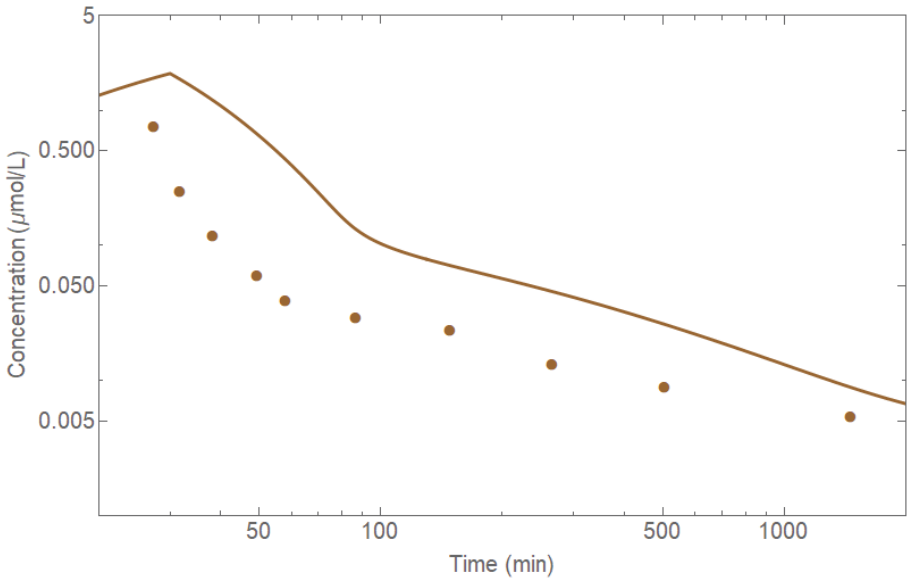



Figure A.15: Comparison between the one-compartment two-molecule saturable model of DOC and the evaluation data with a dosage of 36 mg/m$^2$ over 30 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the one-compartment two-molecule saturable model of DOC given in Tbl. 3.7.
● Witta, et al. [61]b.

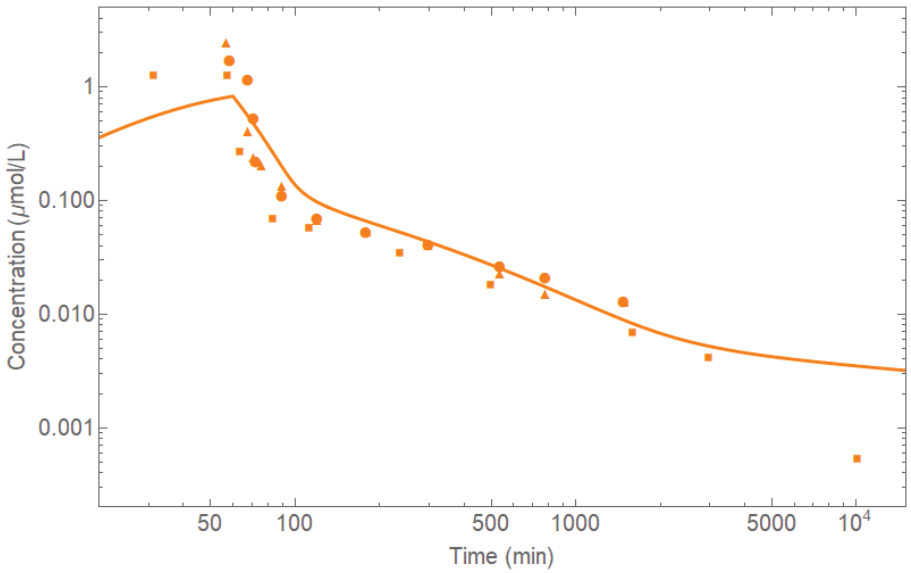



Figure A.16: Comparison between the one-compartment two-molecule saturable model of DOC and the evaluation data with a dosage of 30 mg/m$^2$ over 60 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the one-compartment two-molecule saturable model of DOC given in Tbl. 3.9.
● Cox, et al. [62]b, ▲ Cox, et al. [62]c, ■ Baker et al. [56]a.

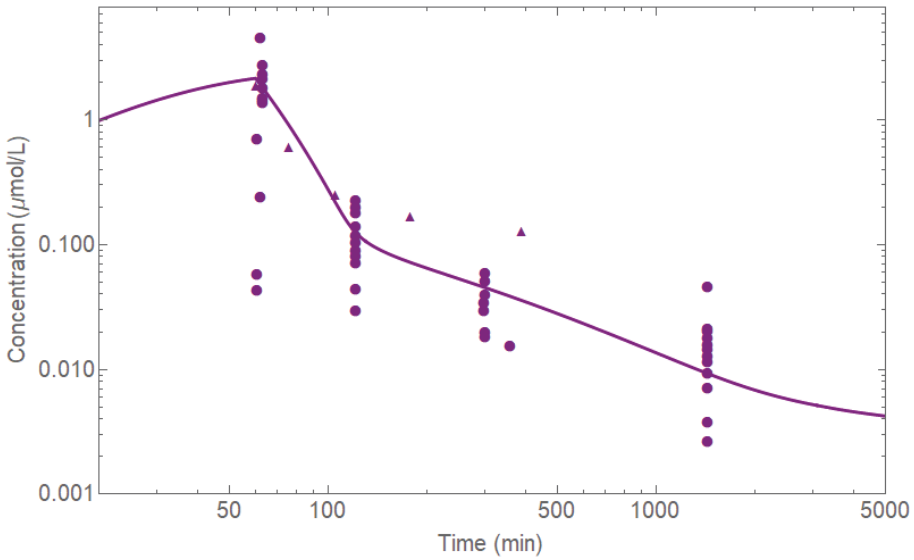



Figure A.17: Comparison between the one-compartment two-molecule saturable model of DOC and the evaluation data with a dosage of 60 mg/m$^2$ over 60 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the one-compartment two-molecule saturable model of DOC given in Tbl. 3.11.
● Blagden, et al. [64]a, ▲ El-Rayes, et al. [66].

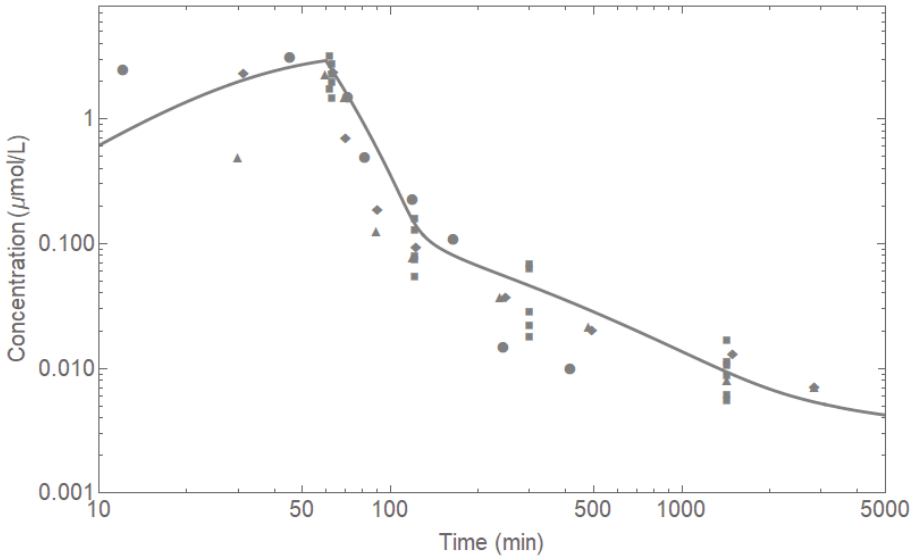



Figure A.18: Comparison between the one-compartment two-molecule saturable model of DOC and the evaluation data with a dosage of 75 mg/m$^2$ over 60 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the one-compartment two-molecule saturable model of DOC given in Tbl. 3.7. ● Awada et al. [63], ▲ Baker et al. [64]b, ■ Blagden et al. [65]b, ◆ Baker et al. [56]b.

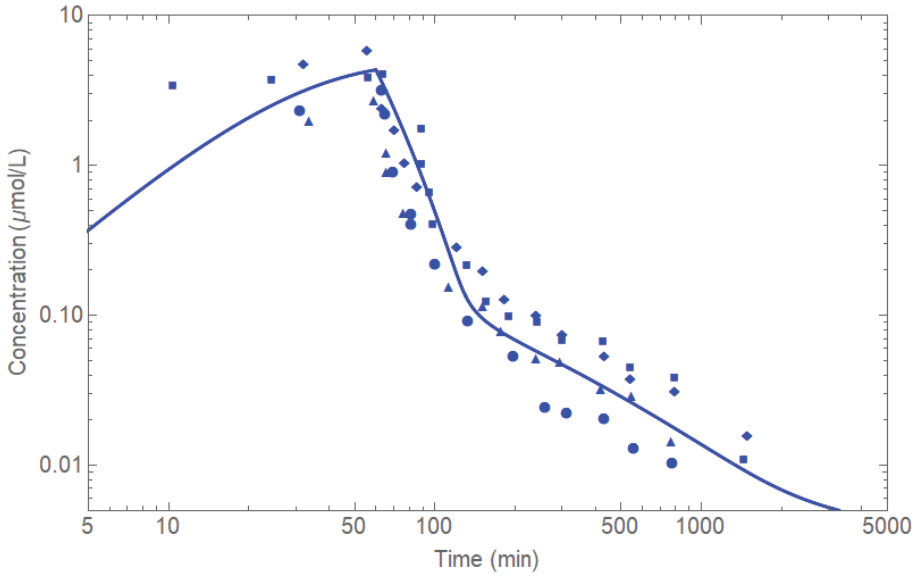



Figure A.19: Comparison between the one-compartment two-molecule saturable model of DOC and the evaluation data with a dosage of 100 mg/m$^2$ over 60 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the one-compartment two-molecule saturable model of DOC given in Tbl. 3.7.
● Clarke, et al. [32]a, ▲ Clarke, et al. [32]b, ■ Clarke, et al. [32]d, ◆ Clarke, et al. [32]e.

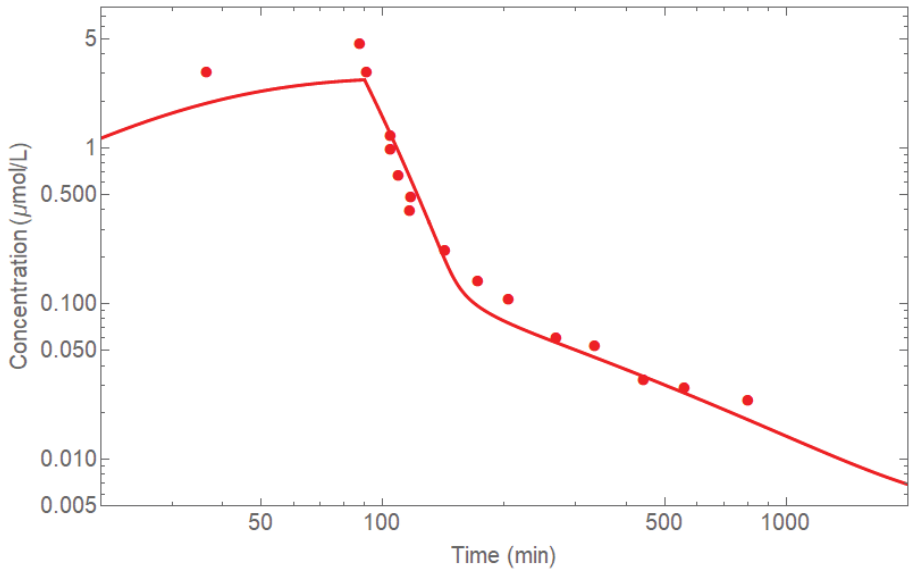



Figure A.20: Comparison between the one-compartment two-molecule saturable model of DOC and the evaluation data with a dosage of 100 mg/m$^2$ over 90 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the one-compartment two-molecule saturable model of DOC given in Tbl. 3.7.
● Clarke, et al. [32]c.

## A.3 Two-Compartment One-Molecule Model

### A.3.1 Calibration Data

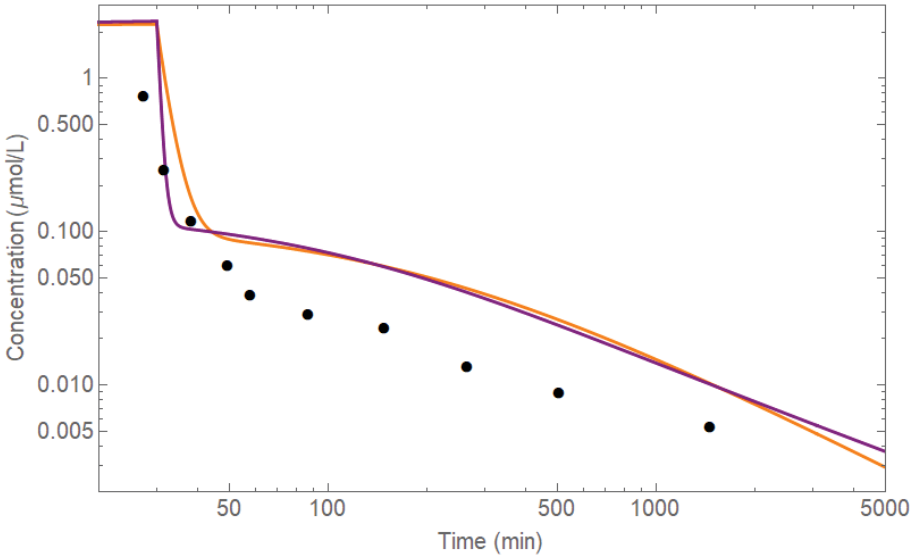



Figure A.21: Comparison between the two-compartment one-molecule models of DOC and a sample of calibration data with a dosage of 30 mg/m$^2$ over 30 min. The calibration data used is the data given in Tbl. 3.5. The two-compartment one-molecule models of DOC are given in Tbl. 3.11. ● Witta et al. [61]a, — Fractal, and — Fractal MM.

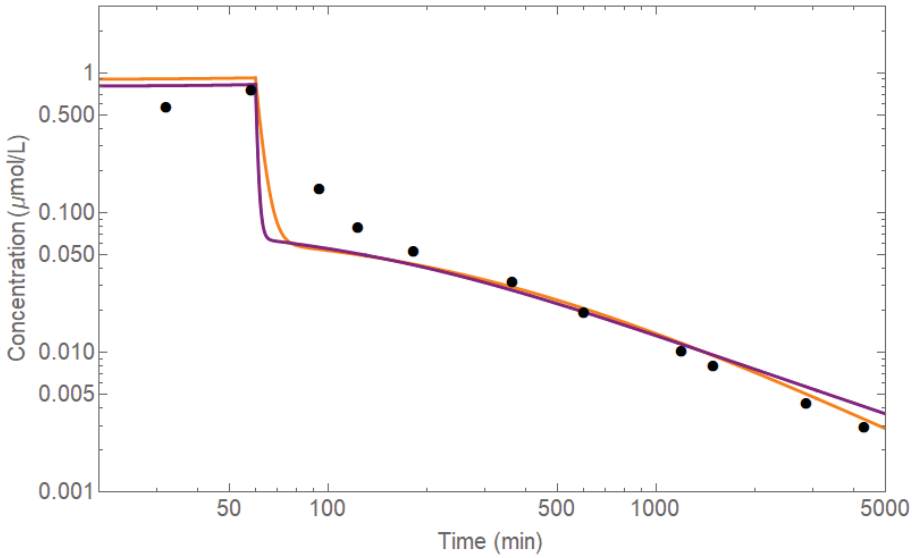



Figure A.22: Comparison between the two-compartment one-molecule models of DOC and a sample of calibration data with a dosage of 20 mg/m$^2$ over 60 min. The calibration data used is the data given in Tbl. 3.5. The two-compartment one-molecule models of DOC are given in Tbl. 3.11. ● Brunsvig et al. [60], — Fractal, and — Fractal MM.

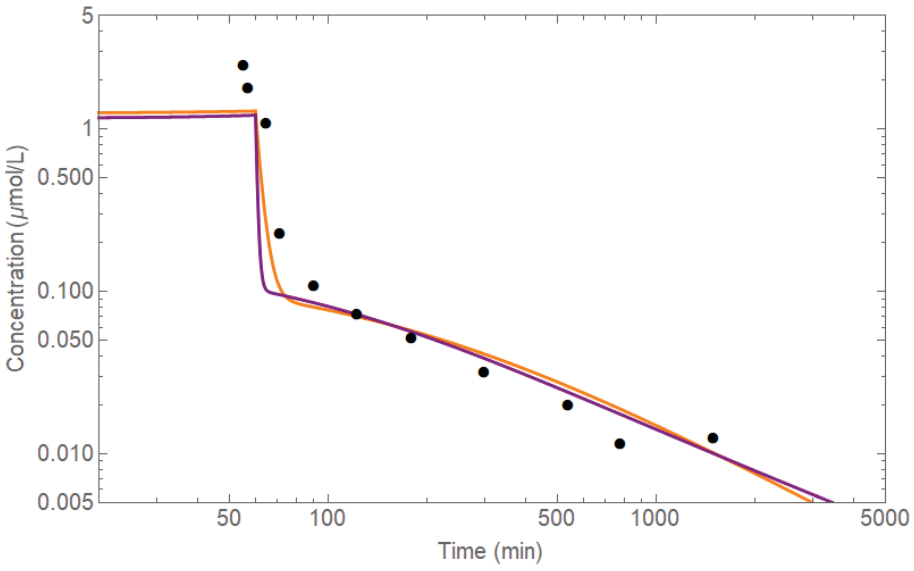



Figure A.23: Comparison between the two-compartment one-molecule models of DOC and a sample of calibration data with a dosage of 30 mg/m$^2$ over 60 min. The calibration data used is the data given in Tbl. 3.5. The two-compartment one-molecule models of DOC are given in Tbl. 3.11. ● Cox et al. [62]a, — Fractal, and — Fractal MM.

### A.3.2 Evaluation Data

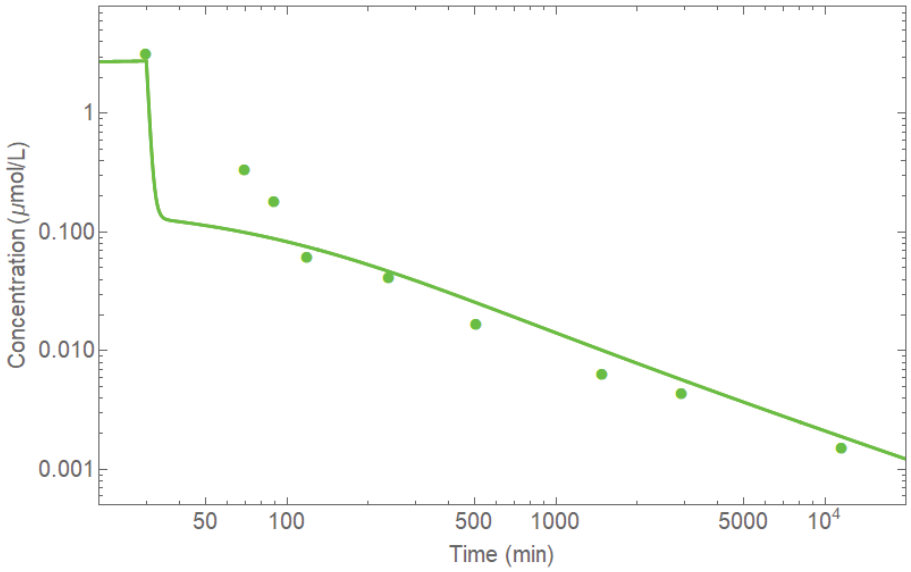



Figure A.24: Comparison between the two-compartment one-molecule fractal MM model of DOC and the evaluation data with a dosage of 35 mg/m$^2$ over 30 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the two-compartment one-molecule fractal MM model of DOC given in Tbl. 3.11.
● Baker, et al. [64]a.

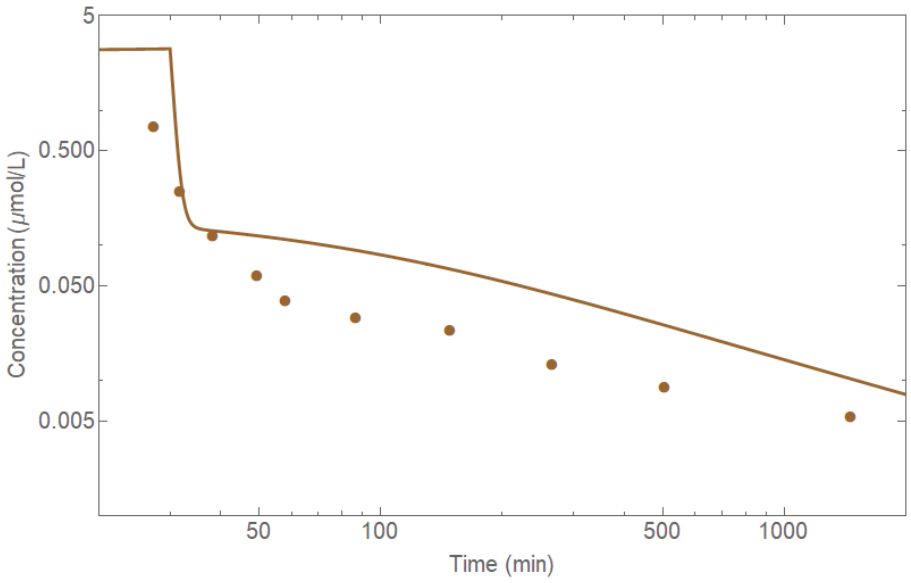



Figure A.25: Comparison between the two-compartment one-molecule fractal MM model of DOC and the evaluation data with a dosage of 36 mg/m$^2$ over 30 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the two-compartment one-molecule fractal MM model of DOC given in Tbl. 3.11.
● Witta, et al. [61]b.

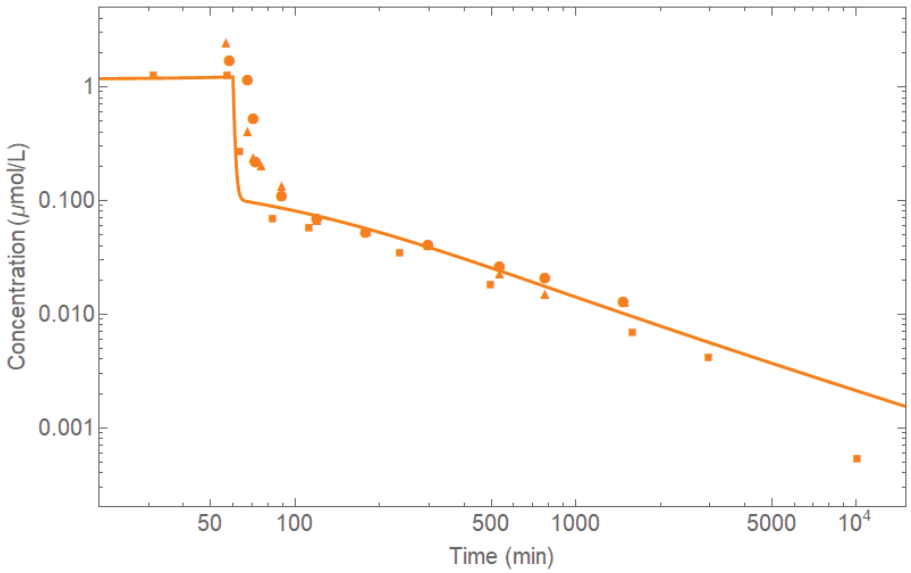



Figure A.26: Comparison between the two-compartment one-molecule fractal MM model of DOC and the evaluation data with a dosage of 30 mg/m$^2$ over 60 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the two-compartment one-molecule fractal MM model of DOC given in Tbl. 3.11.
● Cox, et al. [62]b, ▲ Cox, et al. [62]c, ■ Baker et al. [56]a.

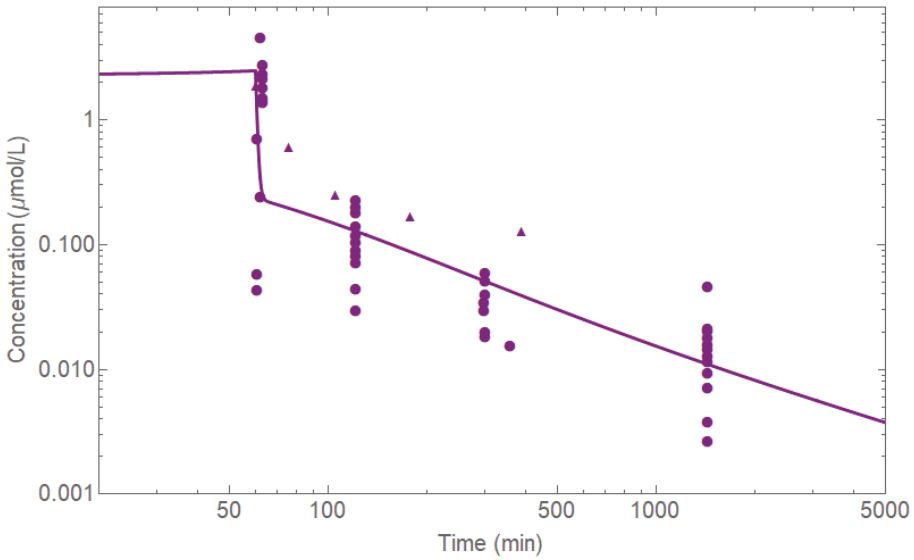



Figure A.27: Comparison between the two-compartment one-molecule fractal MM model of DOC and the evaluation data with a dosage of 60 mg/m$^2$ over 60 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the two-compartment one-molecule fractal MM model of DOC given in Tbl. 3.11.
● Blagden, et al. [64]a, ▲ El-Rayes, et al. [66].

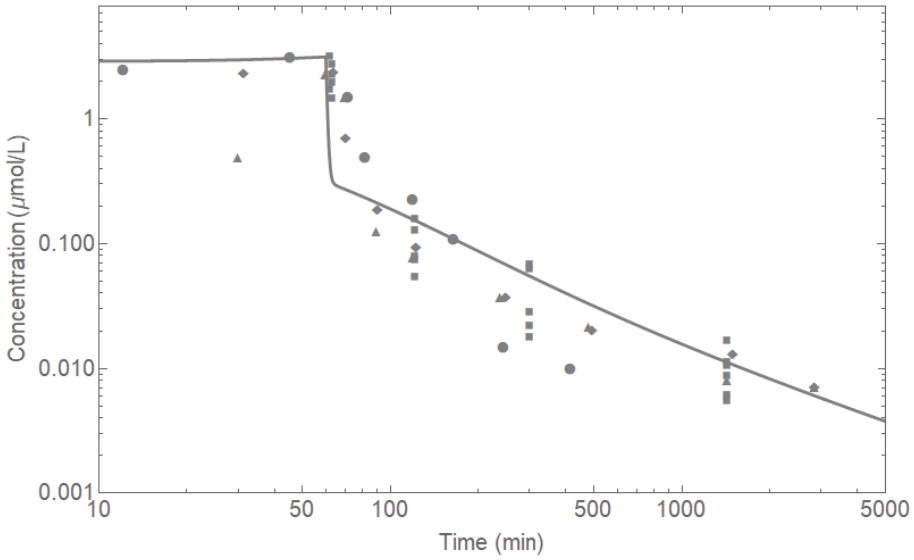



Figure A.28: Comparison between the two-compartment one-molecule fractal MM models of DOC and the evaluation data with a dosage of 75 mg/m$^2$ over 60 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the two-compartment one-molecule fractal MM model of DOC given in Tbl. 3.11. ● Awada et al. [63], ▲ Baker et al. [64]b, ■ Blagden et al. [65]b, ◆ Baker et al. [56]b.

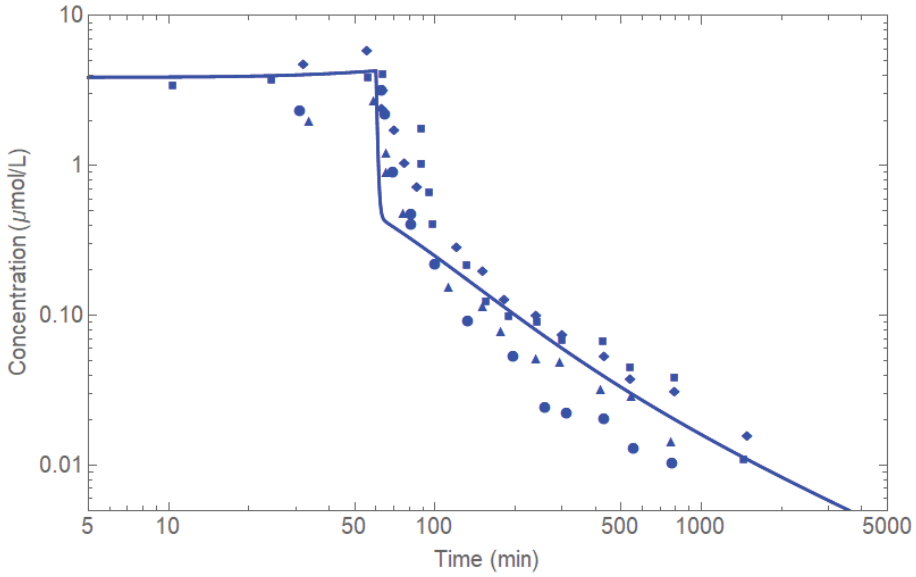



Figure A.29: Comparison between the two-compartment one-molecule fractal MM model of DOC and the evaluation data with a dosage of 100 mg/m$^2$ over 60 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the two-compartment one-molecule fractal MM model of DOC given in Tbl. 3.11.
● Clarke, et al. [32]a, ▲ Clarke, et al. [32]b, ■ Clarke, et al. [32]d, ◆ Clarke, et al. [32]e.

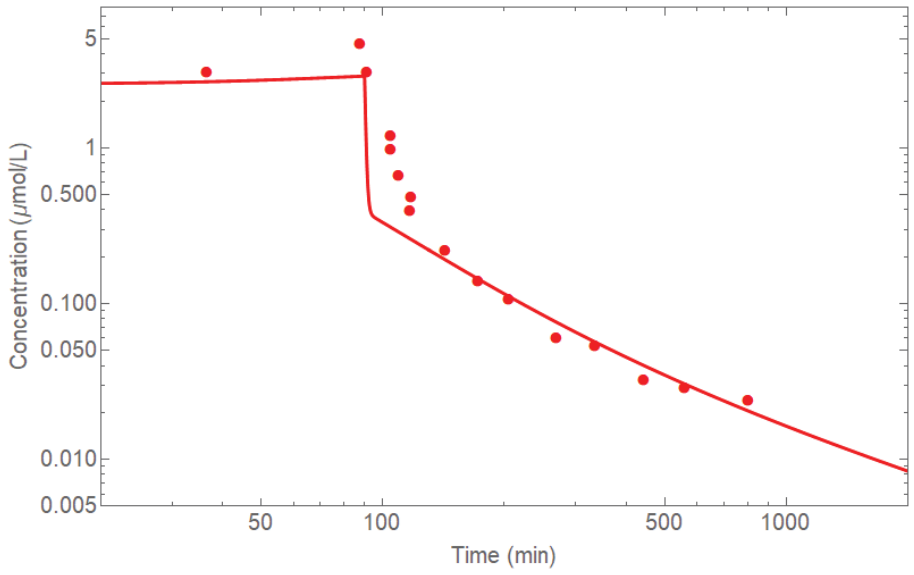



Figure A.30: Comparison between the two-compartment one-molecule fractal MM model of DOC and the evaluation data with a dosage of 100 mg/m$^2$ over 90 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the two-compartment one-molecule fractal MM model of DOC given in Tbl. 3.11.
● Clarke, et al. [32]c.

## A.4 Three-Compartment One-Molecule Models

### A.4.1 Calibration Data

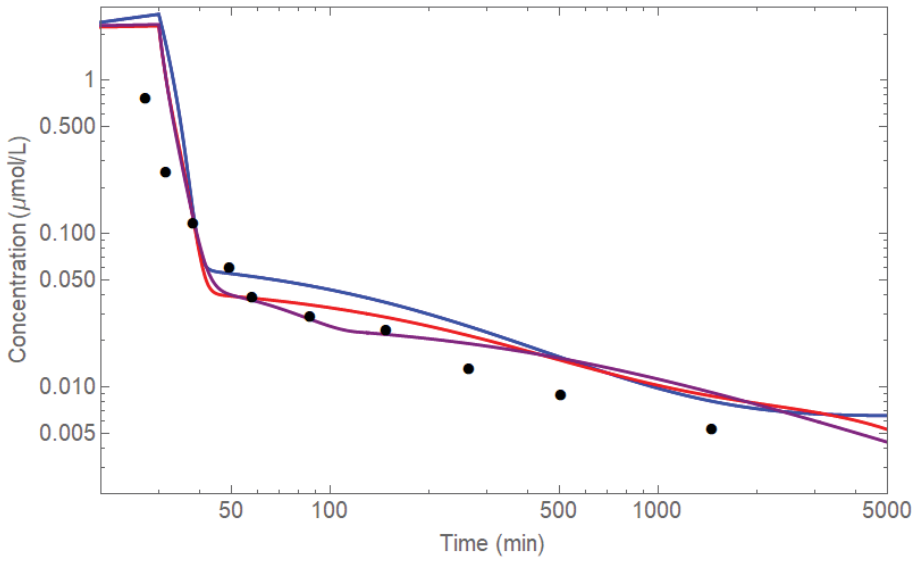



Figure A.31: Comparison between the three-compartment one-molecule models of DOC and a sample of calibration data with a dosage of 30 mg/m$^2$ over 30 min. The calibration data used is the data given in Tbl. 3.5. The three-compartment one-molecule models of DOC are given in Tbl. 3.13. ● Witta et al. [61]a, — Mixed Saturable, — Fractal, and — Fractal MM.

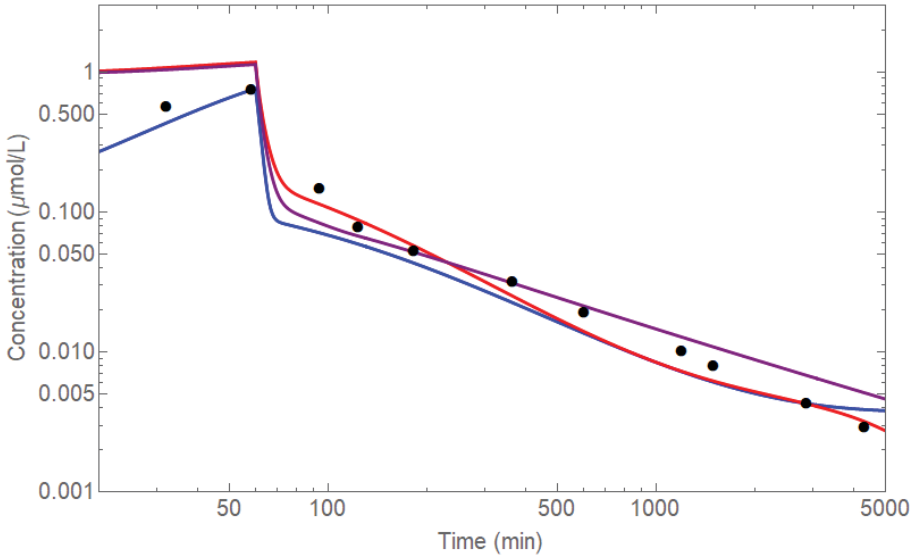



Figure A.32: Comparison between the three-compartment one-molecule models of DOC and a sample of calibration data with a dosage of 20 mg/m$^2$ over 60 min. The calibration data used is the data given in Tbl. 3.5. The three-compartment one-molecule models of DOC are given in Tbl. 3.13. ● Brunsvig et al. [60], — Mixed Saturable, — Fractal, and — Fractal MM.

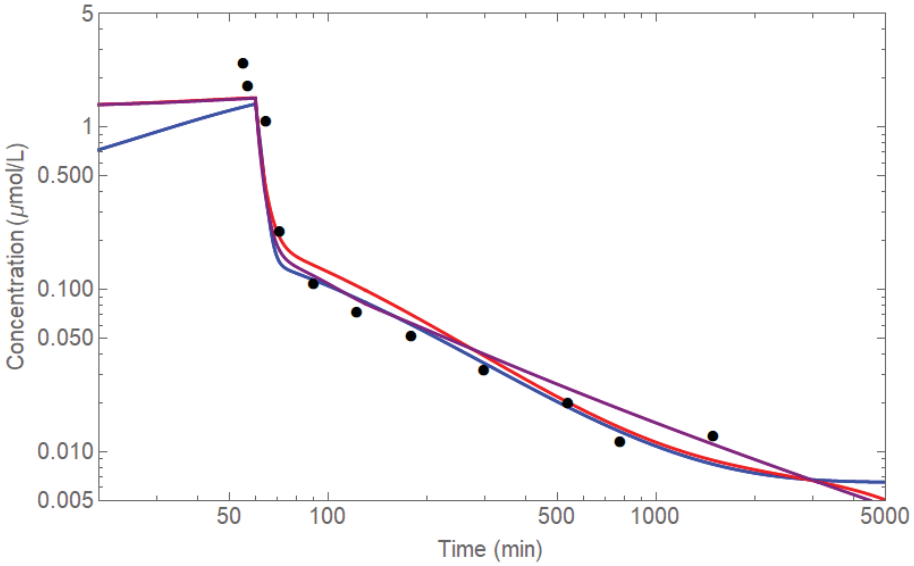



Figure A.33: Comparison between the three-compartment one-molecule models of DOC and a sample of calibration data with a dosage of 30 mg/m$^2$ over 60 min. The calibration data used is the data given in Tbl. 3.5. The three-compartment one-molecule models of DOC are given in Tbl. 3.13. ● Cox et al. [62]a, — Mixed Saturable, — Fractal, and — Fractal MM.

### A.4.2 Evaluation Data

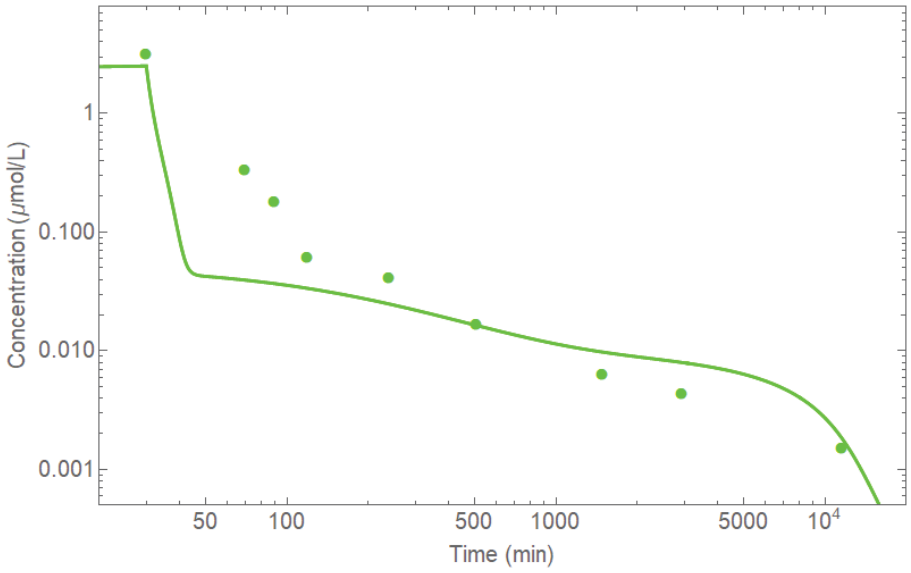



Figure A.34: Comparison between the three-compartment one-molecule fractal MM model of DOC and the evaluation data with a dosage of 35 mg/m$^2$ over 30 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the three-compartment one-molecule fractal MM model of DOC given in Tbl. 3.13.
● Baker, et al. [64]a.

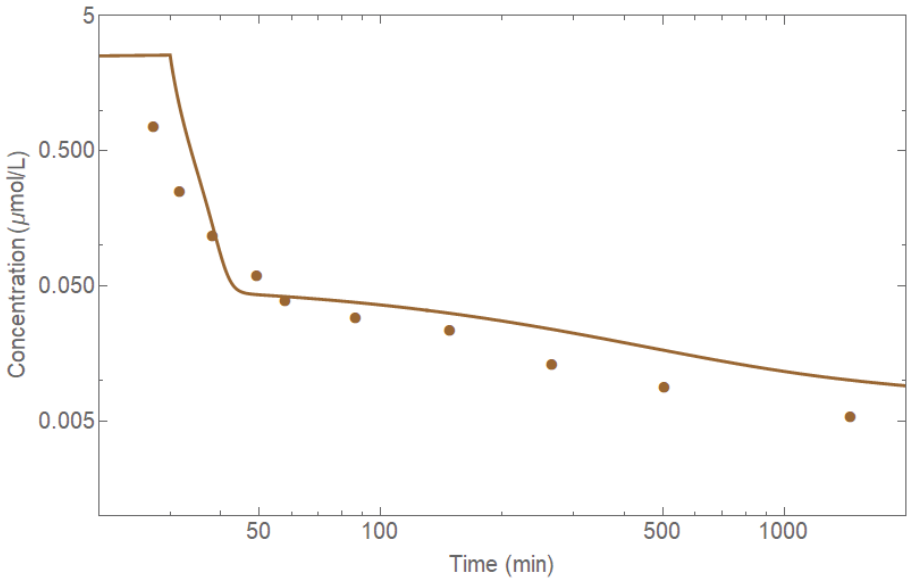



Figure A.35: Comparison between the three-compartment one-molecule fractal MM model of DOC and the evaluation data with a dosage of 36 mg/m$^2$ over 30 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the three-compartment one-molecule fractal MM model of DOC given in Tbl. 3.13.
● Witta, et al. [61]b.

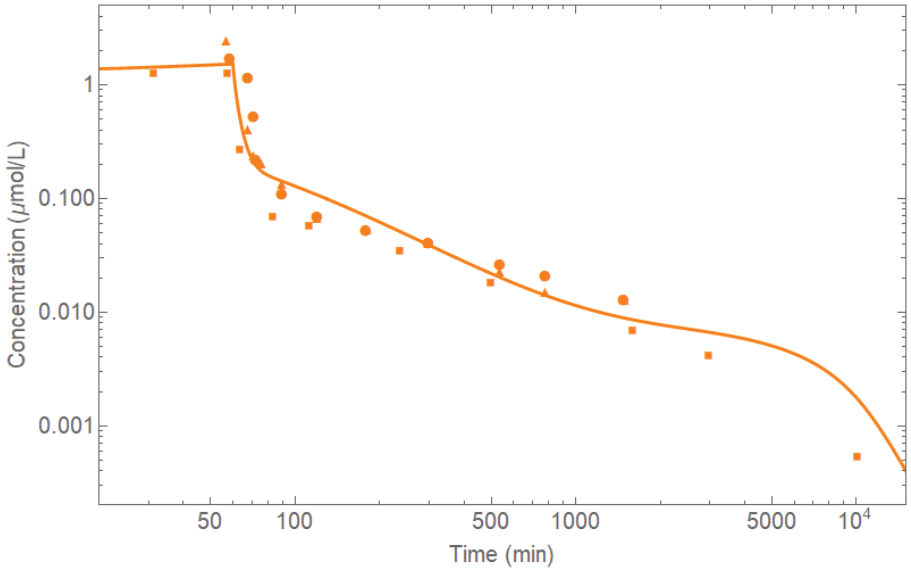



Figure A.36: Comparison between the three-compartment one-molecule fractal MM model of DOC and the evaluation data with a dosage of 30 mg/m$^2$ over 60 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the three-compartment one-molecule fractal MM model of DOC given in Tbl. 3.13.
● Cox, et al. [62]b, ▲ Cox, et al. [62]c, ■ Baker et al. [56]a.

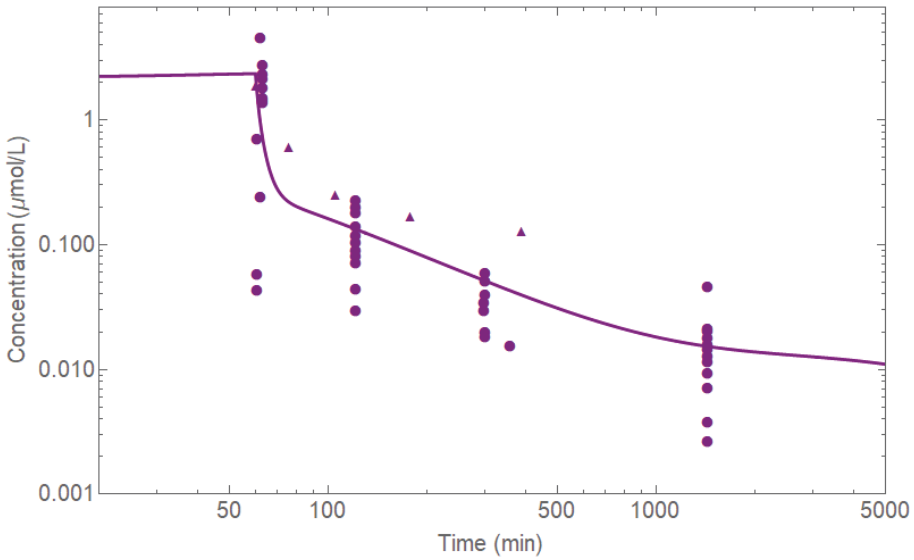



Figure A.37: Comparison between the three-compartment one-molecule fractal MM model of DOC and the evaluation data with a dosage of 60 mg/m$^2$ over 60 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the three-compartment one-molecule fractal MM model of DOC given in Tbl. 3.13.
● Blagden, et al. [64]a, ▲ El-Rayes, et al. [66].

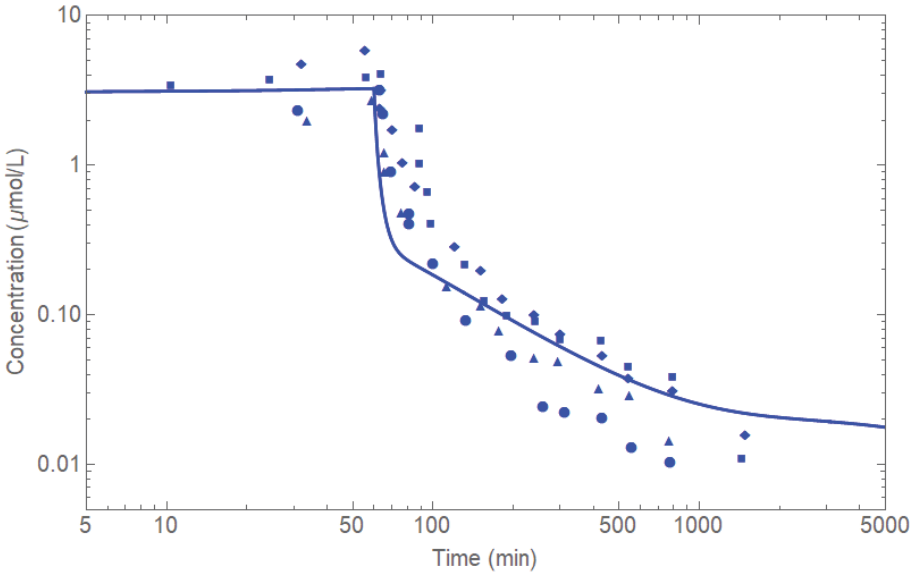



Figure A.38: Comparison between the three-compartment one-molecule fractal MM model of DOC and the evaluation data with a dosage of 100 mg/m$^2$ over 60 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the three-compartment one-molecule fractal MM model of DOC given in Tbl. 3.13.
● Clarke, et al. [32]a, ▲ Clarke, et al. [32]b, ■ Clarke, et al. [32]d, ◆ Clarke, et al. [32]e.

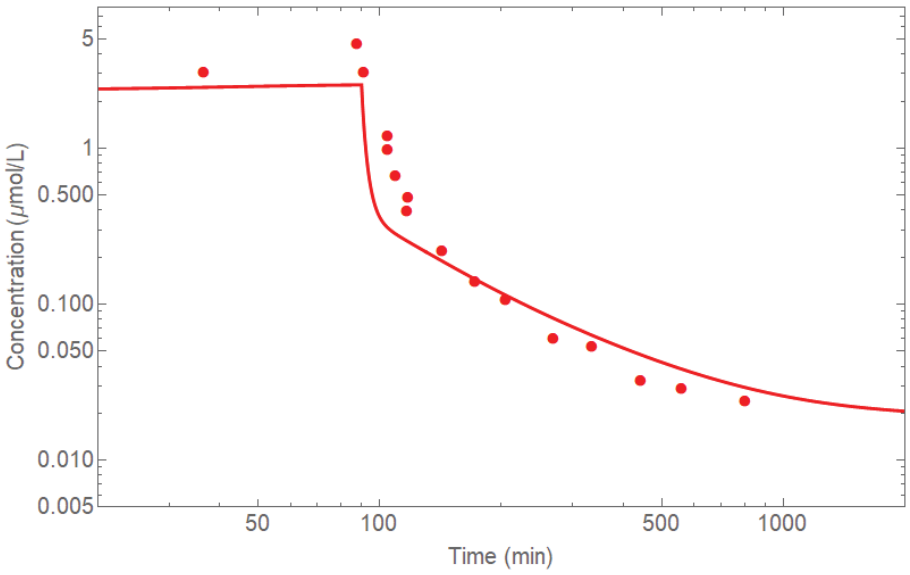



Figure A.39: Comparison between the three-compartment one-molecule fractal MM model of DOC and the evaluation data with a dosage of 100 mg/m$^2$ over 90 min. The evaluation data used is the data given in Tbl. 3.6. The model shown is the three-compartment one-molecule fractal MM model of DOC given in Tbl. 3.13.
● Clarke, et al. [32]c.

# Appendix B

# Additional Models

Many more models than those presented within chapter §3 were tested while developing a PK model of DOC. The models which failed to capture the behaviour of DOC and did not inform the final model are listed below along with the statistics which which illustrate this.

## B.1 One-Compartment One-Molecule Models

### B.1.1 P80 Models

Table B.1: One-compartment P80 model statistics.

| Statistic | Zeroth | Second | MM | Fractal + First | Two Fractal | Saturable |
|---|---|---|---|---|---|---|
| $\eta$ | 2 | 2 | 3 | 4 | 5 | 5 |
| $S_p$ | 0.449 | 0.392 | 0.040 | 0.00364 | 0.00364 | 0.0011 |
| $\text{AIC}_c$ | 7.201 | 6.385 | 1.956 | 6.024 | 62.31 | 55.39 |
| $\Delta^*$ | 19.5 | 18.7 | 14.3 | 18.4 | 74.6 | 67.7 |

$^*$ $AIC_{c,min} = -12.34$

### B.1.2 DOC Models

Table B.2: One-compartment one-molecule DOC model statistics.

| Statistic | MM | Zeroth | First | Second | Mixed | Fractal Second | Fractal MM | Two Fractal |
|---|---|---|---|---|---|---|---|---|
| $\eta$ | 3 | 2 | 2 | 2 | 4 | 4 | 4 | 5 |
| $S_p$ | 1.084 | 1.287 | 1.084 | 0.460 | 0.460 | 0.328 | 0.344 | 0.327 |
| $\text{AIC}_c$ | 78.27 | 225.8 | 78.27 | -670.6 | -666.5 | -961.0 | -919.4 | -960.2 |
| $\Delta^*$ | 1610 | 1758 | 1610 | 861 | 865 | 571 | 613 | 572 |

$^*$ $AIC_{c,min} = -1532$

## B.2 One-Compartment Two-Molecule Models

Table B.3: One-compartment two-molecule DOC model statistics.

| Statistic | First | Mixed Fractal | Mixed Saturable | MM |
|---|---|---|---|---|
| $\eta$ | 4 | 5 | 7 | 7 |
| $S_p$ | 0.418 | 0.308 | 0.319 | 0.416 |
| $AIC_c$ | -750.2 | -1013 | -980.5 | -747.4 |
| $\Delta^*$ | 782 | 519 | 552 | 785 |

* $AIC_{c,min} = -1532$

## B.3 Two-Compartment One-Molecule Models

Table B.4: Two-compartment one-molecule DOC model statistics.

| Statistic | First | Mixed Fractal | MM |
|---|---|---|---|
| $\eta$ | 4 | 5 | 7 |
| $S_p$ | 0.418 | 0.308 | 0.415 |
| $AIC_c$ | -750.3 | -1013 | -747.4 |
| $\Delta^*$ | 782 | 517 | 785 |

* $AIC_{c,min} = -1532$

## B.4 Three-Compartment One-Molecule Models

Table B.5: Three-compartment one-molecule DOC model statistics.

| Statistic | First Order | Mixed Fractal | MM |
|---|---|---|---|
| $\eta$ | 6 | 7 | 11 |
| $S_p$ | 0.250 | 0.195 | 0.221 |
| $AIC_c$ | -1192 | -1407 | -1292 |
| $\Delta^*$ | 340 | 125 | 240 |

* $AIC_{c,min} = -1532$

# Appendix C

# Code Used in Modelling DOC

Custom code was written to efficiently model DOC and generate relevant statistics and metrics for the models. The code will fit a specific model to digitized clinical data sets. Some small adjustments need to be made to the code depending on the model under consideration. The functions `eqn` and `setEqns` contol the differential equations and thus must match with the model under consideration. The code given below is suitable for solving one-molecule DOC models for one- two- and three-compartments. It can be adjusted by modifying the necessary DE term functions within `eqn` and then ensuring that the equations are controlled properly within `setEqns`.

## C.1 Pharmacokinetic Model of Docetaxel (PMoD)

```
# include <iostream>
# include <fstream>
# include <cmath>
# include <iomanip>
# include <array>
# include <ctime>

using namespace std;

// DATA TYPES

struct dataSetInfo{ int numDat;
                    int X;
                    int numPat;
                    double yvtDat[6][2500];
                    double yvtNumDat[6][2500];
                    double yvtCovgTestDat[6][10];
                    int numDatW;};           // Data set information

struct theoInfo{ double yvtTheoDat[6][2500];
                    };                       // Holds solutions

// GOLBAL VARIABLES

int numEqn = 5;       // Number of differential equations
int numDataSet = 1;     // Active number of data sets being solved
```

```
int numParam = 116;     // Total parameters in the model
int numParamON = 0;     // Total fitting parameters in the model

double tol = 0.000001;   // Reference tolerance

double infConst[32],
       infOnT[32],
       infT[32];      // Infusion information

double sol[32][6];      // Holds the initial step for RK4()
double sol2[32][6];     // Holds the second step for RK4()
double param[203];      // Parameter set
double param0[203];     // Holds the parameter set
double paramReturn[203]; //
double paramSmallest[203]; // Holds the parameter set with the smallest VAR()
int ONOFF[203];         // Tracks fitting parameters
double paramErr[2][203]; // Holds range for 1% VAR in parameters

double storeSol[32][6];  // Temp solution storage

int eqnControl[5];      // Number of equations

dataSetInfo dataSet[32];  // Holds all data set information

double powBasis[203][203]; // Holds the basis used in POWELL()

double VAR0,          // Initial VAR
       VARend,        // Final VAR
       aic,           // Temp. AIC
       aicc,          // Temp. AICc
       aicg,          // Temp. AIC for collective data sets
       aiccg,         // Temp. AICc for collective data sets
       VARg,          // VAR for collective data sets
       minParamVAR;      // Used for printing VAR in statistics
int    aiccTest,      // Checks is AICc should be used
       aiccTestg;     // Checks is AICc should be used for collective data
          sets

int extVAR=0; // 0 = Regular Data, 1 = Extrap Data

double smallVAR; // Used only in debugging

double AUC[9],
       AUC24[9],
       AUC24post[9],
       Cmax[9]={0},
       Tmax[9],
       TmaxF[9],
```

```
      ThalfD[9],
      ChalfD[9];     // PK metrics

// FUNCTIONS

void loadDataSets(int j);  // Loads data set j
void loadMetricDataSets();  // Loads 'continuous' data sets used for metrics
void loadSmoothDataSets();  // Loads 'continuous' data sets used for figures
void loadParameters();     // Legacy
void loadExtParameters(int i); // Loads parameter set j
void printDataSets();      // Prints the data set (used for debugging)

double eqn(int i, int j, double t, double X0, double X1, double X2, double
    X3, double X4); \\ The differential equations, i is the equation, j is
    the data set, t is the time, X_ is the quantity of molecule in a
    compartment

// DE terms: n is the associated equation, XA/B is the associated quantity
double infusion(int j, double t);       // Infusion DE term
double flow(int n, double XA);          // Flow DE term
double change(int n, double XA);     // Transformation DE term
double changeD(int n, double XA);       // Transformation DE term with delta
double binding(int n, double XA, double XB); // Binding DE term
double metabolize(int n, double XA);    // Metabolising DE term
double eliminate(int n, double XA);     // Elimination DE term

double infusionStep(int j, double t);   // Infusion DE term w/ H(XA)
double flowStep(int n, double XA);        // Flow DE term w/ H(XA)
double changeStep(int n, double XA);      // Transformation DE term w/ H(XA)
double changeDStep(int n, double XA);     // Transformation DE term with
    delta w/ H(XA)
double bindingStep(int n, double XA, double XB); // Binding DE term w/
    H(XA)*H(XB)
double metabolizeStep(int n, double XA);   // Metabolising DE term w/ H(XA)
double eliminateStep(int n, double XA);      // Elimination DE term w/ H(XA)

float H(double X); // Step function

void setEqns(int j, int k);     // Determines if Reg or Step DE terms are used

void presetInitialConditions();  // Initializes the conditions
void setInitialConditions(int i); // Uses solution to initialize next step
    in RK4()
void RK4();                // Solves DEs
bool convgTest(int j);       // Test is the solution in converged w/in
    tolerance
bool negTest(int j);       // Tests if any solutions (quantities) are negative
```

```
void POWELL(int p);          // Minimizes the variance by adjusting the
    parameters
double VAR();              // Computes the variance between RK4 solutions and
    data sets
void AIC();                // Computes AIC and AICc
void printStats();           // Prints variance for all data sets
void setNumParamON();        // Determines number of active fitting
    parameters
void multiParam(int j, double step, int sign); // Multiplies param set by a
    scalar
void updateBasis();            // Updates vectors used for param search
void resetBasis();             // Resets vectors to their initial directions
void printBasis();             // Prints the basis (used in debugging)
void printParamSample();       // Prints select parameters (used in
    debugging)

void printSolutions();       // Print the quantity and concentrations of the
    molecules in each compartment
void printMetricSolutions();   //Prints concentrations used in metric
    calculations
void printConcentrations();   // Called in printSolutions
void printParameters(int j);  // Prints parameters j

void generateMetricData(int j);  // Generates solutions for metric data sets
void generateWolframSolutions(); // Generates solutions for use in figures
    (in Mathematica)
void generateExtWolframSolutions(int k); // Generates solutions for use in
    figures (in Mathematica)

void generateParamErr(int j);  // Find param values with 1% difference to VAR
void printParameterErr(int j);  // Prints parameter errors

void METRICSgenerator(int k);   // Generates the metrics (AUC-Cmax-t1/2)

int main(){

  clock_t time;
  time = clock(); // Starts timer



  presetInitialConditions(); // Initialize conditions

  eqnControl[0]=0;
  eqnControl[1]=1;
  eqnControl[2]=2;
  eqnControl[3]=3;
  eqnControl[4]=4;      // Initialize equations
```

```
/// INITIAL FITS ////////////////////////////////////////////////////////
for(int n=0; n<6; n++){

       int j=0;
       if(n==0){j=19;};  // Baker05
       if(n==1){j=15;};  // Lim
       if(n==2){j=16;};  // Robert
       if(n==3){j=5;};  // Brunsvig
       if(n==4){j=17;}; // Witta(a)
       if(n==5){j=13;}; // Cox (a)

     loadExtParameters(j);
     loadDataSets(j);

     RK4();
     smallVAR=VAR();      // Compute reference variance

   do{
     VAR0=VAR();

     POWELL(j);

     VARend=VAR();

   }while(VARend<VAR0 && ( (VAR0-VARend)/VAR0>100*tol ||
       abs(VAR0-VARend)>(100*tol) ));

   generateParamErr(j);
   printParameterErr(j);
   printSolutions();

};
   generateWolframSolutions();

   cout << '\a'; /// BEEEEEEP

/// END INITIAL FITS ////////////////////////////////////////////////////

/// GLOBAL FITS /////////////////////////////////////////////////////////

double initVAR,
     globeVAR;

minParamVAR =10000;
```

```
for(int n=0; n<6; n++){

        int j=0;
        if(n==0){j=19;};  // Baker05
        if(n==1){j=15;};  // Lim
        if(n==2){j=16;};  // Robert
        if(n==3){j=5;};  // Brunsvig
        if(n==4){j=17;}; // Witta(a)
        if(n==5){j=13;}; // Cox (a)


   loadDataSets(21);  // Data set 21 combines the data sets above together
   loadExtParameters(j);
   RK4();
   initVAR=VAR();
   loadExtParameters(j+100);
   RK4();
   globeVAR=VAR();

   if(initVAR<globeVAR){
   loadExtParameters(j);};
   printParameters(100+j);    // Set parameters based on smallest VAR (useful
       when run is interupted)

   loadExtParameters(j+100);

   RK4();
   smallVAR=VAR();


   do{
      VAR0=VAR();

      POWELL(100+j);

      VARend=VAR();

   }while(VARend<VAR0 && ( (VAR0-VARend)/VAR0>10*tol ||
       abs(VAR0-VARend)>(10*tol) ));

    if(minParamVAR>VARend){
    minParamVAR = VARend;
    printParameters(21);};

};
   generateParamErr(21);
   printParameterErr(21);
```

```
   generateExtWolframSolutions(21);

/// END GLOBAL FITS ////////////////////////////////////////////////////


/// GENERATE STATISTICS ////////////////////////////////////////////////

   AIC();
   printStats();

/// END GENERATE STATISTICS ////////////////////////////////////////////

/// GENERATE PK METRICS (AUC -- T 1/2 -- C max) ////////////////////////

     METRICSgenerator(21); // Generates metrics for the final parameter set

/// END GENERATE PK METRICS ////////////////////////////////////////////


    time = clock() - time;
    cout << endl << endl << (time/CLOCKS_PER_SEC)/3600 << " hr "
                      << (time/CLOCKS_PER_SEC)/60 -
                          floor((time/CLOCKS_PER_SEC)/3600)*60 << " min "
                      << (time/CLOCKS_PER_SEC) -
                          floor((time/CLOCKS_PER_SEC)/3600)*3600 -
                          floor((time/CLOCKS_PER_SEC)/60 -
                          floor((time/CLOCKS_PER_SEC)/3600)*60)*60 << " sec "
                          << endl; // Prints time it took for program to run


    cout << '\a'; /// BEEEEEEP

  return 0;
}

///////////////////////////// FUNCTIONS ////////////////////////////////

void loadDataSets(int j){

    ifstream fin;

    string toss;

                numDataSet=1;
       if(j==3){numDataSet=15;};
       if(j==4){numDataSet=6;};
       if(j==19){numDataSet=1;};
       if(j==20){numDataSet=3;};
```

```
        if(j==21){numDataSet=6;};
        if(j==22){numDataSet=31;};

    for(int i=0; i<numDataSet; i++){

        fin.open("Data/Data"+to_string(j)+"/Set"+ to_string(i) + ".txt");

        fin >> toss >> infConst[i];

        fin >> toss >> infOnT[i];

        fin >> toss >> infT[i];

        fin >> toss >> dataSet[i].numDat;

        fin >> toss >> dataSet[i].numPat;

        fin >> toss >> dataSet[i].X;

        fin >> toss >> toss;

        for(int j=0; j<dataSet[i].numDat; j++){

            fin >> dataSet[i].yvtDat[0][j] >>
                dataSet[i].yvtDat[dataSet[i].X][j];
        };

        fin.close();
    };
};


void loadMetricDataSets(){

    ifstream fin;

    string toss;

    numDataSet = 9;
    for(int n=0;n<numDataSet;n++){

        fin.open("Data/Data_Metric/Set"+ to_string(n) + ".txt");

        fin >> toss >> infConst[n];

        fin >> toss >> infOnT[n];

        fin >> toss >> infT[n];
```

```
        fin >> toss >> dataSet[n].numDat;

        fin >> toss >> dataSet[n].numPat;

        fin >> toss >> dataSet[n].X;

        fin >> toss >> toss;

        for(int i=0; i<dataSet[n].numDat; i++){

            fin >> dataSet[n].yvtDat[0][i] >>
                dataSet[n].yvtDat[dataSet[n].X][i];
        };

        fin.close();
    };
};

void loadSmoothDataSets(){

    ifstream fin;

    string toss;

    numDataSet = 10;

    for(int i=0;i<numDataSet;i++){

        fin.open("Data_Smooth/Set"+ to_string(i) + ".txt");

        fin >> toss >> infConst[i];

        fin >> toss >> infOnT[i];

        fin >> toss >> infT[i];

        fin >> toss >> dataSet[i].numDat;

        fin >> toss >> dataSet[i].numPat;

        fin >> toss >> dataSet[i].X;

        fin >> toss >> toss;

        for(int j=0; j<dataSet[i].numDat; j++){
```

```
            fin >> dataSet[i].yvtDat[0][j] >>
                dataSet[i].yvtDat[dataSet[i].X][j];
        };

        fin.close();
    };

};

void loadParameters(){

    ifstream fin;

    string toss;

        fin.open("Parameters/Parameters.txt");

        fin >> toss >> toss >> toss;

        for(int i=0; i<numParam ;i++){

            fin >> toss >> ONOFF[i] >> param[i];
        };

        fin.close();
};

void loadExtParameters(int i){

    ifstream fin;

    string toss;

        fin.open("Parameters/Parameters_"+ to_string(i) + ".txt");

        fin >> toss >> toss >> toss;

        for(int i=0; i<numParam ;i++){

            fin >> toss >> ONOFF[i] >> param[i];
        };

        fin.close();
};

void printDataSets(){

    for(int i=0; i<numDataSet; i++){
```

```
        cout << "Data Set " << i << ":" << endl;
        cout << "NumDat " << dataSet[i].numDat << endl;
        cout << "NumPat " << dataSet[i].numPat << endl;
        cout << "INF C: " << infConst[i] << '\t' << "INF T: " << infT[i] <<
            '\t' << "Start T: " << infOnT[i] << endl;

        for(int j=0; j<dataSet[i].numDat; j++){

            cout << dataSet[i].yvtDat[0][j] << '\t' <<
                dataSet[i].yvtDat[dataSet[i].X][j] << endl;;
        };

        cout << endl << endl;
    };
};

double eqn(int i, int j, double t, double X0, double X1, double X2, double
    X3, double X4){

    double val;

    switch(i){
    case 0:
        val =   - flow(12,X0) + flow(24,X1) - flow(36,X0) + flow(48,X2); //
            DOC in C1 -> dX0
        break;
    case 1:
        val =     flow(12,X0) - flow(24,X1); // DOC in C2 -> dX1
        break;
    case 2:
        val =     flow(36,X0) - flow(48,X2) - eliminate(60,X2); // DOC in C3
            -> dX2
        break;
    case 3:
        val = 0; // dX3
        break;
    case 4:
        val = 0; // dX4
        break;
    case 5:
        val = - eliminateStep(0,X0) - flowStep(12,X0) + flowStep(24,X1) -
            flowStep(36,X0) + flowStep(48,X2); // DOC in C1 when <= 0
        break;
    case 6:
        val =   flowStep(12,X0) - flowStep(24,X1); // DOC in C2 when <= 0
        break;
    case 7:
```

```
        val = flowStep(36,X0) - flowStep(48,X2) - eliminateStep(60,X2); //
            DOC in C3 when <= 0
        break;
    case 8:
        val = 0; // dX3
        break;
    case 9:
        val = 0; // dX4
        break;
    case 10:
        val = infConst[j] - flow(12,X0) + flowStep(24,X1) - flow(36,X0) +
            flowStep(48,X2); // DOC in C1 inf ON
        break;
    case 11:
        val = 0;  // X=0
        break;
    default:
        cout << "ERROR CHOOSING EQUATION" << endl;
    };

    return val;
};

double infusion(int j, double t){

    return infConst[j];
};

double flow(int n, double XA){

    double val;

    val = (param[n+0]*pow(XA,param[n+3]))/(1+param[n+6]*pow(XA,param[n+9]))
         +(param[n+1]*pow(XA,param[n+4]))/(1+param[n+7]*pow(XA,param[n+10]))
         +(param[n+2]*pow(XA,param[n+5]))/(1+param[n+8]*pow(XA,param[n+11]));

    return val;
};

double change(int n, double XA){

    double val;

    val = (param[n+0]*pow(XA,param[n+6]))/(1+param[n+9]*pow(XA,param[n+12]))
         +(param[n+1]*pow(XA,param[n+7]))/(1+param[n+10]*pow(XA,param[n+13]))
         +(param[n+2]*pow(XA,param[n+8]))/(1+param[n+11]*pow(XA,param[n+14]));

    return val;
```

```
};

double changeD(int n, double XA){

    double val;

    val = (param[n+0]*param[n+3]*pow(XA,param[n+6]))/
         (1+param[n+9]*pow(XA,param[n+12]))
          +(param[n+1]*param[n+4]*pow(XA,param[n+7]))/
           (1+param[n+10]*pow(XA,param[n+13]))
          +(param[n+2]*param[n+5]*pow(XA,param[n+8]))/
           (1+param[n+11]*pow(XA,param[n+14]));

    return val;
};

double binding(int n, double XA, double XB){

    double val;

    val = (param[n+0]*param[n+3]*pow(XA,param[n+6])*pow(XB,param[n+9]))/
           (1+param[n+12]*pow(XA,param[n+15])*pow(XB,param[n+18]))
          +(param[n+1]*param[n+4]*pow(XA,param[n+7])*pow(XB,param[n+10]))/
           (1+param[n+13]*pow(XA,param[n+16])*pow(XB,param[n+19]))
          +(param[n+2]*param[n+5]*pow(XA,param[n+8])*pow(XB,param[n+11]))/
           (1+param[n+14]*pow(XA,param[n+17])*pow(XB,param[n+20]));

    return val;
};

double metabolize(int n, double XA){

    double val;

    val = (param[n+0]*param[n+3]*pow(XA,param[n+6]))/
           (1+param[n+9]*pow(XA,param[n+12]))
          +(param[n+1]*param[n+4]*pow(XA,param[n+7]))/
           (1+param[n+10]*pow(XA,param[n+13]))
          +(param[n+2]*param[n+5]*pow(XA,param[n+8]))/
           (1+param[n+11]*pow(XA,param[n+14]));

    return val;
};

double eliminate(int n, double XA){

    double val;
```

```
    val = (param[n+0]*pow(XA,param[n+3]))/(1+param[n+6]*pow(XA,param[n+9]))
         +(param[n+1]*pow(XA,param[n+4]))/(1+param[n+7]*pow(XA,param[n+10]))
         +(param[n+2]*pow(XA,param[n+5]))/(1+param[n+8]*pow(XA,param[n+11]));

    return val;
};

double flowStep(int n, double XA){

    double val;

    val = (param[n+0]*pow(XA,param[n+3]))/
          (1+param[n+6]*pow(XA,param[n+9]))*H(XA)
         +(param[n+1]*pow(XA,param[n+4]))/
          (1+param[n+7]*pow(XA,param[n+10]))
         +(param[n+2]*pow(XA,param[n+5]))/
          (1+param[n+8]*pow(XA,param[n+11]));

    return val;
};

double changeStep(int n, double XA){

    double val;

    val = (param[n+0]*pow(XA,param[n+6]))/
          (1+param[n+9]*pow(XA,param[n+12]))*H(XA)
         +(param[n+1]*pow(XA,param[n+7]))/
          (1+param[n+10]*pow(XA,param[n+13]))
         +(param[n+2]*pow(XA,param[n+8]))/
          (1+param[n+11]*pow(XA,param[n+14]));

    return val;
};

double changeDStep(int n, double XA){

    double val;

    val = (param[n+0]*param[n+3]*pow(XA,param[n+6]))/
          (1+param[n+9]*pow(XA,param[n+12]))*H(XA)
         +(param[n+1]*param[n+4]*pow(XA,param[n+7]))/
          (1+param[n+10]*pow(XA,param[n+13]))
         +(param[n+2]*param[n+5]*pow(XA,param[n+8]))/
          (1+param[n+11]*pow(XA,param[n+14]));

    return val;
};
```

```
double bindingStep(int n, double XA, double XB){

    double val;

    val = (param[n+0]*param[n+3]*pow(XA,param[n+6])*pow(XB,param[n+9]))/
          (1+param[n+12]*pow(XA,param[n+15])*pow(XB,param[n+18]))*H(XA)*H(XB)
         +(param[n+1]*param[n+4]*pow(XA,param[n+7])*pow(XB,param[n+10]))/
          (1+param[n+13]*pow(XA,param[n+16])*pow(XB,param[n+19]))
         +(param[n+2]*param[n+5]*pow(XA,param[n+8])*pow(XB,param[n+11]))/
          (1+param[n+14]*pow(XA,param[n+17])*pow(XB,param[n+20]));

    return val;
};

double metabolizeStep(int n, double XA){

    double val;

    val = (param[n+0]*param[n+3]*pow(XA,param[n+6]))/
          (1+param[n+9]*pow(XA,param[n+12]))*H(XA)
         +(param[n+1]*param[n+4]*pow(XA,param[n+7]))/
          (1+param[n+10]*pow(XA,param[n+13]))
         +(param[n+2]*param[n+5]*pow(XA,param[n+8]))/
          (1+param[n+11]*pow(XA,param[n+14]));

    return val;
};

double eliminateStep(int n, double XA){

    double val;

    val = (param[n+0]*pow(XA,param[n+3]))/
          (1+param[n+6]*pow(XA,param[n+9]))*H(XA)
         +(param[n+1]*pow(XA,param[n+4]))/
          (1+param[n+7]*pow(XA,param[n+10]))
         +(param[n+2]*pow(XA,param[n+5]))/
          (1+param[n+8]*pow(XA,param[n+11]));

    return val;
};

float H(double X){

    if(X < 0){return 0;}
      else{ if(abs(X)<tol){
            return 0;}
```

```
      else{ return 1;};};
};

void setEqns(int j, int k){

    int neg = 0;
    int toggle = 0;


  eqnControl[0]=0;
  eqnControl[1]=1;
  eqnControl[2]=2;
  eqnControl[3]=3;
  eqnControl[4]=4;

    if( (sol[j][1]<=0 || sol[j][2]<=0 || sol[j][3]<=0 ||
        isnan(sol[j][1]) || isnan(sol[j][2]) || isnan(sol[j][3])) &&
            (sol[j][0]>0) )
    {
        eqnControl[0] = 5;
        eqnControl[1] = 6;
        eqnControl[2] = 7;
    };


    if( (infOnT[j]<=dataSet[j].yvtDat[0][k])&&
        (dataSet[j].yvtDat[0][k]<(infOnT[j]+infT[j])) )
    {
        eqnControl[0] = 10;
    }
    else{
    if( sol[j][1]<=0 || sol[j][2]<=0 || sol[j][3]<=0 ||
        isnan(sol[j][1]) || isnan(sol[j][2]) || isnan(sol[j][3]))
    {
        eqnControl[0] = 5;
        eqnControl[1] = 6;
        eqnControl[2] = 7;
    }};


    for(int i=0; i<numEqn; i++){
    if(sol[j][0]>2*(infOnT[j]+infT[j]) && sol[j][i+1] == 0){eqnControl[i] =
        11;}
    };

};

void presetInitialConditions(){
```

```
     for(int i=0; i<numDataSet; i++){
        dataSet[i].yvtNumDat[0][0] = 0; // t
        dataSet[i].yvtNumDat[1][0] = 0; // X0 (DOC C1)
        dataSet[i].yvtNumDat[2][0] = 0; // X1 (DOC C2)
        dataSet[i].yvtNumDat[3][0] = 0; // X2 (DOC C3)
        dataSet[i].yvtNumDat[4][0] = 0; // X3 (N/A)
        dataSet[i].yvtNumDat[5][0] = 0; // X4 (N/A)
        };
};

void setInitialConditions(int j){

    for(int i=0; i<numDataSet; i++){
        for(int k=0; k<numEqn+1; k++){
        sol[i][k] = dataSet[i].yvtNumDat[k][j];};
    };

    for(int i=0; i<numDataSet; i++){
        for(int k=0; k<numEqn+1; k++){
        sol2[i][k] = dataSet[i].yvtNumDat[k][j];};
    };
};

void RK4(){

    cout << "SOLVING ..." << endl;

    double hMAX;

    int trigger;

    int l=0;

    double step, step2, tinyStep;
    int numStep, numStep2, numInfStep, numInfStep2;
    int n=0,
        p=0;

    double k1[5],
           k2[5],
           k3[5],
           k4[5];

    double g1[5],
           g2[5],
           g3[5],
           g4[5];
```

```
presetInitialConditions();
for(int i=0;i<numEqn;i++){
  eqnControl[i]=i;
};

for(int j=0; j<numDataSet; j++){

        hMAX = 0.2;

    for(int k=0; k<(dataSet[j].numDat-1); k++){

        n=0;
        p=1;

    do{

        if(negTest(j)){

        setInitialConditions(k);
        setEqns(j, k);

        numStep = ceil((dataSet[j].yvtDat[0][k+1]-
                    dataSet[j].yvtDat[0][k])/(hMAX/(pow(2,n))));
        step = (dataSet[j].yvtDat[0][k+1]-dataSet[j].yvtDat[0][k])/numStep;

    for(int i=0;i<numStep;i++){

        for(int e=0; e<numEqn; e++){
            k1[e] = step*eqn(eqnControl[e], j, sol[j][0], sol[j][1],
                sol[j][2], sol[j][3],
                                        sol[j][4], sol[j][5]);};

        for(int e=0; e<numEqn; e++){
            k2[e] = step*eqn(eqnControl[e], j, sol[j][0]+step/2,
                sol[j][1]+k1[0]/2, sol[j][2]+k1[1]/2, sol[j][3]+k1[2]/2,
                                            sol[j][4]+k1[3]/2,
                                                sol[j][5]+k1[4]/2);};

        for(int e=0; e<numEqn; e++){
            k3[e] = step*eqn(eqnControl[e], j, sol[j][0]+step/2,
                sol[j][1]+k2[0]/2, sol[j][2]+k2[1]/2, sol[j][3]+k2[2]/2,
                                            sol[j][4]+k2[3]/2,
                                                sol[j][5]+k2[4]/2);};

        for(int e=0; e<numEqn; e++){
            k4[e] = step*eqn(eqnControl[e], j, sol[j][0]+step,
                sol[j][1]+k3[0], sol[j][2]+k3[1], sol[j][3]+k3[2],
```

```
                                                    sol[j][4]+k3[3],
                                                        sol[j][5]+k3[4]);};


  sol[j][0] = sol[j][0]+step;
  for(int e=1; e<numEqn+1; e++){
      sol[j][e] = (sol[j][e]+(k1[e-1]+2*k2[e-1]+2*k3[e-1]+k4[e-1])/6);};
      };

numStep2 =
ceil((dataSet[j].yvtDat[0][k+1]
    -dataSet[j].yvtDat[0][k])/(hMAX/(pow(2,n+1))));
step2 = (dataSet[j].yvtDat[0][k+1]-dataSet[j].yvtDat[0][k])/numStep2;

  for(int i=0;i<numStep2;i++){

      for(int e=0; e<numEqn; e++){
          g1[e] = step2*eqn(eqnControl[e], j, sol2[j][0], sol2[j][1],
              sol2[j][2], sol2[j][3],
                                          sol2[j][4], sol2[j][5]);};

      for(int e=0; e<numEqn; e++){
          g2[e] = step2*eqn(eqnControl[e], j, sol2[j][0]+step2/2,
              sol2[j][1]+g1[0]/2, sol2[j][2]+g1[1]/2, sol2[j][3]+g1[2]/2,
                                                sol2[j][4]+g1[3]/2,
                                                    sol2[j][5]+g1[4]/2);};

      for(int e=0; e<numEqn; e++){
          g3[e] = step2*eqn(eqnControl[e], j, sol2[j][0]+step2/2,
              sol2[j][1]+g2[0]/2, sol2[j][2]+g2[1]/2, sol2[j][3]+g2[2]/2,
                                                sol2[j][4]+g2[3]/2,
                                                    sol2[j][5]+g2[4]/2);};

      for(int e=0; e<numEqn; e++){
          g4[e] = step2*eqn(eqnControl[e], j, sol2[j][0]+step2,
              sol2[j][1]+g3[0], sol2[j][2]+g3[1], sol2[j][3]+g3[2],
                                              sol2[j][4]+g3[3],
                                                 sol2[j][5]+g3[4]);};


  sol2[j][0] = sol2[j][0]+step2;
  for(int e=1; e<numEqn+1; e++){
      sol2[j][e] = (sol2[j][e]+(g1[e-1]+2*g2[e-1]+2*g3[e-1]+g4[e-1])/6);
             };
      };

      } else{
```

```
    setEqns(j, k);
    setInitialConditions(k);

    numStep = ceil((dataSet[j].yvtDat[0][k+1]
                -dataSet[j].yvtDat[0][k])/(hMAX/(pow(2,n))));
    step = (dataSet[j].yvtDat[0][k+1]-dataSet[j].yvtDat[0][k])/numStep;

for(int i=0;i<numStep;i++){

    for(int e=0; e<numEqn; e++){
        k1[e] = step*eqn(eqnControl[e], j, sol[j][0], sol[j][1],
            sol[j][2], sol[j][3],
                                        sol[j][4], sol[j][5]);};

    for(int e=0; e<numEqn; e++){
        k2[e] = step*eqn(eqnControl[e], j, sol[j][0]+step/2,
            sol[j][1]+k1[0]/2, sol[j][2]+k1[1]/2, sol[j][3]+k1[2]/2,
                                            sol[j][4]+k1[3]/2,
                                                sol[j][5]+k1[4]/2);};

    for(int e=0; e<numEqn; e++){
        k3[e] = step*eqn(eqnControl[e], j, sol[j][0]+step/2,
            sol[j][1]+k2[0]/2, sol[j][2]+k2[1]/2, sol[j][3]+k2[2]/2,
                                            sol[j][4]+k2[3]/2,
                                                sol[j][5]+k2[4]/2);};

    for(int e=0; e<numEqn; e++){
        k4[e] = step*eqn(eqnControl[e], j, sol[j][0]+step,
            sol[j][1]+k3[0], sol[j][2]+k3[1], sol[j][3]+k3[2],
                                            sol[j][4]+k3[3],
                                                sol[j][5]+k3[4]);};

    trigger = 0;
    p=4;
    for(int e=1;e<numEqn+1;e++){
      if((sol[j][e]+(k1[e-1]+2*k2[e-1]+2*k3[e-1]+k4[e-1])/6)< -tol ||
          isnan(sol[j][e]+(k1[e-1]+2*k2[e-1]+2*k3[e-1]+k4[e-1])/6)){
            trigger = 1;};
    };

    if(trigger==1){

          for(int e=0;e<numEqn+1;e++){
          storeSol[j][e] = sol[j][e]; };

        do{

          for(int e=0;e<numEqn+1;e++){
```

```
      sol[j][e] = storeSol[j][e]; };

       tinyStep = step/pow(2,p);

for(int b=0;b<pow(2,p);b++){

for(int e=0; e<numEqn; e++){
    k1[e] = tinyStep*eqn(eqnControl[e], j, sol[j][0], sol[j][1],
        sol[j][2], sol[j][3],
                                    sol[j][4], sol[j][5]);};

for(int e=0; e<numEqn; e++){
    k2[e] = tinyStep*eqn(eqnControl[e], j, sol[j][0]+tinyStep/2,
        sol[j][1]+k1[0]/2, sol[j][2]+k1[1]/2, sol[j][3]+k1[2]/2,
                                        sol[j][4]+k1[3]/2,
                                            sol[j][5]+k1[4]/2);};

for(int e=0; e<numEqn; e++){
    k3[e] = tinyStep*eqn(eqnControl[e], j, sol[j][0]+tinyStep/2,
        sol[j][1]+k2[0]/2, sol[j][2]+k2[1]/2, sol[j][3]+k2[2]/2,
                                        sol[j][4]+k2[3]/2,
                                            sol[j][5]+k2[4]/2);};

for(int e=0; e<numEqn; e++){
    k4[e] = tinyStep*eqn(eqnControl[e], j, sol[j][0]+tinyStep,
        sol[j][1]+k3[0], sol[j][2]+k3[1], sol[j][3]+k3[2],
                                        sol[j][4]+k3[3],
                                            sol[j][5]+k3[4]);};

        sol[j][0] = sol[j][0]+tinyStep;
    for(int e=1; e<numEqn+1; e++){
        sol[j][e] =
            (sol[j][e]+(k1[e-1]+2*k2[e-1]+2*k3[e-1]+k4[e-1])/6);
        if(abs(sol[j][e])<=tol){sol[j][e]=0;};
        };
};

trigger=0;
for(int e=1;e<numEqn+1;e++){
  if((sol[j][e])<0){trigger = 1;};
};

p++;

    }while(trigger == 1);

for(int e=1;e<numEqn+1;e++){
    k1[e] = 0;
```

```
        k2[e] = 0;
        k3[e] = 0;
        k4[e] = 0;
    };

    sol[j][0]=storeSol[j][0];
    };

sol[j][0] = sol[j][0]+step;
for(int e=1; e<numEqn+1; e++){
    sol[j][e] = (sol[j][e]+(k1[e-1]+2*k2[e-1]+2*k3[e-1]+k4[e-1])/6);
    if(abs(sol[j][e])<=tol){sol[j][e]=0;};
    }; };

    numStep2 = ceil((dataSet[j].yvtDat[0][k+1]
                  -dataSet[j].yvtDat[0][k])/(hMAX/(pow(2,n+1))));
    step2 =
        (dataSet[j].yvtDat[0][k+1]-dataSet[j].yvtDat[0][k])/numStep2;


for(int i=0;i<numStep2;i++){

    for(int e=0; e<numEqn; e++){
        g1[e] = step2*eqn(eqnControl[e], j, sol2[j][0], sol2[j][1],
            sol2[j][2], sol2[j][3],
                                        sol2[j][4], sol2[j][5]);};

    for(int e=0; e<numEqn; e++){
        g2[e] = step2*eqn(eqnControl[e], j, sol2[j][0]+step2/2,
            sol2[j][1]+g1[0]/2, sol2[j][2]+g1[1]/2, sol2[j][3]+g1[2]/2,
                                              sol2[j][4]+g1[3]/2,
                                                 sol2[j][5]+g1[4]/2);};

    for(int e=0; e<numEqn; e++){
        g3[e] = step2*eqn(eqnControl[e], j, sol2[j][0]+step2/2,
            sol2[j][1]+g2[0]/2, sol2[j][2]+g2[1]/2, sol2[j][3]+g2[2]/2,
                                              sol2[j][4]+g2[3]/2,
                                                 sol2[j][5]+g2[4]/2);};

    for(int e=0; e<numEqn; e++){
        g4[e] = step2*eqn(eqnControl[e], j, sol2[j][0]+step2,
            sol2[j][1]+g3[0], sol2[j][2]+g3[1], sol2[j][3]+g3[2],
                                           sol2[j][4]+g3[3],
                                              sol2[j][5]+g3[4]);};


    trigger = 0;
    p=4;
```

```
for(int e=1;e<numEqn+1;e++){
  if((sol2[j][e]+(g1[e-1]+2*g2[e-1]+2*g3[e-1]+g4[e-1])/6)<0)
  {trigger = 1;};
};

if(trigger==1){

      for(int e=0;e<numEqn+1;e++){
      storeSol[j][e] = sol2[j][e]; };

    do{

      for(int e=0;e<numEqn+1;e++){
      sol2[j][e] = storeSol[j][e]; };

       tinyStep = step2/pow(2,p);

for(int b=0;b<pow(2,p);b++){

for(int e=0; e<numEqn; e++){
    g1[e] = tinyStep*eqn(eqnControl[e], j, sol2[j][0], sol2[j][1],
        sol2[j][2], sol2[j][3],
                                      sol2[j][4], sol2[j][5]);};

for(int e=0; e<numEqn; e++){
    g2[e] = tinyStep*eqn(eqnControl[e], j, sol2[j][0]+tinyStep/2,
        sol2[j][1]+g1[0]/2, sol2[j][2]+g1[1]/2, sol2[j][3]+g1[2]/2,
                                          sol2[j][4]+g1[3]/2,
                                              sol2[j][5]+g1[4]/2);};

for(int e=0; e<numEqn; e++){
    g3[e] = tinyStep*eqn(eqnControl[e], j, sol2[j][0]+tinyStep/2,
        sol2[j][1]+g2[0]/2, sol2[j][2]+g2[1]/2, sol2[j][3]+g2[2]/2,
                                          sol2[j][4]+g2[3]/2,
                                              sol2[j][5]+g2[4]/2);};

for(int e=0; e<numEqn; e++){
    g4[e] = tinyStep*eqn(eqnControl[e], j, sol2[j][0]+tinyStep,
        sol2[j][1]+g3[0], sol2[j][2]+g3[1], sol2[j][3]+g3[2],
                                       sol2[j][4]+g3[3],
                                          sol2[j][5]+g3[4]);};

        sol2[j][0] = sol2[j][0]+tinyStep;
    for(int e=1; e<numEqn+1; e++){
        sol2[j][e] =
            (sol2[j][e]+(g1[e-1]+2*g2[e-1]+2*g3[e-1]+g4[e-1])/6);
        if(abs(sol2[j][e])<=tol){sol2[j][e]=0;};
        };
```

```
    };

    trigger=0;
    for(int e=1;e<numEqn+1;e++){
      if((sol2[j][e])<0){trigger = 1;};
    };

    p++;

        }while(trigger == 1);

    for(int e=1;e<numEqn+1;e++){
        g1[e] = 0;
        g2[e] = 0;
        g3[e] = 0;
        g4[e] = 0;
    };

    sol2[j][0]=storeSol[j][0];
    };

sol2[j][0] = sol2[j][0]+step2;
for(int e=1; e<numEqn+1; e++){
    sol2[j][e] = (sol2[j][e]+(g1[e-1]+2*g2[e-1]+2*g3[e-1]+g4[e-1])/6);
    if(abs(sol2[j][e])<=tol){sol2[j][e]=0;};
           };
    };
      n++;

};

for(int e=1; e<numEqn+1; e++){
    if(abs(sol[j][e])<=tol){sol[j][e]=0;};
           };

for(int e=1; e<numEqn+1; e++){
    if(abs(sol2[j][e])<=tol){sol2[j][e]=0;};
           };
n++;

       if(n>=15){cout << "WARNING: SMALL STEP SIZE" << endl;};

}while(convgTest(j) || !negTest(j));

for(int m=0;m<numEqn+1;m++){
dataSet[j].yvtNumDat[m][k+1]=sol2[j][m];
};
```

```
    };
    };
};

bool convgTest(int j){

    double err[6];

    for(int i=1;i<numEqn+1;i++){

    if(sol2[j][i]>=1){
        err[i] = abs(2*(sol2[j][i]-sol[j][i])/(sol2[j][i]+sol[j][i]));}
    else{
        err[i] = abs(sol2[j][i]-sol[j][i]);};
                    };

     if( (err[1]>tol)|| isnan(err[1])||
         (err[2]>tol)|| isnan(err[2])||
         (err[3]>tol)|| isnan(err[3])||
         (err[4]>tol)|| isnan(err[4])||
         (err[5]>tol)|| isnan(err[5])){
           return 1;
       }else{
           return 0;};

};

bool negTest(int j){

    double negTol = 0;
    int test = 1;

    for(int i=1;i<6;i++){
        if(sol2[j][i]<negTol){test = 0;};
        if(isnan(sol2[j][i])){test = 0;};
    };

     if( test==0 ){

           return 0;
       }else{
           return 1;};
};

void POWELL(int p){

    double VARtester[3];
    double VARcontrol[2];
```

```
double VAR0, VAR1;
double step = 0.01;
int direction;
int k;
int r=1;
int loops =0;

int debugger;

setNumParamON();

resetBasis();

do{
    RK4();
    VARcontrol[0] = VAR();

    for(int i=0; i<numParam; i++){
        param0[i]=param[i];
    };

for(int j=0;j<numParamON;j++){

        step = tol*10;

RK4();
VARtester[1] = VAR();

    for(int i=0; i<numParam; i++){
        paramReturn[i]=param[i];
    };


        cout << endl;
        cout << "RUNNING DATASET: " << p << endl;
        cout << "CENTRE" << endl;
        printParamSample();


multiParam(j, step, 1);

RK4();
VARtester[2] = VAR();

        cout << endl;
        cout << "RUNNING DATASET: " << p << endl;
        cout << "+" << endl;
        printParamSample();
```

```
    for(int i=0; i<numParam; i++){
        param[i]=paramReturn[i];
    };

multiParam(j, step, -1);

RK4();
VARtester[0] = VAR();

        cout << endl;
        cout << "-" << endl;
        cout << "RUNNING DATASET: " << p << endl;
        printParamSample();



cout << "VARtesters" << endl << VARtester[0] << endl << VARtester[1] <<
    endl << VARtester[2] << endl << endl;

if((VARtester[2]>=VARtester[1])&&(VARtester[1]>VARtester[0])){
        direction = -1;}else{
if((VARtester[0]>=VARtester[1])&&(VARtester[1]>VARtester[2])){
        direction = 1;}else{
if((VARtester[1]>VARtester[0])&&(VARtester[1]>VARtester[2])){
        direction =
            (VARtester[0]-VARtester[2])/abs(VARtester[0]-VARtester[2]);
                cout << "WARNING: AT PEAK" << endl;
                }else{
              direction = 0;};};
    };

    cout << "->: " << direction << endl;

    for(int i=0; i<numParam; i++){
        param[i]=paramReturn[i];
    };

    RK4();
    cout << "ENDVAR: "; VAR();
    cout << "SMALLEST (run): " << smallVAR << endl;
    cout << "PARAM: " << j << " DIRECTION: " << direction << endl;

if(direction==1||direction==-1){

    multiParam(j, step, direction);
```

```
    VAR1 = VARtester[1];
    VAR0 = VARtester[1+direction];

int n=0,
    m=1,
    run=0,
    flip=0;

    while((VAR0<VAR1)||flip==1){
        VAR1 = VAR0;
        flip=0;

        multiParam(j, step, direction);

        RK4();
        VAR0 = VAR();

        cout << "Step: " << step << endl;
        cout << endl;
        cout << "RUNNING DATASET: " << p << endl;
        cout << "RUN" << endl;

        cout << "NEW VAR: " << VAR0 << endl << "OLD VAR: " << VAR1 << endl;

        if((n==m*5)&&(run==0)){
            step = 1.5*step;
            m++;};
            n++;
        if((VAR0>VAR1)&&step>0.000001){
            run = 1;
            direction = -direction;
            multiParam(j, step, direction);
            step = 0.5*step;
            VAR0=VAR1;
            flip = 1;
        };
    };

    for(int i=0; i<numParam; i++){
        param[i]=paramSmallest[i];
    };

};

printParameters(p);
};

    updateBasis();
```

```
        loops++;
        if(loops == r*numParamON){
            resetBasis();
            r++;
        };

        RK4();
        printSolutions();
        printParameters(p);
        VARcontrol[1] = VAR();


    if((VARcontrol[1]-smallVAR)>0){cout << "ABORT POWELL" << endl <<
        "Smallest: " << smallVAR << endl << "VAR: " << VARcontrol[1] << endl;
                                };


    }while((VARcontrol[0]>VARcontrol[1]) );


    cout << "FINAL SOL" << endl;
    RK4();
    cout << "FINAL VAR: " << VAR() << endl;

};

double VAR(){

    double var=0;
    double varSet=0;
    double Vd;
    int Np =0;
    int nd =0;

    switch(extVAR){

    case 0:

    for(int i=0;i<numDataSet;i++){

    switch(dataSet[i].X){
        case 1:
        Vd = param[111];
        break;
        case 2:
        Vd = param[112];
        break;
```

```
    case 3:
    Vd = param[113];
    break;
    case 4:
    Vd = param[114];
    break;
    case 5:
    Vd = param[115];
    break;
default:
    cout << "ERROR SELECTING V_d" << endl;
    cout << "Data Set: " << i << endl;
};

nd=0;
varSet = 0;
for(int k=0;k<dataSet[i].numDat;k++){
       if( (abs((Vd*dataSet[i].yvtDat[dataSet[i].X][k]
       +dataSet[i].yvtNumDat[dataSet[i].X][k])) > tol)&&
       (dataSet[i].yvtDat[dataSet[i].X][k]!= -1) ){
       varSet = varSet + pow((Vd*dataSet[i].yvtDat[dataSet[i].X][k
                     ]-dataSet[i].yvtNumDat[dataSet[i].X][k])/
                     (Vd*dataSet[i].yvtDat[dataSet[i].X][k]
                     +dataSet[i].yvtNumDat[dataSet[i].X][k]),2);
       nd++;
              };
   };

var = var + varSet*dataSet[i].numPat;


Np = Np + dataSet[i].numPat*nd;

};

if((var*4/Np)<smallVAR){
       smallVAR=var*4/Np;

       for(int i=0; i<numParam; i++){
           paramSmallest[i]=param[i];};
};

break;

case 1:

for(int i=0;i<numDataSet;i++){
```

```
switch(dataSet[i].X){
    case 1:
    Vd = param[111];
    break;
    case 2:
    Vd = param[112];
    break;
    case 3:
    Vd = param[113];
    break;
    case 4:
    Vd = param[114];
    break;
    case 5:
    Vd = param[115];
    break;
default:
    cout << "ERROR SELECTING V_d" << endl;
    cout << "Data Set: " << i << endl;
};

varSet = 0;
for(int k=0;k<dataSet[i].numDat;k++){
       if( (abs((Vd*dataSet[i].yvtDat[dataSet[i].X][k]
       +dataSet[i].yvtNumDat[dataSet[i].X][k])) > tol)&&
       (dataSet[i].yvtDat[dataSet[i].X][k]!= -1) ){
       varSet = varSet + pow((Vd*dataSet[i].yvtDat[dataSet[i].X][k]
                     -dataSet[i].yvtNumDat[dataSet[i].X][k])/
                     (Vd*dataSet[i].yvtDat[dataSet[i].X][k]
                     +dataSet[i].yvtNumDat[dataSet[i].X][k]),2);
              };
   };

var = var + varSet*dataSet[i].numPat*dataSet[i].numDatW;
Np = Np + dataSet[i].numPat*dataSet[i].numDatW*dataSet[i].numDat;


};

if((var*4/Np)<smallVAR){
       smallVAR=var*4/Np;

       for(int i=0; i<numParam; i++){
           paramSmallest[i]=param[i];};
};

break;
};
```

```
    cout << "COMPUTED VARIANCE: " << setprecision(9) << var*4/Np << endl <<
        "SMALLEST: " << smallVAR << endl;

    return var*4/Np;

};

void AIC(){

    double dp=0;
    double RSS=0,
           Vd=0;

    extVAR=0;

    aiccTest=0;
    aiccTestg=0;

    loadExtParameters(21);
    setNumParamON();


    loadDataSets(22);
    RK4();
    VARg=VAR();


    for(int i=0;i<numDataSet;i++){

           dp = dp + (dataSet[i].numDat-2)*(dataSet[i].numPat);
    };

    aicg = 2*(numParamON+1) + (dp)*log(VARg);
    aiccg = aicg + (2*(numParamON+1)*(numParamON+2))/(dp-numParamON);
    if(dp/numParamON<40){aiccTestg = 1;};

    loadDataSets(21);
    RK4();
    dp=0;
    RSS=0;

    for(int i=0;i<numDataSet;i++){

           dp = dp + (dataSet[i].numDat-2)*(dataSet[i].numPat);
    };

    aic = 2*(numParamON+1) + (dp)*log(VAR());
```

```
    aicc = aic + (2*(numParamON+1)*(numParamON+2))/(dp-numParamON);
    if(dp/numParamON<40){aiccTest = 1;};

    printParameters(21);

};

void printStats(){

    cout << "PRINTING STATISTICS ..." << endl;

    loadExtParameters(21);

        ofstream fout;

    fout.open("Statistics/Statistics.txt");
    fout << fixed << setprecision(5);
    fout << "SET" << '\t' << '\t' << "VAR" << '\r' << '\n';

    int j=0;

    for(int k=0; k<23; k++){
                    j=k;
           if(k==20){j=25;};
           if(k==21){j=26;};
           if(k==22){j=21;};

    loadDataSets(j);
    RK4();

    fout << j << '\t' << '\t' << VAR() << '\r' << '\n';
                     };

    fout.close();

};

void setNumParamON(){

    numParamON=0;

    for(int j=0; j<numParam;j++){
        if(ONOFF[j]==1){numParamON++;};
    };

};

void multiParam(int j, double step, int sign){
```

```
        for(int i=0;i<numParam;i++){
            param[i] = sign*step*powBasis[j][i] + param[i];
            if(param[i]<tol){param[i]=0;};
        };
};

void updateBasis(){

    double norm = 0;

    for(int i=(numParamON-1);i>0;i--){
        for(int j=0;j<numParam;j++){
            powBasis[i][j] = powBasis[i-1][j];
        };
    };


    double newVector[198]={0};
    double directionMatch[198]={0};
    double closestDirection = 0;
    int k=0;

    char debugger;

    for(int i=0;i<numParam;i++){
        newVector[i] = param[i]-param0[i];
        norm = norm + newVector[i]*newVector[i];
    };

    for(int i=0;i<numParam;i++){
        newVector[i] = newVector[i]/pow(norm,0.5);
    };

    for(int i=0;i<numParamON;i++){
        for(int j=0;j<numParam; j++){
            directionMatch[i] = directionMatch[i] +
                newVector[j]*powBasis[i][j];
        };
    };

    for(int i=0;i<numParamON;i++){
        if(directionMatch[i]>closestDirection){
            closestDirection = directionMatch[i];
            k = i;
        };
    };
```

```
    for(int i=0;i<numParam;i++){
        powBasis[k][i] = newVector[i];
    };

};

void resetBasis(){

    for(int i=0;i<numParamON;i++){
        for(int j=0;j<numParam;j++){
            powBasis[i][j]=0;
    };};

    int i=0;
    for(int j=0; j<numParam;j++){
        if(ONOFF[j]==1){
            powBasis[i][j]=1;
            i++;};
    };
};

void printBasis(){

    cout << endl << "BASE VECTORS" << endl <<endl;

    for(int i=0;i<numParamON;i++){
        cout << "{" << powBasis[i][0];
        for(int j=1;j<numParam;j++){
         cout << ", " << '\t' << powBasis[i][j];
         };
         cout << "}" << endl;
       };
       cout << endl;
};

void printParamSample(){

           cout << fixed << setprecision(7)
           << "K1: " << param[13] << endl
           << "Vd: " << param[115] << endl << endl;
};

void printSolutions(){

    cout << "PRINTING SOLUTIONS ..." << endl;

        ofstream fout;
```

```
    for(int j=0; j<numDataSet; j++){

    fout.open("Solutions/Eqn"+ to_string(j) + ".txt");

    fout << "T" << '\t' << '\t';

    for(int i=0;i<numEqn;i++){

     fout << "X" << i << '\t' << '\t';};
     fout << '\r' << '\n';

    for(int i=0;i<dataSet[j].numDat;i++){
        fout << fixed << setprecision(9);
    for(int k=0;k<numEqn+1;k++){
           fout << dataSet[j].yvtNumDat[k][i] << '\t';
           };
           fout << '\r' << '\n';
    };

    fout.close();};

    printConcentrations();

};

void printMetricSolutions(){

    cout << "PRINTING METRIC SOLUTIONS ..." << endl;
    int debugger;

       ofstream fout;

    for(int n=0;n<numDataSet;n++){


    fout.open("Data/Data_Metric/Set"+ to_string(n) + ".txt");


    fout << "Constant(micromol/(m^2min)):" << '\t' << infConst[n] << '\r' <<
        '\n'
         << "StartTime(min): " << '\t' << infOnT[n] << '\r' << '\n'
         << "InfLength(min): " << '\t' << infT[n] << '\r' << '\n';
    fout << "NumDataPoint: " << '\t' << '\t'<< dataSet[n].numDat << '\r' <<
        '\n';
    fout << "NumPatient: " << '\t' << '\t' << dataSet[n].numPat << '\r' <<
        '\n'
         << "DrugNum: " << '\t' << '\t' << dataSet[n].X ;
```

```
    fout << '\r' << '\n' << '\r' << '\n' << '\r' << '\n' << '\r' << '\n'
         << "Time(min)" << '\t' << "X" << dataSet[n].X << "(micromol/L)" <<
            '\r' << '\n';

        fout << 0 << '\t' << 0 << '\r' << endl;

    double Vd;
    switch(dataSet[n].X){
        case 1:
        Vd = param[111];
        break;
        case 2:
        Vd = param[112];
        break;
        case 3:
        Vd = param[113];
        break;
        case 4:
        Vd = param[114];
        break;
        case 5:
        Vd = param[115];
        break;
    default:
        cout << "ERROR SELECTING V_d" << endl;
        cout << "Data Set: " << n << endl;
    };

    for(int i=1;i<dataSet[n].numDat;i++){
        fout << dataSet[n].yvtNumDat[0][i] << '\t' <<
            dataSet[n].yvtNumDat[dataSet[n].X][i]/Vd << '\r' << '\n';
    };

    fout.close();
    };

};

void printConcentrations(){

    cout << "PRINTING CONCENTRATIONS ..." << endl;

    ofstream fout;

    for(int j=0; j<numDataSet; j++){

    fout.open("Concentrations/Eqn"+ to_string(j) + ".txt");
```

```
    fout << "T" << '\t' << '\t';

    for(int i=0;i<numEqn;i++){

     fout << "X" << i << '\t' << '\t';};
     fout << '\r' << '\n';

    for(int i=0;i<dataSet[j].numDat;i++){
        fout << fixed << setprecision(9);

            fout << dataSet[j].yvtNumDat[0][i] << '\t' <<
                   dataSet[j].yvtNumDat[1][i]/param[111] << '\t' <<
                   dataSet[j].yvtNumDat[2][i]/param[112] << '\t' <<
                   dataSet[j].yvtNumDat[3][i]/param[113] << '\t' <<
                   dataSet[j].yvtNumDat[4][i]/param[114] << '\t' <<
                   dataSet[j].yvtNumDat[5][i]/param[115] ;

            fout << '\r' << '\n';
    };

    fout.close();};

};

void printParameters(int j){

    cout << "PRINTING PARAMETERS ..." << endl;


    ofstream fout;

        fout.open("Parameters/Parameters_"+ to_string(j) +".txt");

        fout << "Param" << '\t' << "on/off" << '\t' << "Value" << '\r' <<
            '\n';

        for(int i=0; i<numParam; i++){
            fout << "a" << i << '\t' << ONOFF[i] << '\t' << setprecision(9) <<
                param[i] << '\r' << '\n';
        };

        fout << '\r' << '\n' << "VAR: " << '\t' << VAR();
        if(j==21){fout << '\r' << '\n' << "VARg: " << '\t' << VARg;
                  fout << '\r' << '\n' << "AIC: " << '\t' << aic;
                  fout << '\r' << '\n' << "AICc: " << '\t' << aicc << '\t' <<
                      aiccTest;
                  fout << '\r' << '\n' << "AICg: " << '\t' << aicg;
```

```
                fout << '\r' << '\n' << "AICcg: " << '\t' << aiccg << '\t'
                    << aiccTestg;
        };

        fout.close();
};

void generateMetricData(int j){

    string debugger;

    loadMetricDataSets();
    loadExtParameters(j);

    RK4();
    printConcentrations();

    printMetricSolutions();

};

void generateWolframSolutions(){

    char debugger;

    cout << "PRINTING CONCENTRATIONS for WOLFRAM ... " << endl;

    ofstream fout;

    loadSmoothDataSets();

    fout.open("Wolfram/Results.nb");

    for(int k=0; k<22; k++){
    loadExtParameters(k);
    RK4();

    for(int j=0; j<numDataSet; j++){

        fout << "numResult" << j << k << ":={";
        for(int i=0;i<dataSet[j].numDat-1;i++){
            fout << fixed << setprecision(9);

            fout << "{" << dataSet[j].yvtNumDat[0][i] << ", " <<
                dataSet[j].yvtNumDat[1][i]/param[111] << "}, ";
        };
            fout << "{" << dataSet[j].yvtNumDat[0][dataSet[j].numDat-1] << ",
                " << dataSet[j].yvtNumDat[1][dataSet[j].numDat-1]/param[111]
```

```
                << "}}";

            fout << '\r';
    };

    for(int j=0; j<numDataSet; j++){

        fout << "numResultP80" << j << k << ":={";
        for(int i=0;i<dataSet[j].numDat-1;i++){
            fout << fixed << setprecision(9);

            fout << "{" << dataSet[j].yvtNumDat[0][i] << ", " <<
                dataSet[j].yvtNumDat[5][i]/param[115] << "}, ";
        };
            fout << "{" << dataSet[j].yvtNumDat[0][dataSet[j].numDat-1] << ",
                " << dataSet[j].yvtNumDat[5][dataSet[j].numDat-1]/param[115]
                << "}}";

            fout << '\r';
    };

};
    fout.close();
};

void generateExtWolframSolutions(int k){

    char debugger;

    cout << "PRINTING CONCENTRATIONS for WOLFRAM ... " << endl;

    ofstream fout;

    loadSmoothDataSets();

    fout.open("Wolfram/Results_"+to_string(k)+".nb");

    loadExtParameters(k);
    RK4();

    for(int j=0; j<numDataSet; j++){

        fout << "numResult" << j << k << ":={";
        for(int i=0;i<dataSet[j].numDat-1;i++){
            fout << fixed << setprecision(9);

            fout << "{" << dataSet[j].yvtNumDat[0][i] << ", " <<
                dataSet[j].yvtNumDat[1][i]/param[111] << "}, ";
```

```
        };
            fout << "{" << dataSet[j].yvtNumDat[0][dataSet[j].numDat-1] << ",
                " << dataSet[j].yvtNumDat[1][dataSet[j].numDat-1]/param[111]
                << "}}";

            fout << '\r';
    };

    for(int j=0; j<numDataSet; j++){

        fout << "numResultP80" << j << k << ":={";
        for(int i=0;i<dataSet[j].numDat-1;i++){
            fout << fixed << setprecision(9);

            fout << "{" << dataSet[j].yvtNumDat[0][i] << ", " <<
                dataSet[j].yvtNumDat[5][i]/param[115] << "}, ";
        };
            fout << "{" << dataSet[j].yvtNumDat[0][dataSet[j].numDat-1] << ",
                " << dataSet[j].yvtNumDat[5][dataSet[j].numDat-1]/param[115]
                << "}}";

            fout << '\r';
    };

    fout.close();

};

void generateParamErr(int j){

        double VARup;
        double VARdown;

        double step = tol*5;
        int n;

        string debugger;

    loadExtParameters(j);
    if(j==100){
        loadDataSets(21);
    }else{
        loadDataSets(j);
    };

        RK4();

        VARup = 1.01*VAR();
```

```
for(int i=0; i<numParam; i++){
param0[i] = param[i];
paramErr[0][i] = param[i];
paramErr[1][i] = param[i];
};

for(int k=0; k<numParam; k++){

        step = tol*10;

    if(ONOFF[k]==1){

    do{
        n=0;
        do{

        param[k]=param[k]+step;
        RK4();

        n++;
        step=2*step;
        cout << "VAR UP^: " << VARup << endl;
       // cin >> debugger;

        }while(VAR()<VARup && n<100000);

        n=0;
        step=0.25*step;
        do{

        param[k]=param[k]-step;
        RK4();

        n++;
        step=0.5*step;
        cout << "VAR UPv: " << VARup << endl;

        }while(VAR()>VARup && n<100000);

      }while(abs(VARup-VAR())>0.0001);


        paramErr[0][k] = param[k];

        param[k]=param0[k];

        step = tol*5;
```

```
            do{
                n=0;
                do{

                param[k]=param[k]-step;
                if(param[k]<0){param[k]=0;};
                RK4();

                n++;
                step=2*step;
                }while(VAR()<VARup && n<100000 && param[k]>0);

                n=0;
                step=0.25*step;

                if(VAR()<VARup && param[k]==0){}else{
                do{

                param[k]=param[k]+step;
                RK4();

                n++;
                step=0.5*step;
                }while(VAR()>VARup && n<100000);};

              }while(abs(VARup-VAR())>0.0001 && param[k]!=0);

                paramErr[1][k] = param[k];

                param[k]=param0[k];

            };
        };
};

void printParameterErr(int j){

    cout << "PRINTING PARAMETERS ..." << endl;

    ofstream fout;

        fout.open("ParameterErr/Parameters_"+ to_string(j) +".txt");

        fout << "Param" << '\t' << "Value" << '\t' << '\t' << "(min)" << '\t'
            << '\t' << "(max)" << '\t' << '\t' << "(-)" << '\t' << '\t' <<
            "(+)" << '\r' << '\n';
```

```
        for(int i=0; i<numParam; i++){
            fout << "a" << i << '\t' << fixed << setprecision(7) << param[i]
                << '\t' << paramErr[1][i] << '\t' << paramErr[0][i] << '\t' <<
                param[i]-paramErr[1][i] << '\t' << paramErr[0][i]-param[i] <<
                '\r' << '\n';
        };

        fout << '\r' << '\n' << "VAR: " << '\t' << VAR();

        fout.close();
};

void METRICSgenerator(int k){

    cout << "PRINTING PARAM" << k << " METRICS..." << endl;

    int debugger;

    double aucCounter,    /// 0->INF
          aucCounter24,   /// 0->24
          aucCounter24post,  /// 0->24+intT
          aucCounteri24; /// infT->24

    int INFint=0;

    generateMetricData(k);
    loadMetricDataSets();

    cout << "#" << "; " << "AUC" << '\t' << '\t'<< "AUC24" << '\t' << "Cmax"
        << '\t' << "Thalf" << '\t' << "Chalf" << endl;

    /// AUC + C_MAX
    for(int j=0;j<numDataSet;j++){

        aucCounter=0;
        aucCounter24=0;
        aucCounter24post=0;
        INFint=0;

        for(int i=0;i<dataSet[j].numDat-1;i++){
            aucCounter = aucCounter +
                (dataSet[j].yvtDat[0][i+1]-dataSet[j].yvtDat[0][i])*
            (dataSet[j].yvtDat[dataSet[j].X][i+1]
            +dataSet[j].yvtDat[dataSet[j].X][i]);
        };

        for(int i=0;dataSet[j].yvtDat[0][i]<1440;i++){
```

```
        aucCounter24 = aucCounter24 +
            (dataSet[j].yvtDat[0][i+1]-dataSet[j].yvtDat[0][i])*
        (dataSet[j].yvtDat[dataSet[j].X][i+1]
        +dataSet[j].yvtDat[dataSet[j].X][i]);

        if(dataSet[j].yvtDat[0][i]==infT[j]){INFint=i;};

        if(Cmax[j]<dataSet[j].yvtDat[dataSet[j].X][i])
        {Cmax[j]=dataSet[j].yvtDat[dataSet[j].X][i];
         Tmax[j]=dataSet[j].yvtDat[0][i];};
        if(Cmax[j]<=dataSet[j].yvtDat[dataSet[j].X][i])
        {TmaxF[j]=dataSet[j].yvtDat[0][i];};
    };

    for(int i=0;dataSet[j].yvtDat[0][i]<(11520);i++){

        aucCounter24post = aucCounter24post + (dataSet[j].yvtDat[0][i+1]
        -dataSet[j].yvtDat[0][i])*(dataSet[j].yvtDat[dataSet[j].X][i+1]
        +dataSet[j].yvtDat[dataSet[j].X][i]);
    };

    AUC[j]=aucCounter*0.5;
    AUC24[j]=aucCounter24*0.5;
    AUC24post[j]=aucCounter24post*0.5;

/// DISTRIBUTION HALF LIFE

    int i=INFint;
    do{
        ThalfD[j]=dataSet[j].yvtDat[0][i];
        ChalfD[j]=dataSet[j].yvtDat[dataSet[j].X][i];
        i++;

    }while(Cmax[j]*0.5<=dataSet[j].yvtDat[dataSet[j].X][i]);

    if(abs(dataSet[j].yvtDat[dataSet[j].X][i]-Cmax[j]*0.5)<
    abs(dataSet[j].yvtDat[dataSet[j].X][i-1]-Cmax[j]*0.5)){
        ThalfD[j]=dataSet[j].yvtDat[0][i];
        ChalfD[j]=dataSet[j].yvtDat[dataSet[j].X][i];
    };

    cout << j << "; " << AUC[j] << '\t' << AUC24[j] << '\t' << Cmax[j] <<
        '\t' << ThalfD[j] << '\t' << ChalfD[j] << endl;
};

   ofstream fout;

fout.open("Metrics/Param_"+ to_string(k) + ".txt");
```

```
        fout << "INF #" << '\t' << "AUC" << '\t' << "AUC24" << '\t' <<
            "AUC24post" << '\t' << "Cmax" << '\t' << "TmaxI" << '\t' <<
            "TmaxF" << '\t' << "Thalf" << '\t' << "Chalf" << '\r' << '\n';

        fout << fixed << setprecision(4);

    for(int n=0;n<numDataSet;n++){

        fout << n << '\t' << AUC[n] << '\t' << AUC24[n] << '\t' <<
            AUC24post[n] << '\t' << Cmax[n] << '\t' << Tmax[n] << '\t' <<
            TmaxF[n] << '\t' << ThalfD[n] << '\t' << ChalfD[n] << '\r' << '\n';

    };
    fout.close();
};

//
// THE END
//
```