



Modelling kidney with compartment models from PET images

Maria Rantala

Master Thesis
February 2022

DEPARTMENT OF MATHEMATICS AND STATISTICS

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin OriginalityCheck service.

UNIVERSITY OF TURKU
Department of Mathematics and Statistics

RANTALA, MARIA: Modelling kidney with compartment models from PET images

Master Thesis, 52 pages, 9 appendix pages

Mathematics

February 2022

It is a well established procedure in medical research to analyse tracer activities from PET images and to model certain regions of the body with the help of these measured activities. In this thesis, the PET images are used to examine concentration activity of a tracer in kidney and the behaviour of the concentration activity is modelled with compartment models. One-tissue compartment model has been used before to examine kidney functionality. The focus of this study was to examine the goodness-of-fit of one-tissue compartment model compared to two-tissue compartment model with an additional compartment with irreversible trapping of the tracer. The compartment models were formulated mathematically with the help of Laplace transform and Cauchy's Residue Theorem.

The study had two secondary interests in addition to comparing the two compartment models. The first one was to determine the blood volume value for specific regions of interest in kidney. In short, the blood volume is a model correction that takes in consideration the blood component in the regions of interest. The other secondary interest in the study was delay, which was the time it took the tracer to move from aorta into the regions of interest in kidney. The goal was to see how delay behaved with varying blood volume values.

The study used the PET images of 9 healthy male volunteers. The concentration activities were measured from these images and then compared with the modelled concentration activity values. To determine the most suitable blood volume percentage, 21 fittings were done for both models with fixed blood volume percentages that ranged from 0% to 100% with 5% intervals. The fittings were done with programming language R using a package called `kinfitr`, which has been created for PET kinetic modelling by Granville Matheson. Slight modifications were made to the code concerning the blood volume value. The fitted values in the models were the K_1 , k_2 and k_3 values of the compartment models, the delay and the residual standard error of delay. AIC score and root mean square error were used as the goodness-of-fit measures for the models.

Contents

1	Introduction	1
2	Compartment models	2
2.1	Simple chemical reactions	2
2.2	Model for chemically inert diffusible tracer (1TCM)	3
2.3	Model for retained tracer (2TCM with irreversible factor)	4
2.3.1	Conventional 2TCM in series	4
2.3.2	Conventional 2TCM in parallel	5
2.3.3	The irreversible factor in 2TCM	6
3	Mathematical theory needed in model formulation	7
3.1	Convolution	7
3.2	Laplace transform and it's properties	7
3.3	Cauchy's Residue Theorem	12
4	Formulation of the models	15
4.1	1TCM	15
4.2	2TCM with irreversible factor	16
4.2.1	2TCM in series	16
4.2.2	2TCM in parallel	17
5	Positron emission tomography and kidney modelling	18
5.1	Kidney anatomy	18
5.2	Blood Volume	20
5.3	Kidney modelling	21
5.4	Patlak analysis	21
6	Simulation	24
6.1	kinfitr	24
7	Data Analysis	25
7.1	Subjects and procedure	25
7.2	Drawing ROIs with Carimas	25
7.3	AIC score	27
7.4	Errors	28
7.4.1	Root mean square error	28
7.4.2	Residual standard error	28
7.5	Analysis	28
7.6	Statistical Analysis	29
7.6.1	Shapiro-Wilk's test	29
7.6.2	Paired t-test	30
7.6.3	Paired Wilcoxon signed rank test	30
7.6.4	Kendall's τ	31
8	Results	32

9	Conclusions	51
10	Appendix	53

1 Introduction

Positron emission tomography (PET) is an imaging technique that uses radiopharmaceutical substances to quantitatively and non-invasively measure biochemical and physiological processes. PET imaging is mainly used to detect or measure changes in physiological activities like metabolism, blood flow or regional chemical composition and absorption. Major advantage of PET is that the imaging procedure can be repeatedly performed. The basics of PET imaging is that the technique detects pairs of gamma rays emitted indirectly by a positron-emitting radionuclide.

The object of this study was to model renal plasma flow with the help of PET imaging. Compartment models have been used before to determine plasma flow with one-tissue compartment model (1TCM), also known as two compartment model. In this study we aim to find out weather we can gain more accurate results by using two-tissue compartment model (2TCM) instead of the widely used and researched 1TCM. The specific model of 2TCM used in this study has a compartment with irreversible plasma flow, meaning that the second compartment is considered as a trapping mechanism.

In this study the activities of $H_2^{15}O$ tracer were measured from PET images of 9 healthy volunteers' kidneys. The activities represent the renal plasma flow which is the key interest of this study.

A kidney has a lot of blood vessels and capillaries so when studying kidney activities we have to take in consideration the activity of the blood. Since we're only interested in renal plasma flow, we have to correct the tissue activities with a blood volume correction after using the compartment models. The blood volume itself can be divided into arterial and venous fractions. The arterial blood volume can be determined from $C^{15}O$ images.[5] On other hand the venous fraction is proportionate to arterial blood volume.

Another interest in this study was to determine how the fitted values in the models behave with varying blood volume values. Accurate information on blood volume in kidneys has not yet been sufficiently studied. We tried to assess the proper blood volume value for our chosen regions of interest by studying model fittings with multiple blood volume percentages.

The activities from PET images have been measured with a program called Carimas which has been developed in Turku PET Centre. All the fittings and results shown in this study have been made using R package called kinfitr.[8]

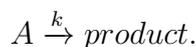
2 Compartment models

Compartment models are mathematical models in which a system consists of a fixed number of compartments that interact with each other. [12] The compartments are more of functional entities than physical areas. The compartments can have different functions and compositions but each compartment is assumed to be a homogeneous entity within which the entities being modelled are equivalent. If the modelled entities were not equivalent the compartment in question would be divided into two compartments. [10]

Compartment models assume that the concentration of a radiopharmaceutical substances is uniformly equal in a compartment. [12] Between these compartments there is either transition of matter or functional changes. Both of these changes can be modelled with simple chemical reactions which we go through in the subsection 2.1. In our case however we know that only transition of matter is happening between the compartments because the study uses a chemically inert and diffusible tracer, H_2^{15}O . So we don't have to take in consideration the possible reactions it would have in the body. H_2^{15}O is also good choice for a tracer because the tissue/blood partition coefficient of water is almost the same in each individual.

2.1 Simple chemical reactions

Let A be a set of molecules which are identical and independent of each other. Set A changes into product with speed k :



This simple reaction for one molecule is called elementary unimolecular reaction. [14] $A(t)$ can also be described as molecule concentration in time t . Now $\frac{dA}{dt} = -kA$ where k is the speed of the reaction. Let $A(0)$ be the molecule concentration at time $t = 0$. The differential equation can be separated:

$$\frac{1}{A}dA = -kdt$$

Let's solve $A(t)$ by integrating:

$$\int_{A(0)}^{A(t)} \frac{1}{A}dA = \int_0^t -kdt.$$

The integration yields

$$\ln A(t) - \ln A(0) = \ln \frac{A(t)}{A(0)} = -kt$$

which becomes

$$A(t) = A(0)e^{-kt}.$$

2.2 Model for chemically inert diffusible tracer (1TCM)

The simplest compartment model is the two-compartment model or otherwise called the one-tissue compartment model (1TCM) since the first compartment is for the arterial input function and the second for the tissue of interest. The following model description of 1TCM has been done by Inaba et al. [6].

PET scanners measure concentration of radioactivity (Bq/ml) as a function of time from the region of interest (ROI). Since human body is a complicated system and compartment model can only be used as a simplified model of it's parts we include in the model the concept of tissue fraction α [g/ml]:

$$\alpha = \frac{W}{V}, \quad (1)$$

where W [g] is the tissue mass in the ROI and V [ml] is the volume of the ROI. We assume a uniform and constant flow F [ml/min] and instantaneous equilibrium in the ROI.

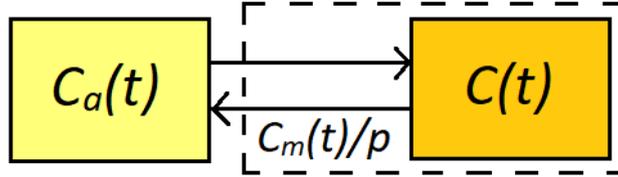


Figure 1: The activity concentrations in 1TCM. The yellow compartment is the arterial input function and the orange is the tissue compartment.

Figure 1 shows the concentration of radioactivity flowing into the ROI (arterial input function) as $C_a(t)$ [Bq/ml], the concentration inside the ROI as $C(t)$ [Bq/ml] and the concentration flowing out of the ROI as $C_m(t)/p$ [Bq/g] where $C_m(t)$ [Bq/ml] is the net tissue concentration in the ROI and p [ml/g] is the tissue/blood partition coefficient. The tracer balance in the ROI is then

$$\frac{dA(t)}{dt} = EFC_a(t) - \frac{EF}{p}C_m(t), \quad (2)$$

where $A(t)$ [Bq] is the total tracer amount in the ROI and E is the extraction fraction between plasma and tissue. $C_m(t)$ is not observable but it is related to the measured ROI concentration $C(t)$:

$$C(t) = \alpha C_m(t).$$

Equation (2) now becomes

$$\frac{dA(t)}{dt} = EFC_a(t) - \frac{EF}{p} \frac{C(t)}{\alpha}. \quad (3)$$

Since

$$\frac{A(t)[\text{Bq}]}{V[\text{ml}]} = C(t)[\text{Bq/ml}],$$

we can divide Equation (3) with volume of the ROI, V , and we get the differential equation of measured ROI concentration activity, $\frac{dC(t)}{dt}$:

$$\frac{dC(t)}{dt} = \frac{EF}{V}C_a(t) - \frac{EF}{p} \frac{C(t)}{W}. \quad (4)$$

Let f be the regional blood flow defined by

$$f = F/W. \quad (5)$$

Using Equations (1) and (5) we can reduce Equation (4) to a form where tissue fraction and regional blood flow can be seen:

$$\begin{aligned} \frac{dC(t)}{dt} &= \frac{EWf}{V}C_a(t) - \frac{EWf}{p} \frac{C(t)}{W}, \\ \frac{dC(t)}{dt} &= \alpha EfC_a(t) - \frac{Ef}{p}C(t). \end{aligned}$$

We note the coefficients αEf and $\frac{Ef}{p}$ as the speed coefficients K_1 and k_2 :

$$\frac{dC(t)}{dt} = K_1C_a(t) - k_2C(t). \quad (6)$$

This is the commonly used form of 1TCM differential equation. The tracer activity $C_a(t)$, which is the arterial input function, is carried to the tissue compartment with speed K_1 . The tracer concentration $C(t)$ flows out of the tissue compartment with speed k_2 . The tissue compartment is assumed to have instantaneous equilibrium.

2.3 Model for retained tracer (2TCM with irreversible factor)

2.3.1 Conventional 2TCM in series

Three-compartment model or two-tissue compartment model (2TCM) works exactly the same as 1TCM, except that another tissue compartment is added. First we introduce 2TCM with compartments in series. We denote the two tissue compartments as C_1 and C_2 . New values, k_3 and k_4 , are assigned as factors of concentration flow from first tissue compartment to the second and reversed respectively. (Figure 2) For the first compartment we get

$$\frac{dC_1(t)}{dt} = K_1C_a(t) - (k_2 + k_3)C_1(t) + k_4C_2(t) \quad (7)$$

and for the second we get

$$\frac{dC_2(t)}{dt} = k_3C_1(t) - k_4C_2(t). \quad (8)$$

The total tissue concentration is the sum of the two tissue compartments:

$$C_T = C_1(t) + C_2(t).$$

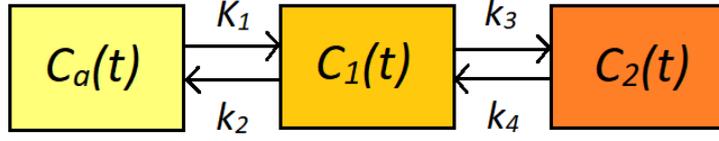


Figure 2: The activity concentrations in 2TCM with compartments in series. The yellow compartment is the arterial input function and the orange ones are tissue compartments.

2.3.2 Conventional 2TCM in parallel

In 2TCM, the compartments can also be arranged in parallel so that there are two parallel 1TCM where both are sharing the same arterial input function. (Figure 3) The corresponding differential equations for this model are

$$\frac{dC_1(t)}{dt} = K_{1a}C_a(t) - k_{2a}C_1(t)$$

and

$$\frac{dC_2(t)}{dt} = K_{1b}C_a(t) - k_{2b}C_2(t).$$

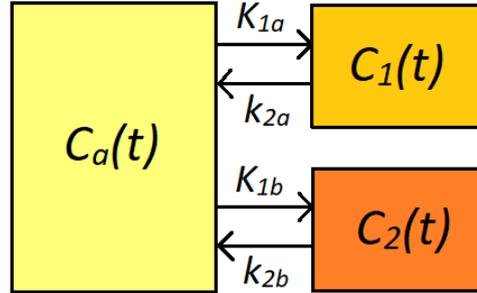


Figure 3: The activity concentrations with 2TCM in parallel. The yellow compartment is the arterial input function and the orange ones are tissue compartments.

The parameters of the model in series can be transformed to the parameters of the model in parallel and vice versa [10]:

$$\begin{aligned} K_{1a} &= \frac{K_1}{k_{2b} - k_{2a}}(k_3 + k_4 - k_{2a}), \\ k_{2a} &= \frac{k_2 + k_3 + k_4 - \sqrt{(k_2 + k_3 + k_4)^2 - 4k_2k_4}}{2}, \\ K_{1b} &= \frac{K_1}{k_{2b} - k_{2a}}(k_{2b} - k_3 - k_4), \\ k_{2b} &= \frac{k_2 + k_3 + k_4 + \sqrt{(k_2 + k_3 + k_4)^2 - 4k_2k_4}}{2}. \end{aligned} \tag{9}$$

In this case as well, the total tissue concentration is the sum of the two tissue compartments:

$$C_T = C_1(t) + C_2(t).$$

2.3.3 The irreversible factor in 2TCM

The conventional 2TCM in series can be transformed into a model with irreversible factor by assigning $k_4 = 0$. This means that the second compartment retains the tracer instead of having a reversed flow as in the conventional model depicted in Figure 2. Therefore Equations (7) and (8) become

$$\frac{dC_1(t)}{dt} = K_1 C_a(t) - (k_2 + k_3) C_1(t) \quad (10)$$

and

$$\frac{dC_2(t)}{dt} = k_3 C_1(t). \quad (11)$$

Since the parameters of the model with compartments in series can be transformed to the parameters of the model in parallel, we achieve the same effect with 2TCM in parallel by assigning $k_4 = 0$ and calculating the parameters K_{1a}, k_{1b} , K_{2b} and k_{2b} in Equations (9):

$$\begin{aligned} k_{2a} &= \frac{k_2 + k_3 - \sqrt{(k_2 + k_3)^2}}{2} = 0, \\ k_{2b} &= \frac{k_2 + k_3 + \sqrt{(k_2 + k_3)^2}}{2} = k_2 + k_3, \\ K_{1a} &= \frac{K_1}{k_{2b} - k_{2a}} (k_3 - k_{2a}) = \frac{K_1 k_3}{k_2 + k_3}, \\ K_{1b} &= \frac{K_1}{k_{2b} - k_{2a}} (k_{2b} - k_3) = \frac{K_1 k_2}{k_2 + k_3}. \end{aligned}$$

Thus the differential equations of the model become:

$$\frac{dC_1(t)}{dt} = K_{1a} C_a(t) - k_{2a} C_1(t) = K_{1a} C_a(t) \quad (12)$$

and

$$\frac{dC_2(t)}{dt} = K_{1b} C_a(t) - k_{2b} C_2(t). \quad (13)$$

3 Mathematical theory needed in model formulation

In this section we go through some mathematical definitions needed in formulating our models. The differential equations we get from the compartment models of 1TCM and 2TCM need to be solved in order to find out the final formula to calculate the concentration in the tissue compartments.

3.1 Convolution

Convolution is an operation on two functions which produces a new function expressing how the shape of one is modified by the other.

Definition 3.1 (Convolution). [4] Let f and g be two partially continuous functions between $[0, \infty]$. The convolution of these functions is

$$(f \otimes g)(t) = \int_0^t f(\tau)g(t - \tau)d\tau$$

and equivalently

$$(f \otimes g)(t) = \int_0^t f(t - \tau)g(\tau)d\tau.$$

The equivalence of convolutions is proved by changing the variables and letting $u = t - \tau$ thus getting the next result:

$$(f \otimes g)(t) = \int_0^t f(t - u)g(u)du.$$

Therefore

$$(f \otimes g)(t) = (g \otimes f)(t).$$

3.2 Laplace transform and it's properties

Convolution has useful properties concerning Laplace transform and we use these properties to solve the convolution function in the formulation of the model. Following definitions, theorems and lemmas have been derived from [4].

Definition 3.2 (Laplace transform). Let $f : [0, \infty) \rightarrow \mathbb{R}$ be locally integrable and let $s \in \mathbb{R}$. The *Laplace transform* of function f is

$$F(s) = \mathcal{L}\{f(t)\}(s) = \int_0^\infty f(t)e^{-st}dt.$$

A condition of exponential order is needed to ensure the existence of a Laplace transform.

Definition 3.3 (Exponential order). Function f is of exponential order $\alpha \in \mathbb{R}$ if there exist real constants $M > 0$ and $T > 0$ such that

$$|f(t)| \leq Me^{\alpha t} \text{ holds for all } t > T.$$

Theorem 1. *If $f : [0, \infty] \rightarrow \mathbb{R}$ is partially continuous and of exponential order $\alpha \in \mathbb{R}$, then $\mathcal{L}\{f\}(s)$ exists for all $s > \alpha$.*

Proof. We want to prove that integral

$$\int_0^{\infty} e^{-st} f(t) dt \quad (14)$$

converges with all $s > \alpha$. For function f

$$\begin{aligned} \left| \int_0^R e^{-st} f(t) dt \right| &\leq \int_0^R e^{-st} |f(t)| dt \leq \int_0^R e^{-st} M e^{\alpha t} dt \\ &= M \int_0^R e^{-(s-\alpha)t} dt = \frac{M}{s-\alpha} (1 - e^{-(s-\alpha)R}) \end{aligned}$$

holds for all $s > \alpha$ because the exponent is negative. When $R \rightarrow \infty$, then

$$\int_0^{\infty} e^{-st} |f(t)| dt \leq \frac{M}{s-\alpha}$$

where $s > \alpha$. Therefore, the integral in Equation (14) is absolutely convergent for all $s > \alpha$ and there exists a Laplace transform $\mathcal{L}\{f\}(s)$ for function f with all $s > \alpha$. \square

Heaviside step function allows us to define bounds for integration.

Definition 3.4 (Heaviside step function). The Heaviside step function is

$$H(x) = \begin{cases} 0, & x < 0 \\ 1, & x > 0 \end{cases} .$$

This Heaviside step function is not defined when $x = 0$.

Lemma 1. *Let F be the Laplace transform of function f and let it be defined with $s > a \geq 0$. Then*

$$\mathcal{L}\{H(t-a)f(t-a)\}(s) = e^{-as}F(s),$$

where $t \in \mathbb{R}$ and $H(t-a)$ is the Heaviside step function at point $t-a$.

Proof. From the definition of Laplace transform we get

$$\mathcal{L}\{H(t-a)f(t-a)\}(s) = \int_0^{\infty} e^{-st} H(t-a) f(t-a) dt = \int_a^{\infty} e^{-st} f(t-a) dt,$$

where the last equation follows from the definition of Heaviside step function: $H(t-a) = 0$ when $t < a$ and $H(t-a) = 1$ when $t > a$. Next we denote $v = t-a$ and $dv = dt$ and get

$$\int_a^{\infty} e^{-st} f(t-a) dt = \int_0^{\infty} e^{-as} e^{-sv} f(v) dv = e^{-as} \int_0^{\infty} e^{-sv} f(v) dv = e^{-as} F(s).$$

\square

Lemma 2 (Convolution property of Laplace transform). *Let f and g be two partially continuous functions in $[0, \infty)$ and let their exponential order be α . The Laplace transform of convolution of f and g is*

$$\mathcal{L}\{f(t) \otimes g(t)\}(s) = F(s)G(s),$$

where $F(s) = \mathcal{L}\{f(t)\}(s)$ and $G(s) = \mathcal{L}\{g(t)\}(s)$.

Proof. The Laplace transform of convolution of functions f and g is

$$\begin{aligned} \mathcal{L}\{f \otimes g\} &= \int_0^{\infty} e^{-st}(f \otimes g)(t)dt \\ &= \int_0^{\infty} e^{-st} \int_0^t f(t - \tau)g(\tau)d\tau dt. \end{aligned}$$

Here the Heaviside step function can be used to make the integration bounds same for both integrals:

$$\mathcal{L}\{f \otimes g\} = \int_0^{\infty} e^{-st} \int_0^{\infty} H(t - \tau)f(t - \tau)g(\tau)d\tau dt.$$

According to Lemma 1

$$\mathcal{L}\{f(t - \tau)H(t - \tau)\}(s) = \int_0^{\infty} e^{-st}H(t - \tau)f(t - \tau)dt = e^{-\tau s}F(s),$$

so

$$\mathcal{L}\{f \otimes g\} = \int_0^{\infty} g(v)e^{-sv}F(s)dv = F(s) \int_0^{\infty} e^{-s\tau}g(\tau)d\tau = F(s)G(s).$$

□

In the formulation of the model we take a Laplace transform of a first order differential equation so we need to know what becomes of the Laplace transform of a differential equation. Turns out that the differential equation becomes an easily solvable algebraic equation.

Theorem 2 (Laplace transform of a derivative function). *If both f and f' are partially continuous in $[0, \infty]$ and exponential order α , then the Laplace transform of a derivative function is*

$$\mathcal{L}\left\{\frac{df(t)}{dt}\right\} = sF(s) - f(0),$$

where F is the Laplace transform of function f .

Proof. The Laplace transform for derivative function $\frac{df(t)}{dt}$ is

$$\mathcal{L}\left\{\frac{df(t)}{dt}\right\} = \int_0^{\infty} \frac{df(t)}{dt}e^{-st}dt.$$

Let $u = e^{-st}$ and $dv = \frac{df(t)}{dt} dt$. By calculating the integration by parts we get

$$\int_0^{\infty} u \cdot dv = \left|_0^{\infty} uv - \int_0^{\infty} v \cdot du,\right.$$

where $du = -se^{-st}$ and $v = f(t)dt$. The result is

$$\begin{aligned} \int_0^{\infty} \frac{df(t)}{dt} e^{-st} &= \lim_{b \rightarrow \infty} \left|_0^b e^{-st} f(t) - \int_0^{\infty} f(t)(-s)e^{-st} dt \right. \\ &= \lim_{b \rightarrow \infty} e^{-sb} f(b) - e^0 f(0) + s \int_0^{\infty} f(t)e^{-st} dt \\ &= 0 - f(0) + sF(s) \\ &= sF(s) - f(0). \end{aligned}$$

□

While solving the differential equation in the formulation of the models we make use of the linearity property of the Laplace transform.

Theorem 3 (Linearity of Laplace transform). *Let $f, g : [0, \infty) \rightarrow \mathbb{R}$ be functions which have Laplace transformations with parameter $s \in \mathbb{R}$. Now \mathcal{L} is linear and*

$$\mathcal{L}[f + g](s) = \mathcal{L}f(s) + \mathcal{L}g(s).$$

Proof. We know that an integral has a property of linearity and we have assumed that f and g are integrable over $[0, \infty)$ so that

$$\begin{aligned} \mathcal{L}\{[f + g](s)\} &= \int_0^{\infty} e^{-st}(f(t) + g(t))dt \\ &= \int_0^{\infty} e^{-st}f(t)dt + \int_0^{\infty} e^{-st}g(t)dt \\ &= \mathcal{L}\{f(s)\} + \mathcal{L}\{g(s)\}. \end{aligned}$$

□

In order to prove the theorem for *the inverse Laplace transform* we need to define *complex Laplace transform*, *The Fourier transform* and *the inverse Fourier transform*.

Definition 3.5 (Complex Laplace transform). Let $f : [0, \infty) \rightarrow \mathbb{R}$. The *complex Laplace transform* of function is

$$F(s) = \mathcal{L}\{f(t)\}(s) = \int_0^{\infty} f(t)e^{-st} dt,$$

where $s = \gamma + i\omega$ is a complex number with real numbers γ and ω .

The Fourier transform and the inverse Fourier transform are defined here for the need to prove other theorems. They will not be discussed any more deeply. Further proofs can be found in [9].

Definition 3.6 (Fourier transform). Let function $U(t)$ and its derivative $U'(t)$ be continuous in parts and let U be integrable over \mathbb{R} . Then the *Fourier transform* of function $U(t)$ is

$$F(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} U(t)e^{-i\omega t} dt,$$

where $\omega \in \mathbb{R}$ is called the frequency and its units are radians per unit time.

Definition 3.7 (Inverse Fourier transform). The *Inverse Fourier transform* of Fourier transform $F(\omega)$ is

$$U(t) = \int_{-\infty}^{\infty} F(\omega)e^{i\omega t} d\omega.$$

In the formulation of the models it's necessary to find out the function from which the known Laplace Transformed function has been derived from and for that we need the *inverse Laplace Transform*. We define it as so called Bromwich integral in order to prove the linear property of the inverse Laplace transform[4].

Theorem 4 (Inverse Laplace transform). *The inverse Laplace transform is*

$$f(t) = \mathcal{L}^{-1}\{F(s)\}(t) = \frac{1}{2\pi i} \lim_{T \rightarrow \infty} \int_{\gamma-iT}^{\gamma+iT} e^{st}F(s)ds,$$

where F is the complex Laplace transform of function f and γ is the real part of complex variable s .

Proof. Let function U be defined as followed:

$$U(t) = \begin{cases} f(t), & t \geq 0 \\ 0, & t < 0 \end{cases}.$$

Let's apply Fourier transform and inverse Fourier transform:

$$U(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} U(t)e^{-i\omega t} dt \right] e^{i\omega t} d\omega.$$

The integrable function U becomes 0 when $t < 0$ so the previous equation becomes

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\int_0^{\infty} f(t)e^{-i\omega t} dt \right] e^{i\omega t} d\omega.$$

We can change variables with $s = \gamma + i\omega$ where the real part γ is fixed and $d\omega = \frac{ds}{i}$. The new boundaries for integration are $s = \gamma - i\omega$, $s = \gamma + i\omega$ and the integral is

$$f(t) = \frac{1}{2\pi i} \int_{\gamma-i\infty}^{\gamma+i\infty} \left[\int_0^{\infty} f(t)e^{-st} dt \right] e^{st} ds$$

which becomes

$$f(t) = \mathcal{L}^{-1}\{F(s)\}(t) = \frac{1}{2\pi i} \int_{\gamma-i\infty}^{\gamma+i\infty} F(s)e^{st} ds.$$

□

Note that the relation of Laplace transform and inverse Laplace transform can be seen in Lemma 1:

$$\mathcal{L}\{H(t-a)f(t-a)\}(s) = e^{-as}F(s),$$

where F is the Laplace transform of function f , H is the Heaviside step function, $t \in \mathbb{R}$ and $s > a \geq 0$. This can be reversed into a form with inverse Laplace transform [2]:

$$\mathcal{L}^{-1}\{F(s-a)\}(t) = e^{at}f(t),$$

where f is the inverse Laplace transform of function F .

3.3 Cauchy's Residue Theorem

In order to solve an integral of inverse Laplace transform we can use Cauchy's Residue Theorem, which is useful in evaluating contour integrals over a simple closed contour C where the integrand has the form $f(z)/(z-z_0)^k$ and function f is an analytic function. In the case of our model formulation, the integral of inverse Laplace transform has an essential singularity or a pole at $z_0 \in \mathbb{C}$. The definitions and theorems in this subsection can be found from [9].

Definition 3.8 (A punctured disk). The punctured disk of radius $\varepsilon > 0$ centered at the point z_0 is denoted by

$$D_\varepsilon^*(z_0) = \{z : 0 < |z - z_0| < \varepsilon\}.$$

Definition 3.9 (Analytic function). When a complex function f can be represented by a Taylor series at z_0 then it is differentiable in some neighbourhood of z_0 . The function f is said to be analytic at z_0 if its derivative exists at each point z in some neighbourhood of z_0 . If f is analytic at each point in the region R , then we say that f is analytic on R .

Here we make use of Laurent's Theorem to introduce and define the *residue* of a function, which will be needed in the Cauchy's Residue Theorem.

Theorem 5 (Laurent's Theorem). *Let $0 \leq r < R$, and let function f be analytic in the annulus $A = A(r, R, \alpha) = \{r < |z - \alpha| < R\}$. If ρ is any number such that $r < \rho < R$, then for all $z_0 \in A$, f has the Laurent series representation*

$$f(z_0) = \sum_{-\infty}^{\infty} c_n(z_0 - \alpha)^n = \sum_1^{\infty} c_{-n}(z_0 - \alpha)^{-n} + \sum_0^{\infty} c_n(z_0 - \alpha)^n, \quad (15)$$

where for $n = 1, 2, \dots$, the coefficients c_{-n} and c_n are given by

$$c_{-n} = \frac{1}{2\pi i} \int_{C_\rho^+\alpha} \frac{f(z)dz}{(z - \alpha)^{-n+1}}$$

and

$$c_n = \frac{1}{2\pi i} \int_{C_\rho^+\alpha} \frac{f(z)dz}{(z - \alpha)^{n+1}}.$$

Proof. [9, p. 226-228] □

The point α is called an isolated singularity of a complex function f if f is not analytic at α , but there exists a real number $R > 0$ such that f is analytic everywhere in the punctured disk $D_R^*(\alpha)$. Functions with isolated singularities have a Laurent series, since the punctured disk $D_R^*(\alpha)$ is the same as the annulus $A(0, R, \alpha)$.

Definition 3.10 (Residue). Let function f have a nonremovable isolated singularity at the point z_0 . Now f can be presented as a Laurent series in disk $D_R^*(z_0)$ for all $z \in D_R^*(z_0)$. Therefore

$$f(z) = \sum_{-\infty}^{\infty} a_n(z - z_0)^n,$$

where coefficient a_{-1} is called the *residue* of the function f at the point z_0 . Residue is noted as

$$\text{Res}[f, z_0] = a_{-1}.$$

Let function f be analytic in disk $D_R^*(z_0)$. According to the Laurent's Theorem, the Laurent series coefficients in Equation (15) are given by

$$a_n = \frac{1}{2\pi i} \int_{C_r^+(z_0)} \frac{f(\xi)d\xi}{(\xi - z_0)^{n+1}}, n = 0, \pm 1, \pm 2, \dots$$

where $C_r^+(z_0) = \{z : |z - z_0| = r\}$ is any positively oriented circle with $r < R$. If we set $n = -1$ in Equation (16), then we obtain

$$\int_C f(\xi)d\xi = 2\pi i a_{-1} = 2\pi i \text{Res}[f, z_0], \tag{16}$$

where z_0 is the only singularity of f that lies inside C .

The Cauchy-Goursat theorem states that within certain domains the integral of an analytic function over a simple closed contour is zero. The extended Cauchy-Goursat theorem is used in proving Cauchy's Residue Theorem, but it will not be proven here.

Theorem 6 (Extended Cauchy-Goursat Theorem). *Let C, C_1, C_2, \dots, C_n be simple closed positively oriented contours with the property that C_k lies interior to C for $k = 1, 2, \dots, n$ and the set interior to C_k has no points in common with the set interior to C_j if $k \neq j$. Let f be analytic on domain D that contains all the contours and the region between C and $C_1 + C_2 + \dots + C_n$ and let $z \in D$. Then*

$$\int_C f(z)dz = \sum_{k=1}^n \int_{C_k} f(z)dz.$$

Proof. [9, p. 175-186] □

Theorem 7 (Cauchy's Residue Theorem). *Let D be a simply connected domain, and let C be a simple closed positively oriented contour that lies in D . If function f is analytic inside C and on C , except at the points z_1, z_2, \dots, z_n that lie inside C , then*

$$\int_C f(z)dz = 2\pi i \sum_{k=1}^n \text{Res}[f, z_k]$$

Proof. Since there are a finite number of singular points inside C , there exists an $r > 0$ such that the positively oriented circles $C_k = C_r^+(z_k)$, for $k = 1, 2, \dots, n$, are mutually disjoint and all lie inside C . Using the extended Cauchy-Goursat theorem, it follows that

$$\int_C f(z)dz = \sum_{k=1}^n \int_{C_k} f(z)dz.$$

Since f is analytic in a punctured disk with center z_k that contains the circle C_k , Equation (16) can be used to obtain

$$\int_{C_k} f(z)dz = 2\pi i \text{Res}[f, z_k],$$

for $k = 1, 2, \dots, n$. Using this and the previous equation results in

$$\int_C f(z)dz = 2\pi i \sum_{k=1}^n \text{Res}[f, z_k].$$

□

A *pole* is a singularity of a complex function. This means that in the point of the pole the function in question is not defined or the function stops being well-behaved in some way.

Definition 3.11 (Pole). Let f have an isolated singularity at α with Laurent series expansion

$$f(z) = \sum_{-\infty}^{\infty} c_n(z - \alpha)^n$$

for all $z \in A(0, R, \alpha)$. If k is a positive integer such that $c_{-k} \neq 0$ and $c_n = 0$ $n = -k - 1, -k - 2, -k - 3, \dots$, then we say that f has a pole of order k at α .

Theorem 8 (Residue at pole). *Let function f have a pole of order 1 in point z_0 , then*

$$\text{Res}[f, z_0] = \lim_{z \rightarrow z_0} (z - z_0)f(z).$$

Proof. If function f has a pole of order 1 in z_0 , then according to Definition 3.11

$$f(z) = \frac{a_{-1}}{z - z_0} + a_0 + a_1(z - z_0) + a_2(z - z_0)^2 + \dots$$

By multiplying both sides of the equation with $(z - z_0)$ and by taking the limit $z \rightarrow z_0$ we get

$$\begin{aligned} \lim_{z \rightarrow z_0} (z - z_0)f(z) &= \lim_{z \rightarrow z_0} [a_{-1} + a_0(z - z_0) + a_1(z - z_0)^2 + \dots] \\ &= a_{-1} \\ &= \text{Res}[f, z_0]. \end{aligned}$$

□

4 Formulation of the models

Now we move to the formulation of 1TCM and 2TCM by solving their differential equations presented in Section 2. The mathematical theory in Section 3 will help in solving the model equations into a formulas which depict the tissue concentration activity. These activities will be the single tissue compartment activity, $C(t)$, for 1TCM and the total activity of two tissue compartments, $C_T(t)$, for 2TCM.

4.1 1TCM

We start to solve Equation (6), which was the differential equation for 1TCM concentrations:

$$\frac{dC(t)}{dt} = K_1 C_a(t) - k_2 C(t).$$

First we move the terms that have $C(t)$ to the left side of the equation and then letting $C(t) = y(t)$ and $K_1 C_a(t) = f(t)$:

$$y'(t) + k_2 y(t) = f(t).$$

According to Theorem 1, $\mathcal{L}\{y'(t)\} = sY(s) - y(0)$ where in this case $y(0) = C(0) = 0$. With Theorem 2 we can make the differential equation into its Laplace transform:

$$sY(s) + k_2 Y(s) = F(s).$$

Solving $Y(s)$ gives us

$$\begin{aligned} (s + k_2)Y(s) &= F(s) \\ Y(s) &= \frac{1}{s + k_2} F(s) \\ Y(s) &= G(s)F(s). \end{aligned}$$

We notice that the formula follows the convolution property of Laplace transform, so according to Lemma 1

$$y(t) = (f \otimes g) = (g \otimes f). \quad (17)$$

We need to calculate the unknown $g(t)$ to reduce the equation. Definition 2.4 says that

$$g(t) = \mathcal{L}^{-1}\{G(s)\}(t) = \frac{1}{2\pi i} \lim_{T \rightarrow \infty} \int_{\gamma - iT}^{\gamma + iT} \frac{1}{s + k_2} e^{st} ds.$$

Cauchy's Residue theorem tells us that

$$\begin{aligned} g(t) &= \frac{1}{2\pi i} \lim_{T \rightarrow \infty} \int_{\gamma - iT}^{\gamma + iT} \frac{1}{s + k_2} e^{st} ds \\ &= \frac{1}{2\pi i} \left(2\pi i \sum_{k=1}^n \text{Res} \left[\frac{1}{s + k_2} e^{st}, s_k \right] \right) \\ &= \sum_{k=1}^n \text{Res} \left[\frac{1}{s + k_2} e^{st}, s_k \right]. \end{aligned}$$

We can see that there is only one point in which $G(s)$ is not defined and that is $s_0 = -k_2$. Now we can use Theorem 5 to help us calculate the residue:

$$\begin{aligned}
g(t) &= \text{Res} \left[\frac{1}{s + k_2} e^{st}, s_0 \right] \\
&= \lim_{s \rightarrow s_0} (s - s_0) \frac{1}{s + k_2} e^{st} \\
&= \lim_{s \rightarrow -k_2} (s + k_2) \left(\frac{1}{s + k_2} e^{st} \right) \\
&= \lim_{s \rightarrow -k_2} e^{st} \\
&= e^{-k_2 t}.
\end{aligned}$$

The final formula for concentration of activity $C(t)$ comes from changing back the original functions of the model:

$$y(t) = (f \otimes g)$$

$$C(t) = K_1 C_a(t) \otimes e^{-k_2 t}. \quad (18)$$

4.2 2TCM with irreversible factor

4.2.1 2TCM in series

Solving 2TCM in series uses the same procedure with Laplace transform as in solving 1TCM. Therefore the same procedure won't be done here. Instead we present the final formula for conventional 2TCM with compartments in series with the irreversible factor already taken into account. The differential equations of 2TCM in series with irreversible factor ($k_4 = 0$) were Equation (10) and Equation (11):

$$\begin{aligned}
\frac{dC_1(t)}{dt} &= K_1 C_a(t) - (k_2 + k_3) C_1(t), \\
\frac{dC_2(t)}{dt} &= k_3 C_1(t).
\end{aligned}$$

The solutions for these differential equations are

$$\begin{aligned}
C_1(t) &= K_1 C_a(t) \otimes e^{-(k_2+k_3)t}, \\
C_2(t) &= k_3 C_1(t) \otimes 1.
\end{aligned}$$

Thus the total concentration is

$$\begin{aligned}
C_T(t) &= C_1(t) + C_2(t) \\
&= K_1 C_a(t) \otimes e^{-(k_2+k_3)t} + k_3 C_1(t) \otimes 1 \\
&= \int_0^t K_1 C_a(\tau) e^{-(k_2+k_3)(t-\tau)} d\tau + \int_0^t k_3 C_a(\tau) d\tau.
\end{aligned}$$

4.2.2 2TCM in parallel

The differential equations for 2TCM in parallel with irreversible factor ($k_4 = 0$) were Equation (12) and Equation (13):

$$\begin{aligned}\frac{dC_1(t)}{dt} &= K_{1a}C_a(t), \\ \frac{dC_2(t)}{dt} &= K_{1b}C_a(t) - k_{2b}C_2(t).\end{aligned}$$

The solutions to these equations are

$$\begin{aligned}C_1(t) &= K_{1a}C_a(t) \otimes 1, \\ C_2(t) &= K_{1b}C_a(t) \otimes e^{-k_{2b}t}.\end{aligned}$$

The total concentration is then

$$\begin{aligned}C_T(t) &= C_1(t) + C_2(t) \\ &= K_{1a}C_a(t) \otimes 1 + K_{1b}C_a(t) \otimes e^{-k_{2b}t} \\ &= \int_0^t K_{1a}C_a(\tau)d\tau + \int_0^t K_{1b}C_a(\tau)e^{-k_{2b}(t-\tau)}d\tau.\end{aligned}$$

5 Positron emission tomography and kidney modelling

PET imaging is categorized as a type of nuclear medicine since it uses radioactive substances. Physiologically active compounds are tagged with positron-emitting radionuclides, such as isotopes of oxygen, nitrogen, carbon or fluorine, while maintaining their biological properties. [10] The labelled compounds are administered intravenously in tracer amounts. PET imaging has a unique molecular sensitivity as it can use very small amount of tracer in the range of nano- to picomolar concentration without disturbing the normal biochemistry of the target tissue. [7]

Radioactive atoms are unstable so they spontaneously disintegrate into other atoms. In this process, energy is released by the emission of ionizing radiation. When a positron-emitting radionuclide decays, the decay energy is shared between positron and neutrino. The positron can travel a short distance in tissue before losing enough energy to be able to interact with an electron. This leads to the annihilation of the positron and the electron. Two 511 keV gamma photons arise in the process, travelling in approximately opposite direction from the site of the annihilation. The emitted gamma radiation makes it possible to quantify and locate the radioactive material.

The result of radioactivity measurement is the number of detected events (counts) per measurement time. This is much lower than the number of actual disintegrations during that time. Detection efficiency is different for each detector system (and changes over time, is affected by humidity and temperature, etc). Therefore all radioactivity measurement devices (for example PET scanners, well counters and dose calibrators) need to be cross-calibrated using a sample with known radioactivity. Cross-calibrations provide us with calibration coefficient for each device, which are used to convert the device-dependent values of counts per second to comparable values of Bq.

The standard unit of radioactivity is Becquerel, and it is equal to one decay per second. The "activities" that we actually are using in PET data analyses are concentrations of radioactivity: Bq/ml or Bq/g.

Dynamic imaging is a type of data acquisition available in PET. It can capture the metabolic information on a real time fashion, which enables one to derive valuable physiologic quantitative parameters. Dynamic PET imaging has been extensively used in a number of potential applications in neurology, cardiology and oncology. [7] In this study we utilize dynamic PET images to get absolute measurements of tracer pharmacokinetics using kinetic modelling analysis.

5.1 Kidney anatomy

While studying renal plasma flow, a few things are needed to be understood about kidney anatomy. The theory and the images about kidney anatomy have been derived from Nicola Thomas's book *Renal Nursing* [13].

The kidneys are paired organs located on either side of the spinal column in the abdomen. Each kidney is about 11 cm long and weighs about 150 g. On the concave side of the kidney lies the hilus from which the ureter and the main blood vessels

and nerves access the kidney. Figure 4 shows the two distinct regions of the kidney: the inner area with the pyramid shapes is called the medulla and the outer area just beneath the fibrous capsule is called the cortex.

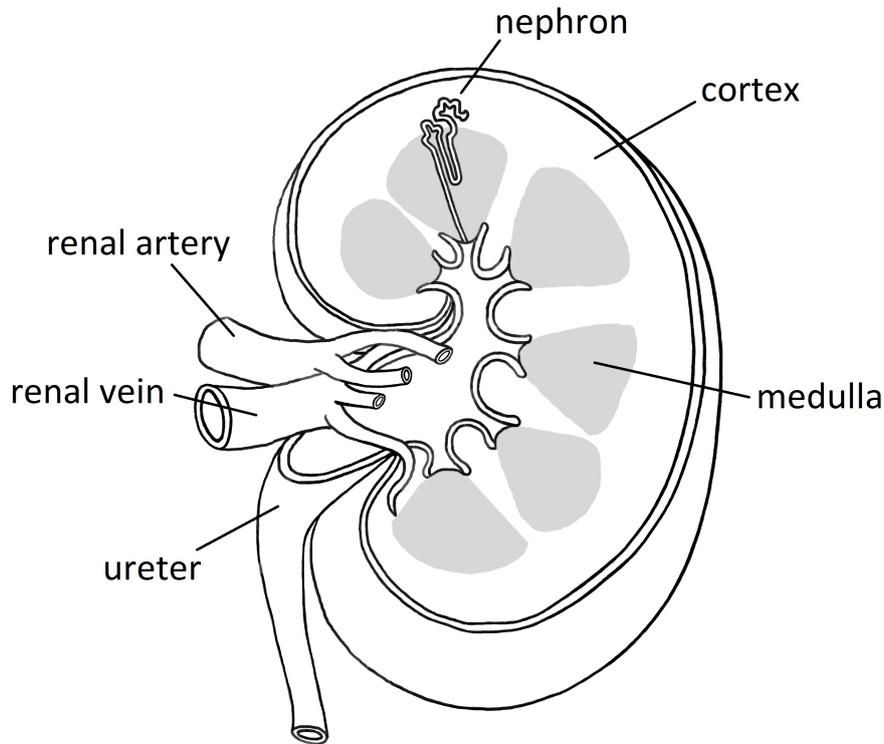


Figure 4: Structure of a kidney and the path of urine drainage. A nephron is shown in more detail in the upper half of the image. [13]

The functional unit of the kidney is called a nephron. Each kidney contains approximately 1 million nephrons. Broadly speaking, the cortex contains the filtering and reabsorptive components of the nephrons, while the medulla contains the concentrating and diluting components of the nephrons. The nephrons can be broadly divided into two types. About 85% of the nephrons are cortical nephrons which have short loops of Henle and are located in the cortex area. The other 15% of the nephrons are juxtamedullary nephrons which have long loops of Henle extending deep into the medulla area of the kidney. Figure 4 shows a nephron on scale of a whole kidney while Figure 5 shows the structure of a cortical nephron.

The structure of the nephron (Figure 5) has five components, each with a distinct process. The Bowman's capsule forms a blind-ending capsule around a knot of capillaries called the glomerulus which is the site of filtration. The proximal convoluted tubule is the site where most of the reabsorption happens along with some secretion. The loop of Henle is where the concentration and dilution of urine mainly happens. The distal convoluted tubule is the site of "fine-tuning" reabsorption and more secretion. The collecting duct is also an important site for the concentration of urine and for carrying urine into the renal pelvis at the center. These processes are interdependent and intimately related to each other by the shape of the nephron. In the end, the urine moves through the pelvis and down the ureter into the bladder.

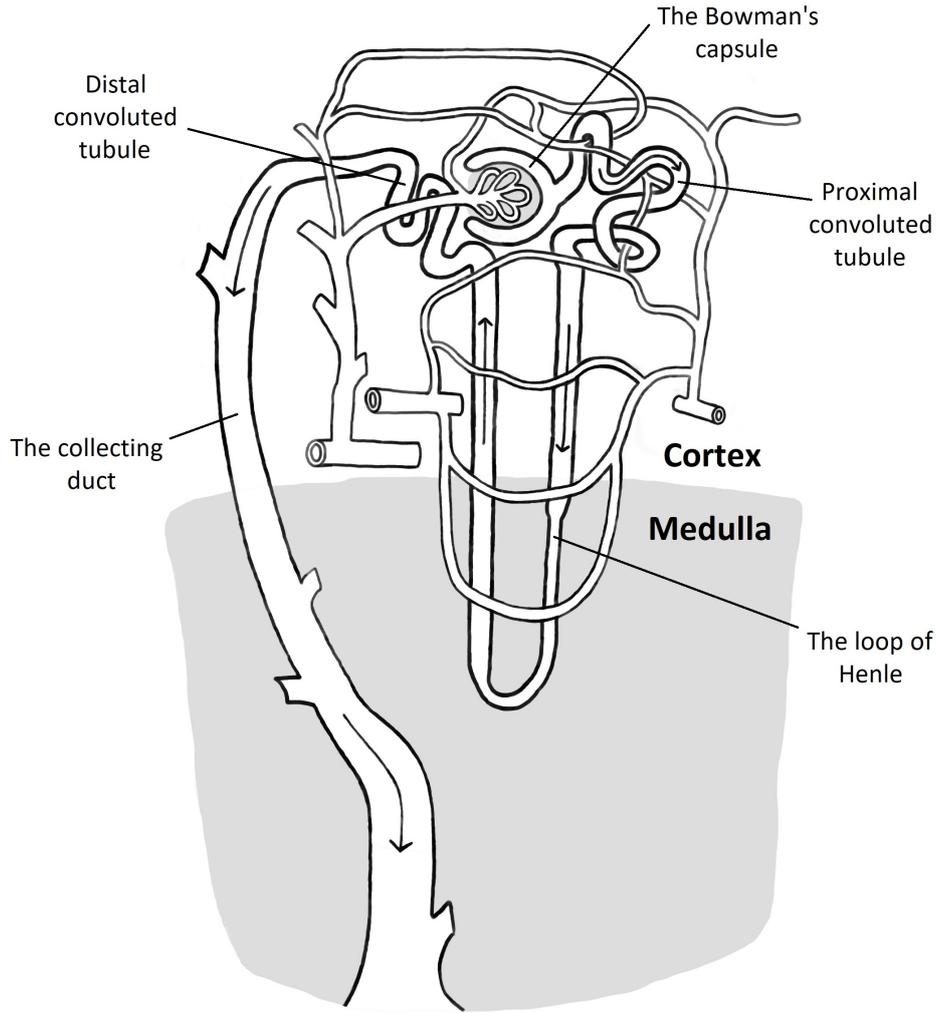


Figure 5: Structure of a cortical nephron. [13]

5.2 Blood Volume

Blood volume is something that may need to be accounted for in the analysis of PET imaging when the target region has large contribution from blood component such as the heart, kidney and liver.[7] There are no large blood vessels in the kidney, but a lot of capillaries which makes determining blood volume important.

The blood volume can be divided into arterial fraction and venous fraction. [10] We denote these respectively V_A and V_B . The V_A represents the blood that flows from arteries to the ROI. The V_B is the much lesser volume of blood that flows through smaller arteries, venules and capillaries. The V_A [ml/ml] value of a ROI can be determined from $C^{15}O$ PET images:

$$V_A = \frac{(\sum a_i)/n}{w\rho},$$

where a_i [Bq/ml] are the activity concentrations from time points i , n is the amount of time points, w [Bq/g] is the well count for said subject and ρ [g/ml] is the density of blood. The V_B value is determined as a percentage of V_A . The ratio of blood

volume was one of the areas of interest in this study since it has not been thoroughly determined in previous studies for the kidney.

5.3 Kidney modelling

Regional tissue blood flow (ml/min/g) or the perfusion has been quantitatively assessed using H_2^{15}O and PET imaging in the cerebral and myocardial tissues. A single compartment model has been a good approximation in those organs.

Quantitative assessment is still challenging in the abdominal organs such as kidney, liver and pancreas, while the tissue perfusion can be essential information in those regions. Although there are several publications which were aimed at assessing quantitative perfusion values, such as Inaba et al. [5], further improvement is needed.

In a study by Inaba et al. [5] a preliminary experiment was carried out prior to the clinical trials in order to determine how water content would be washed out from the human kidney. The results indicated that water injected into the kidney was washed out approximately monoexponentially. Based on these results, a one compartment model can be used for the analysis of renal blood flow. The input function to the kidney can be thought of as the time-activity curve of the aorta. H_2^{15}O flows in and out of the kidney by renal blood flow.

This data demonstrated only up to the 1 minute after the bolus injection of H_2^{15}O , and after that data have not been shown. An additional trapping mechanism might be needed, for example a k_3 factor for compartment model. In the following subsection we show with the Patlak plot analysis the significance of the additional component else than K_1 and k_2 .

5.4 Patlak analysis

Patlak analysis is a graphical analysis method and a model-free linearisation approach for irreversible tracer. [7] The analysis assumes that there is at least one irreversible reaction or transport step where the tracer or its labelled product can't escape.

The compartment models assume that all reversible compartments have an equilibrium with plasma so the ratio of the tracer concentration in plasma and in reversible tissue compartments must remain stable. In these circumstances the apparent distribution volume is affected only by the accumulation of tracer in irreversible compartments. However, this can happen only after the early sharp concentration changes when the plasma curve descends slow enough for tissue compartments to follow.

A simple example of a model with irreversible uptake is one tissue compartment model as shown in the Figure 6. The following calculations of Patlak model for ITCM are presented on Turku PET Centre website. [10]

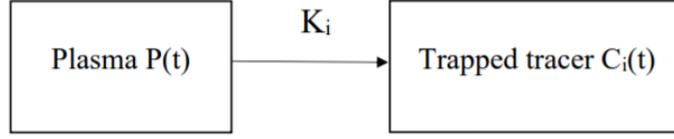


Figure 6: Patlak model in the case of 1TCM.

$P(t)$ notes the arterial input function and $C_i(t)$ is the signal from the tissue compartment irreversibly trapping the tracer. The signal we get from our ROI will be

$$ROI(t) = C_i(t) + V_B P(t), \quad (19)$$

where V_B is the blood volume value. We can see that

$$\frac{dC_i(t)}{dt} = K_i P(t),$$

where K_i is net influx rate. It's meaning becomes more relevant later. Integration of this equation gives

$$C_i(t) = K_i \int_0^t P(t) dt.$$

Equation (19) thus becomes

$$ROI(t) = K_i \int_0^t P(t) dt + V_B P(t).$$

Apart from the tracer irreversibly taken up, this signal takes the background also into consideration. The background signal can be reduced to a constant value by dividing the equation with the input function:

$$\frac{ROI(t)}{P(t)} = K_i \frac{\int_0^t P(t) dt}{P(t)} + V_B.$$

Defining $x = \frac{\int_0^t P(t) dt}{P(t)}$ and $y = \frac{ROI(t)}{P(t)}$ gives us

$$y = K_i x + V_B,$$

which is the Patlak plot. It is a straight line and its slope depicts the rate of net influx K_i . The value x corresponds to time normalized for the variations in plasma concentration. The value y is the measured signal normalized for the variations in plasma concentration and it is expressed as a unitless value

$$y = (\text{measured signal})/(\text{input function}).$$

The rate of net influx, K_i , takes into account both net inward transport and trapping of the radiotracer in tissue. If the plasma concentration was assumed to be on a steady level for a long time, after some time, the tracer concentrations in all reversible tissue compartments will reach an equilibrium. At the same time the total tissue concentration will continue to increase because of the irreversible

process. The total tissue concentration increases linearly with a slope being equal to the net influx rate K_i . This means that a positive K_i value, i.e. a positive Patlak slope, implicates that there is irreversible uptake happening in the ROI.

In practice, a bolus injection of tracer needs a short time to be well-mixed within the plasma so the Patlak plot is linear only after some time when steady-state conditions apply outside the compartment of reversible uptake.

In the case of compartment models we can see from a Patlak plot calculated for the data of this study that the slope for every ROI is positive meaning that there is irreversible uptake happening. This shows the limits of the one-tissue compartment model since it does not take in consideration the possibility of tracer getting trapped. Figure 7 shows the Patlak plot of Subject 1 calculated with Carimas.

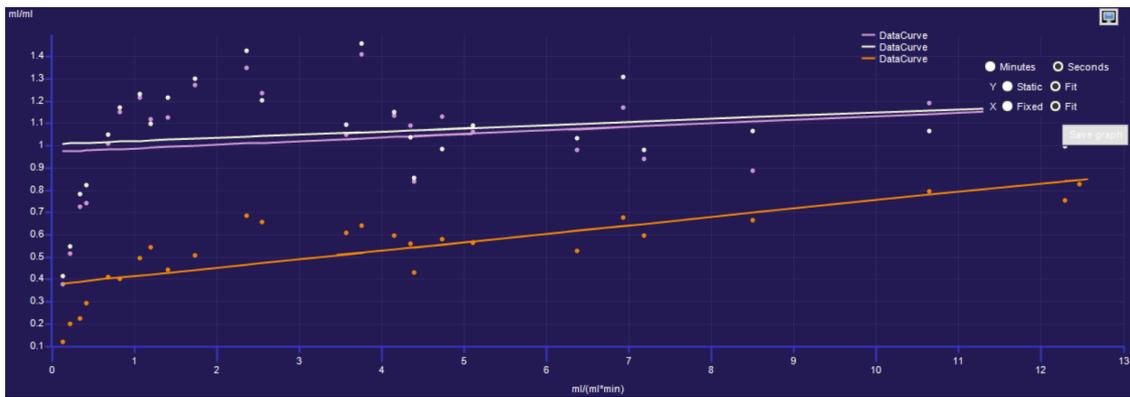


Figure 7: Patlak plot of Subject 1. The lines correspond to different ROIs following: white is Left Cortex, lilac is Right Cortex and orange is the Medulla Average.

6 Simulation

6.1 kinfitr

The fitting of compartment model values (K_1 , k_2 , k_3 and delay) and the simulation of activity curves for the ROIs have been done with programming language R. A new R package called kinfitr was utilized in the simulation. kinfitr is a package for PET kinetic modelling by Granville Matheson [8]. kinfitr utilizes a function called nls.mulstart that finds the best fit of a non-linear model based on AIC score.

Some alterations were made to the code concerning the compartment model since the blood volume correction in the kinfitr package's 1TCM and 2TCM differed from the one used in this study. The blood volume correction in kinfitr is

$$\text{compartment model} \cdot (1 - V_B) + V_B \cdot C_b,$$

where C_b is the activity concentration of the whole blood and V_B is the venous blood volume. This was changed to

$$\text{compartment model} + V_B \cdot C_a,$$

where C_a is the arterial input function. The final code used for the simulation can be found in Appendix at the end of the thesis.

Some boundaries were defined in the fittings for fitted values of K_1 , k_2 , k_3 and delay. K_1 , k_2 and k_3 were fitted between 0 and 10. Delay was fitted between -8 and 8 seconds. In reality, it is impossible for the delay to be negative, but for the purposes of seeing how the k values and delay generally vary with different V_B percentages, the fitted values were allowed to vary more freely.

7 Data Analysis

7.1 Subjects and procedure

The subjects of this study were 9 healthy young volunteers who were all male. They were aged 23 ± 2.1 years old, the minimum age being 20 and the maximum age 27. The subjects body weights varied 62 ± 7 kg, height 172 ± 6 cm and BMI 21 ± 2 . In this thesis the subjects were denoted with numbers from 1 to 9 in order to tell the subjects apart.

The used PET scanner was mCT with True V from Siemens. Image reconstruction was done by Point-Spread Function OSEM, post filter with $4 \times 4 \times 4$ mm Gaussian. The detector for $C^{15}O$ images was BeWell QS2 from Molecular Imaging Labs co. Osaka, Japan. The devices were cross-calibrated and well counts calculated.

The procedure consisted of two sets of dynamic PET imaging, the first one following intravenous $H_2^{15}O$ tracer over 1 minute and the second one following inhalation of $C^{15}O$ tracer over 2 minutes. For $H_2^{15}O$ the PET scan sequences in seconds were 1 x 25 s, 17 x 5 s, 5 x 10 s, 4 x 20 s and 6 x 30 s, in total 480 s. For the $C^{15}O$ the sequences were 8 x 60 s, in total 480 s.

The $H_2^{15}O$ tracer is chemically inert and diffusible and the tissue/blood partition coefficient of water is almost the same in each individual. The $C^{15}O$ images were obtained to correct the $H_2^{15}O$ activity concentrations in the blood, in other words, to determine the blood volume. (See subsection 7.5)

7.2 Drawing ROIs with Carimas

The data analysis in this study, meaning the drawing of each ROI and deriving the measured activities, have been done with a software called Carimas which has been developed at Turku PET Centre. Carimas can be used to dynamically visualize and analyse PET images. This is done by first drawing ROIs to PET images and then analysing the activities with models available in Carimas, such as compartment models.

While drawing ROIs we have to take into account the partial volume effect (PVE). [10] PVE is a combination of two factors: image sampling and the limited resolution of PET. Image sampling refers to the fact that PET voxel (a 3D pixel) has a definite volume, which may consist only partially of the desired tissue. Together these factors blur the images. Multiple tissue types can contribute to the measured radioactivity concentration of even single voxels, and more so to the volumes-of-interest consisting of many voxels.

There are two ways the radioactivity can spill to surrounding voxels. Spill-out refers to a situation where a single source of radioactivity spills to the surrounding area. Spill-in is the opposite situation where the surrounding radioactivity spills to area without much radioactivity. The situations are depicted in Figures 8 and 9.

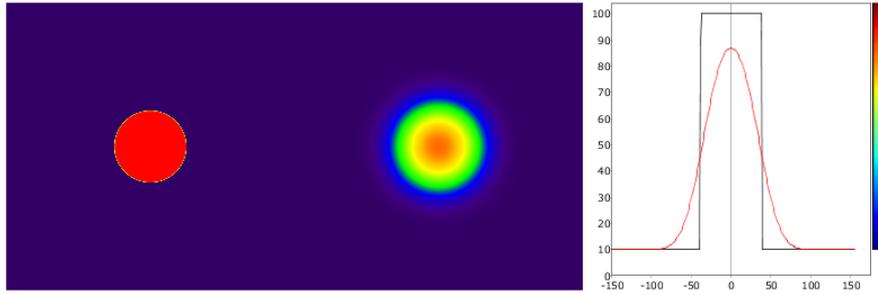


Figure 8: Spill-out.

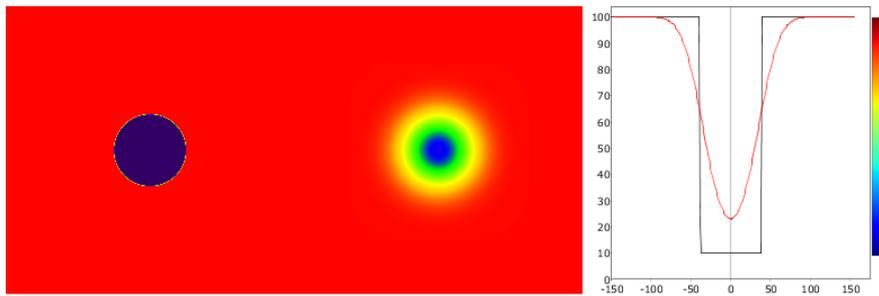


Figure 9: Spill-in.

The chosen ROIs in this study were the cortices of both left and right kidneys and the medulla area. We will call these ROIs Left Cortex, Right Cortex and Medulla Average. Additionally, an aorta ROI was drawn to function as arterial input function. Figures 10, 11 and 12 show the drawn ROIs for Subject 2 of aorta, Right Cortex (the Left Cortex is the same but on the other kidney) and Medulla Average.



Figure 10: PET image with drawn aorta ROI.

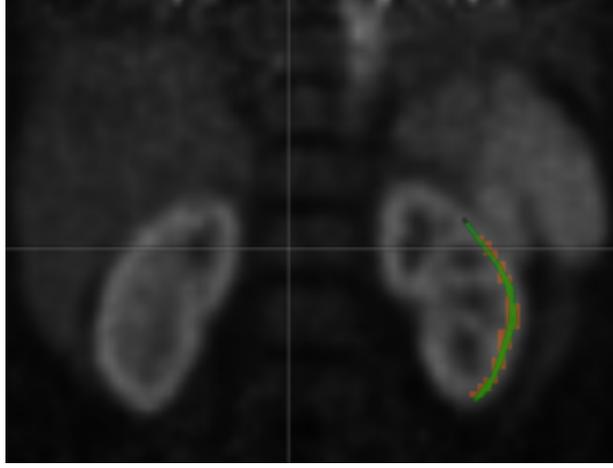


Figure 11: PET image with drawn right cortex ROI.

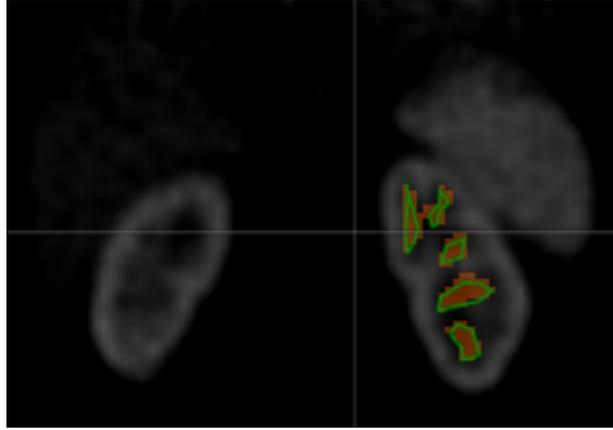


Figure 12: PET image with Medulla Average.

7.3 AIC score

Akaike Information Criterion (AIC) is one of the estimates used for measuring the goodness of both models, 1TCM and 2TCM with irreversible factor, in the analysis of this study. AIC score was proposed by Akaike in 1973 and is now a widely used estimate for model selection.

The AIC score is defined by

$$\begin{aligned} \text{AIC} &= -2l(\hat{\boldsymbol{\theta}}) - 2p \\ &= -2 \sum_{\alpha=1}^n \log f(X_{\alpha} | \hat{\boldsymbol{\theta}}) - 2p, \end{aligned}$$

where $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimator that maximizes the log-likelihood function $l(\boldsymbol{\theta})$, X_{α} are the given data and p is the number of free parameters in the model. [1] The number of free parameters is equal to the dimensions of the parameter vector $\boldsymbol{\theta}$ contained in a specified model $f(X|\boldsymbol{\theta})$. The model $f(X|\hat{\boldsymbol{\theta}})$ determined by

$\hat{\theta}$ is called the maximum likelihood model. Clearly, the larger the maximum log-likelihood $l(\hat{\theta})$ becomes, the better the model is. Thus smaller AIC scores indicate better goodness of the model.

7.4 Errors

7.4.1 Root mean square error

Another estimate for model adequacy used in the analysis is the root mean square error (RMSE). RMSE is a measure to calculate the difference between measured values and the fitted values. RMSE is calculated

$$RMSE = \sqrt{\frac{\sum_{t=1}^T (\hat{x}_t - x_t)^2}{T}}$$

where \hat{x}_t are the fitted values, x_t the measured values and T the number of observed time points. In this thesis RMSE is used to calculate the error between measured activity values and the activity curves the compartment model produces.

7.4.2 Residual standard error

In addition to the measures of model goodness, an error was calculated for the delay variable in the fittings for both models. The calculated error used was *residual standard error* (RSE) or otherwise called *residual standard deviation*. The RSE is

$$RSE = \sqrt{\frac{\sum_{i=1}^n (r_i)^2}{n - k}},$$

where r_i are the residuals, n is the number of data points and k is the number of predictors in the model. [3] Residuals are the differences between the data and the fitted values of the model. In the case of this thesis, the predictors are the variables in the model. The quantity $n - k$ is also called the degrees of freedom for estimating the residual errors. The RSE can be thought of as a measure of the average distance each observation falls from its prediction in the model.

7.5 Analysis

The blood volume V_A was first calculated from the $C^{15}O$ images. The activities in $C^{15}O$ images stabilized after 240 seconds so the activity concentrations were calculated as an average from the last time points after 240 seconds, i.e. from 240 seconds to 480 seconds. The density of blood was fixed as 1.06 g/ml. The well counts had been previously cross-calibrated and documented.

The main fittings of 1TCM and 2TCM were done with $H_2^{15}O$ activity concentrations. The fitted values derived from these fittings were K_1 , k_2 , k_3 (in the case of 2TCM) and delay. The main goal for the study was to determine the delay for different ROIs. Increasing the number of fitted variables might give the fittings too much freedom and the reasonable value boundaries in biological respect might be

crossed. Therefore the fittings were performed with fixed V_B values as different percentages of the measured blood volume derived from $C^{15}O$ images. The rounds of fitting used the fixed values of blood volume V_A from 0% to 100% with 5% intervals, leading to 21 fittings. To evaluate the models and to assess the error of this method AIC scores, root mean square errors and standard errors of delays were calculated with every fitting.

7.6 Statistical Analysis

On the statistical side of the analysis, we look into the distributions of some fitted values and measures of model goodness. The distributions of values are examined per subject or with paired subjects. For examining the possible normality of the distributions, we use *Shapiro-Wilk's test*. After that, we examine whether the paired distributions are similar or not. For this we use *paired t-test*, if both of the distributions were normal, and *Wilcoxon signed rank test*, if one or neither of the distributions were normal. In order to examine the relationships between different values from the fittings, *Kendall's τ* was used to measure the correlation.

7.6.1 Shapiro-Wilk's test

Shapiro-Wilk's test is a procedure for testing a complete sample for normality. [11] The test statistic is obtained by dividing the square of an appropriate linear combination of the sample order statistics by the usual symmetric estimate of variance.

Let $m = (m_1, \dots, m_n)$ denote the vector of expected values of standard normal order statistics and let $V = (v_{ij})$ be the corresponding $n \times n$ covariance matrix.

Additionally, let $y = (y_1, \dots, y_n)$ denote a vector of ordered random observations. If the y_i are a sample from normal distribution, for whose unknown mean is μ and unknown variance is σ^2 , then y_i may be presented as

$$y_i = \mu + \sigma x_i$$

for all $i=1, \dots, n$. The test statistic for normality is

$$W = \frac{\left(\sum_{i=1}^n a_i y_i \right)^2}{\sum_{i=1}^n (y_i - \bar{y})^2},$$

where

$$a = (a_1, \dots, a_n) = \frac{mV^{-1}}{(mV^{-1}V^{-1}m)^{1/2}}$$

and

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i.$$

If the calculated p-value from Shapiro-Wilk's test is less than the chosen critical significance level α , then the null hypothesis is rejected.

7.6.2 Paired t-test

Paired t-test is a method used for comparing the means of two dependent groups. [15] The method is based on n pairs of observations: $(X_{11}, X_{12}), (X_{21}, X_{22}), \dots, (X_{n1}, X_{n2})$, where each pair might be dependent. To form the difference scores for these pairs of observations, take their difference and denote the results by $D_1 = X_{11} - X_{12}, D_2 = X_{21} - X_{22}, \dots, D_n = X_{n1} - X_{n2}$. Let μ_1 and μ_2 denote the sample mean for the first group and second group respectively and let μ_D denote the mean of the differences. Then

$$\mu_D = \mu_1 - \mu_2.$$

The goal is to test the null hypothesis

$$H_0 : \mu_1 = \mu_2,$$

which is the same as

$$H_0 : \mu_D = 0.$$

The assumption of the null hypothesis is that the difference scores have a normal distribution. Let us denote

$$\bar{D} = \frac{1}{n} \sum_{i=1}^n D_i$$

and

$$s_D^2 = \frac{1}{n-1} \sum_{i=1}^n (D_i - \bar{D})^2.$$

The test statistic is then

$$T_D = \frac{\bar{D}}{s_D/\sqrt{n}}.$$

Assuming that D has a normal distribution, the critical value is t with $n-1$ degrees of freedom. The null hypothesis of equal means is rejected if $|T_D| \geq t$.

After the critical value t and the degrees of freedom are determined, a p-value can be found using a table of values from Student's t-distribution. If the calculated p-value is below the chosen threshold for statistical significance level α , then the null hypothesis is rejected.

7.6.3 Paired Wilcoxon signed rank test

When we can not assume that two dependent groups are normally distributed, we use the paired Wilcoxon signed rank test in order to test a hypothesis that the marginal distributions are identical between these groups. This means that it tests the hypothesis that the distribution of D , which includes the difference scores, is symmetric about zero. [15]

To apply the Wilcoxon signed rank test, the difference scores need to be formed the same way as with the paired t-test and the difference scores, that are equal to zero, are discarded. For i th pair of observations, the difference scores are then

$$D_i = X_{i1} - X_{i2},$$

where $i = 1, \dots, n$ and n indicates the number of difference scores not equal to zero. The $|D_i|$ values are ranked and the ranks associated with $|D_i|$ are denoted by U_i . If $D_i > 0$, then we set

$$R_i = U_i.$$

Otherwise, we set

$$R_i = -U_i.$$

Positive numbers are said to have a sign of 1 and negative numbers a sign of -1. Therefore, R_i is the value of the rank corresponding to $|D_i|$ multiplied by the sign of D_i .

If there are no tied values among the $|D_i|$ values, it suffices to calculate the sum of the positive R_i values as the test statistic W . The null hypothesis of identical distributions is rejected if W is sufficiently small or sufficiently large. The sum of positive R_i values determines the probability of Type I error, which is when null hypothesis is rejected, even though it holds true.

When there are tied values among the $|D_i|$ values, or if there are no tied values and the sample size is sufficiently large (for example greater than or equal to 40), the test statistic

$$W = \frac{\sum R_i}{\sqrt{\sum R_i^2}}$$

can be used. The null hypothesis is rejected if $|W| \geq c$, where c is the $1 - \alpha/2$ quantile of a standard normal distribution. α is the chosen critical significance level for the test. Rejecting the null hypothesis with the signed rank test indicates that the two dependent groups have different distributions.

When the p-value of the paired Wilcoxon signed rank test is less than the chosen critical significance level α , the null hypothesis is rejected, just like with Shapiro-Wilk's test and paired t-test.

7.6.4 Kendall's τ

Kendall's rank correlation coefficient or Kendall's tau (τ) is used to measure the monotonic association between two measured quantities. [15] A monotonic relationship is considered as consistently increasing or decreasing, but it does not have to be a straight line.

Consider two pairs of observations, which are (X_1, Y_1) and (X_2, Y_2) . This pair is said to be concordant, if Y increases as X increases or if Y decreases as X decreases. Otherwise, the pair is called discordant. Put another way, the slope between the two points is positive, when the pair is concordant. The slope between the two points is negative, when the pair is discordant.

Based on all pairs of points, Kendall's τ is the difference between the number of positive slopes and the number of negative slopes divided by the total number of slopes. Thus, its value is between -1 and 1. If all pairs of points are concordant, Kendall's τ is equal to 1. If all pairs of points are discordant, Kendall's τ is equal to -1. When Kendall's τ is 1, the two rankings are the same, and when Kendall's τ is -1, the other ranking is the reverse of the other.

8 Results

The fitting results included the fitted values, K_1 , k_2 , k_3 and delay, and the measures of model goodness, AIC score and RMSE. In addition the RSE of delay was calculated. Tables 1 and 2 present the fitted and calculated values of 1TCM and 2TCM for Subject 1 as an example of the fitting results. Fittings were done with 5% intervals of V_B value, but for visual purposes Tables 1 and 2 have the results with 10% intervals.

Table 1: The example results of fitting 1TCM for Left Cortex of Subject 1. Results are shown here with 10% intervals of fixed V_B values as a ratio of V_A . The units of K_1 and k_2 are [ml/ml] and the unit of delay is [s].

	V_B	K_1	k_2	delay	RSE of delay	AIC	RMSE
$V_B/V_A \cdot 100\% = 0\%$							
Left Cortex	0.00	4.33	4.64	-0.01	79.33	628.08	2824.01
Right Cortex	0.00	3.80	4.19	-0.22	77.02	625.65	2722.06
Medulla Average	0.00	0.92	2.10	-2.96	498.79	630.42	2925.96
$V_B/V_A \cdot 100\% = 10\%$							
Left Cortex	0.01	4.28	4.64	0.14	33.82	628.07	2823.66
Right Cortex	0.01	3.75	4.19	-0.03	11.29	625.64	2721.50
Medulla Average	0.03	0.86	2.08	-1.35	6.96	630.09	2911.36
$V_B/V_A \cdot 100\% = 20\%$							
Left Cortex	0.02	4.22	4.63	0.29	15.72	628.05	2822.83
Right Cortex	0.02	3.70	4.18	0.14	14.23	625.62	2720.73
Medulla Average	0.05	0.78	2.01	0.12	4.64	629.20	2872.28
$V_B/V_A \cdot 100\% = 30\%$							
Left Cortex	0.03	4.17	4.63	0.44	10.10	628.02	2821.50
Right Cortex	0.04	3.64	4.18	0.32	8.61	625.60	2719.66
Medulla Average	0.08	0.70	1.89	1.39	2.67	627.86	2814.46
$V_B/V_A \cdot 100\% = 40\%$							
Left Cortex	0.05	4.11	4.62	0.59	7.36	627.98	2819.68
Right Cortex	0.05	3.59	4.17	0.50	6.61	625.56	2718.21
Medulla Average	0.10	0.61	1.73	2.54	1.76	626.22	2745.32
$V_B/V_A \cdot 100\% = 50\%$							
Left Cortex	0.06	4.05	4.62	0.73	5.75	627.92	2817.37
Right Cortex	0.06	3.53	4.16	0.67	4.98	625.52	2716.40
Medulla Average	0.13	0.51	1.54	3.56	1.27	624.86	2689.44
$V_B/V_A \cdot 100\% = 60\%$							
Left Cortex	0.07	3.99	4.61	0.87	4.68	627.86	2814.59
Right Cortex	0.07	3.47	4.15	0.84	4.18	625.46	2714.29
Medulla Average	0.16	0.42	1.31	4.41	1.01	624.93	2692.57
$V_B/V_A \cdot 100\% = 70\%$							
Left Cortex	0.08	3.93	4.59	1.01	0.00	627.78	2811.36
Right Cortex	0.08	3.41	4.14	1.00	3.41	625.41	2711.92
Medulla Average	0.18	0.34	1.11	6.28	0.96	623.87	2649.65
$V_B/V_A \cdot 100\% = 80\%$							
Left Cortex	0.09	3.87	4.58	1.15	3.52	627.70	2807.71
Right Cortex	0.10	3.35	4.12	1.16	2.98	625.34	2709.36
Medulla Average	0.21	0.26	0.88	7.05	0.79	624.63	2680.08
$V_B/V_A \cdot 100\% = 90\%$							
Left Cortex	0.10	3.80	4.57	1.29	3.04	627.60	2803.68
Right Cortex	0.11	3.29	4.11	1.32	2.54	625.28	2706.70
Medulla Average	0.23	0.20	0.68	7.74	0.72	629.03	2864.93
$V_B/V_A \cdot 100\% = 100\%$							
Left Cortex	0.11	3.74	4.55	1.42	2.66	627.50	2799.31
Right Cortex	0.12	3.23	4.09	1.48	2.21	625.21	2704.03
Medulla Average	0.26	0.15	0.50	8.00	0.72	637.46	3255.19

Table 2: The results of fitting 2TCM for Subject 1. Results are shown here with 10% intervals of fixed V_B values as a ratio of V_A . The units of K_1 and k_2 are [ml/ml] and the unit of delay is [s].

	V_B	K_1	k_2	k_3	delay	RSE of delay	AIC	RMSE
$V_B/V_A \cdot 100\% = 0\%$								
Left Cortex	0.00	4.56	5.01	0.02	0.33	87.84	622.48	2441.86
Right Cortex	0.00	4.00	4.52	0.02	0.13	89.27	620.02	2352.31
Medulla Average	0.00	1.19	3.21	0.10	-0.67	109.61	594.60	1600.54
$V_B/V_A \cdot 100\% = 10\%$								
Left Cortex	0.01	4.50	5.01	0.02	0.47	32.83	622.48	2441.69
Right Cortex	0.01	3.94	4.51	0.02	0.30	24.26	620.02	2352.29
Medulla Average	0.03	1.10	3.16	0.10	0.53	6.71	594.51	1598.15
$V_B/V_A \cdot 100\% = 20\%$								
Left Cortex	0.02	4.45	5.00	0.02	0.62	15.46	622.47	2441.29
Right Cortex	0.02	3.88	4.51	0.02	0.48	13.75	620.02	2352.24
Medulla Average	0.05	0.99	3.06	0.11	1.62	2.92	594.53	1598.84
$V_B/V_A \cdot 100\% = 30\%$								
Left Cortex	0.03	4.38	5.00	0.02	0.76	9.96	622.45	2440.66
Right Cortex	0.04	3.83	4.51	0.02	0.64	8.38	620.01	2352.18
Medulla Average	0.08	0.87	2.89	0.12	2.60	1.67	595.04	1611.23
$V_B/V_A \cdot 100\% = 40\%$								
Left Cortex	0.05	4.32	4.99	0.02	0.90	7.26	622.43	2439.81
Right Cortex	0.05	3.77	4.50	0.02	0.81	6.47	620.01	2352.14
Medulla Average	0.10	0.75	2.66	0.12	3.48	1.15	596.89	1656.99
$V_B/V_A \cdot 100\% = 50\%$								
Left Cortex	0.06	4.26	4.98	0.02	1.03	5.66	622.40	2438.76
Right Cortex	0.06	3.71	4.49	0.02	0.97	4.87	620.01	2352.17
Medulla Average	0.13	0.62	2.36	0.13	4.24	0.90	601.28	1770.95
$V_B/V_A \cdot 100\% = 60\%$								
Left Cortex	0.07	4.20	4.97	0.02	1.17	4.60	622.37	2437.53
Right Cortex	0.07	3.64	4.48	0.02	1.13	3.87	620.02	2352.31
Medulla Average	0.16	0.52	2.20	0.14	5.85	1.13	606.01	1902.45
$V_B/V_A \cdot 100\% = 70\%$								
Left Cortex	0.08	4.13	4.96	0.02	1.30	3.85	622.33	2436.14
Right Cortex	0.08	3.58	4.46	0.02	1.29	3.34	620.03	2352.65
Medulla Average	0.18	0.41	1.85	0.15	6.56	0.92	610.95	2050.28
$V_B/V_A \cdot 100\% = 80\%$								
Left Cortex	0.09	4.06	4.94	0.02	1.43	3.29	622.29	2434.65
Right Cortex	0.10	3.52	4.44	0.02	1.44	2.80	620.04	2353.26
Medulla Average	0.21	0.32	1.48	0.15	7.18	0.81	618.51	2299.19
$V_B/V_A \cdot 100\% = 90\%$								
Left Cortex	0.10	3.99	4.93	0.02	1.56	2.86	622.25	2433.10
Right Cortex	0.11	3.45	4.42	0.02	1.60	2.40	620.07	2354.26
Medulla Average	0.23	0.23	1.12	0.15	7.79	0.78	628.42	2671.54
$V_B/V_A \cdot 100\% = 100\%$								
Left Cortex	0.11	3.92	4.91	0.02	1.68	2.42	622.21	2431.56
Right Cortex	0.12	3.38	4.40	0.02	1.74	2.09	620.12	2355.78
Medulla Average	0.26	0.17	0.78	0.13	8.00	0.80	640.06	3187.17

To see the overall results, the mean and standard deviation of all values between all subjects are shown in the following tables. Tables 3, 4 and 5 are the results with 1TCM for Left Cortex, Right Cortex and Medulla Average respectively. For visual purposes, the tables have the results with 10% intervals, even though the fittings were done with 5% intervals.

Table 3: The means and standard deviations calculated from Left Cortex of all subjects modelled with 1TCM. Results are shown here with 10% intervals of fixed V_B values as a ratio of V_A . The units of K_1 and k_2 are [ml/ml] and the unit of delay is [s].

$V_B/V_A \cdot 100\%$	V_B	K_1		k_2		delay	
		mean	sd	mean	sd	mean	sd
0 %	0.00	3.45	0.73	3.68	0.76	-0.85	0.91
10 %	0.01	3.40	0.72	3.68	0.76	-0.59	0.85
20 %	0.03	3.35	0.72	3.67	0.76	-0.34	0.80
30 %	0.04	3.29	0.71	3.67	0.76	-0.10	0.76
40 %	0.06	3.23	0.70	3.66	0.76	0.14	0.72
50 %	0.08	3.18	0.70	3.66	0.77	0.40	0.69
60 %	0.08	3.12	0.69	3.65	0.77	0.63	0.67
70 %	0.10	3.06	0.68	3.63	0.77	0.84	0.67
80 %	0.11	2.99	0.68	3.61	0.77	1.06	0.66
90 %	0.13	2.93	0.67	3.58	0.77	1.27	0.68
100 %	0.14	2.86	0.67	3.56	0.78	1.46	0.72
		AIC		RMSE		RSE of delay	
		mean	sd	mean	sd	mean	sd
0 %	0.00	621.60	26.25	2751.92	978.86	160.62	84.93
10 %	0.01	621.57	26.27	2751.03	978.49	20.72	8.73
20 %	0.03	621.50	26.31	2748.57	977.59	9.65	5.26
30 %	0.04	621.40	26.36	2744.80	975.87	8.54	2.94
40 %	0.06	621.28	26.39	2739.98	972.93	6.19	2.15
50 %	0.08	621.09	26.42	2732.10	968.83	5.50	1.38
60 %	0.08	620.85	26.45	2722.28	963.81	4.43	1.08
70 %	0.10	620.59	26.48	2711.69	958.44	3.46	1.44
80 %	0.11	620.31	26.51	2699.93	952.20	3.31	0.67
90 %	0.13	620.00	26.51	2686.95	945.00	2.93	0.64
100 %	0.14	619.77	26.55	2674.11	938.85	2.69	0.79

Table 4: The means and standard deviations calculated from Right Cortex of all subjects modelled with 1TCM. Results are shown here with 10% intervals of fixed V_B values as a ratio of V_A . The units of K_1 and k_2 are [ml/ml] and the unit of delay is [s].

$V_B/V_A \cdot 100\%$	V_B	K_1		k_2		delay	
		mean	sd	mean	sd	mean	sd
0 %	0.00	3.38	0.33	3.87	0.31	-0.61	0.44
10 %	0.01	3.32	0.33	3.87	0.31	-0.36	0.44
20 %	0.03	3.26	0.33	3.86	0.31	-0.13	0.44
30 %	0.04	3.21	0.33	3.85	0.31	0.11	0.45
40 %	0.06	3.15	0.33	3.85	0.32	0.35	0.46
50 %	0.07	3.09	0.33	3.83	0.32	0.56	0.49
60 %	0.08	3.02	0.33	3.81	0.33	0.77	0.52
70 %	0.10	2.95	0.33	3.79	0.33	0.98	0.54
80 %	0.11	2.89	0.33	3.77	0.33	1.19	0.56
90 %	0.12	2.82	0.34	3.75	0.34	1.39	0.59
100 %	0.14	2.75	0.34	3.72	0.34	1.58	0.61
		AIC		RMSE		RSE of delay	
		mean	sd	mean	sd	mean	sd
0 %	0.00	613.90	19.00	2371.13	654.20	130.54	46.71
10 %	0.01	613.88	19.00	2370.32	653.99	17.83	8.19
20 %	0.03	613.83	18.99	2368.44	653.43	10.64	3.45
30 %	0.04	613.75	18.98	2365.52	652.41	6.83	2.08
40 %	0.06	613.63	18.95	2361.01	651.03	6.60	2.55
50 %	0.07	613.48	18.93	2355.54	650.05	5.42	2.02
60 %	0.08	613.35	18.91	2350.89	648.68	3.93	1.79
70 %	0.10	613.22	18.88	2345.93	646.87	3.87	1.10
80 %	0.11	613.07	18.83	2340.39	644.68	3.28	0.93
90 %	0.12	612.92	18.77	2334.34	642.04	2.81	0.81
100 %	0.14	612.79	18.69	2327.92	638.89	2.08	0.98

Table 5: The means and standard deviations calculated from Medulla Average of all subjects modelled with 1TCM. Results are shown here with 10% intervals of fixed V_B values as a ratio of V_A . The units of K_1 and k_2 are [ml/ml] and the unit of delay is [s].

$V_B/V_A \cdot 100\%$	V_B	K_1		k_2		delay	
		mean	sd	mean	sd	mean	sd
0 %	0.00	0.93	0.16	1.65	0.41	-5.61	1.99
10 %	0.02	0.88	0.15	1.64	0.41	-4.10	1.96
20 %	0.05	0.83	0.15	1.60	0.40	-2.67	1.93
30 %	0.07	0.77	0.14	1.54	0.39	-1.30	1.91
40 %	0.10	0.70	0.14	1.46	0.39	-0.03	2.01
50 %	0.11	0.63	0.14	1.36	0.38	1.10	2.08
60 %	0.14	0.56	0.14	1.24	0.37	2.34	2.38
70 %	0.17	0.49	0.14	1.12	0.35	3.51	2.43
80 %	0.19	0.42	0.14	0.98	0.33	4.38	2.46
90 %	0.22	0.36	0.13	0.85	0.32	5.18	2.26
100 %	0.24	0.31	0.13	0.72	0.30	5.71	2.02
		AIC		RMSE		RSE of delay	
		mean	sd	mean	sd	mean	sd
0 %	0.00	633.74	22.95	3257.92	1052.11	604.52	131.24
10 %	0.02	633.50	22.94	3245.84	1047.13	12.94	3.49
20 %	0.05	632.82	22.84	3210.71	1029.77	6.39	1.72
30 %	0.07	631.59	22.74	3149.39	1003.25	4.21	1.20
40 %	0.10	629.75	22.57	3060.14	967.71	3.02	0.75
50 %	0.11	627.28	22.15	2942.68	920.56	2.38	0.82
60 %	0.14	624.33	22.00	2812.87	884.24	1.90	0.63
70 %	0.17	620.21	22.62	2652.52	872.27	1.55	0.55
80 %	0.19	616.44	23.47	2516.41	858.81	1.21	0.43
90 %	0.22	615.09	22.94	2457.84	808.29	1.00	0.30
100 %	0.24	616.10	23.71	2503.83	825.95	0.88	0.22

According to tables 3 and 4, the K_1 and k_2 seem to decrease with increasing V_B value in Left Cortex and Right Cortex with 1TCM. Conversely, the delay seems to increase when V_B value increases. The goodness-of-fit measures, AIC and RMSE, and the RSE of delay seem to decrease with increasing V_B value. In both Left Cortex and Right Cortex, the smallest AIC scores and RMSE values seem to be in the fittings with V_B value being 100% of V_A . This could mean that the most suitable V_B percentage for the cortices could be 100%.

Table 5 shows that the values of Medulla Average with 1TCM behave very much the same than the values of the cortices. The smallest AIC and RMSE values seem both to be in the fitting with V_B being 90% of V_A . This can suggest that the most suitable V_B percentage could be 90% for Medulla Average with 1TCM.

Tables 6, 7 and 8 are the means and standard deviations of values with 2TCM for Left Cortex, Right Cortex and Medulla Average respectively.

Table 6: The means and standard deviations calculated from Left Cortex of all subjects modelled with 2TCM. Results are shown here with 10% intervals of fixed V_B values as a ratio of V_A . The units of K_1 , k_2 and k_3 are [ml/ml] and the unit of delay is [s].

$V_B/V_A \cdot 100\%$	V_B	K_1		k_2		k_3			
		mean	sd	mean	sd	mean	sd		
0 %	0.00	3.79	0.74	4.22	0.80	0.03	0.01		
10 %	0.01	3.73	0.73	4.22	0.80	0.03	0.01		
20 %	0.03	3.67	0.73	4.22	0.80	0.03	0.01		
30 %	0.04	3.61	0.72	4.21	0.80	0.03	0.01		
40 %	0.06	3.54	0.71	4.19	0.81	0.03	0.01		
50 %	0.08	3.47	0.71	4.17	0.81	0.03	0.01		
60 %	0.08	3.40	0.70	4.16	0.81	0.03	0.01		
70 %	0.10	3.33	0.69	4.14	0.81	0.03	0.01		
80 %	0.11	3.25	0.69	4.11	0.81	0.03	0.01		
90 %	0.13	3.18	0.68	4.08	0.81	0.03	0.01		
100 %	0.14	3.10	0.68	4.05	0.81	0.03	0.01		
		delay		AIC		RMSE		RSE of delay	
		mean	sd	mean	sd	mean	sd	mean	sd
0 %	0.00	-0.08	0.72	608.86	21.99	2080.58	531.38	185.75	135.18
10 %	0.01	0.15	0.68	608.83	22.00	2079.70	531.27	22.49	9.77
20 %	0.03	0.37	0.64	608.76	22.03	2077.52	530.74	11.63	4.32
30 %	0.04	0.59	0.60	608.65	22.03	2074.16	529.49	7.46	2.80
40 %	0.06	0.80	0.59	608.52	22.04	2070.05	527.97	6.54	1.57
50 %	0.08	1.00	0.59	608.39	22.06	2066.25	527.12	5.40	1.40
60 %	0.08	1.20	0.58	608.26	22.08	2062.00	526.18	4.36	1.10
70 %	0.10	1.40	0.58	608.10	22.08	2057.17	524.80	3.61	0.92
80 %	0.11	1.59	0.59	607.94	22.04	2051.85	522.81	3.05	0.78
90 %	0.13	1.78	0.60	607.78	21.93	2046.20	520.03	2.61	0.67
100 %	0.14	1.97	0.62	607.73	21.79	2040.42	516.20	2.26	0.58

Table 7: The means and standard deviations calculated from Right Cortex of all subjects modelled with 2TCM. Results are shown here with 10% intervals of fixed V_B values as a ratio of V_A . The units of K_1 , k_2 and k_3 are [ml/ml] and the unit of delay is [s].

$V_B/V_A \cdot 100\%$	V_B	K_1		k_2		k_3			
		mean	sd	mean	sd	mean	sd		
0 %	0.00	3.73	0.31	4.47	0.45	0.03	0.01		
10 %	0.01	3.66	0.31	4.47	0.45	0.03	0.01		
20 %	0.03	3.60	0.31	4.46	0.45	0.03	0.01		
30 %	0.04	3.53	0.31	4.45	0.45	0.03	0.01		
40 %	0.06	3.46	0.31	4.43	0.46	0.03	0.01		
50 %	0.07	3.39	0.31	4.41	0.46	0.03	0.01		
60 %	0.08	3.31	0.31	4.39	0.46	0.03	0.01		
70 %	0.10	3.24	0.32	4.37	0.46	0.03	0.01		
80 %	0.11	3.16	0.32	4.34	0.47	0.03	0.01		
90 %	0.12	3.08	0.33	4.31	0.47	0.03	0.01		
100 %	0.14	3.00	0.33	4.28	0.47	0.03	0.01		
		delay		AIC		RMSE		RSE of delay	
		mean	sd	mean	sd	mean	sd	mean	sd
0 %	0.00	0.15	0.56	600.25	19.51	1817.09	496.73	499.23	739.65
10 %	0.01	0.37	0.56	600.24	19.50	1816.59	496.44	21.55	7.89
20 %	0.03	0.58	0.57	600.20	19.46	1815.36	495.70	11.34	4.18
30 %	0.04	0.79	0.58	600.14	19.40	1813.40	494.52	7.39	2.82
40 %	0.06	0.98	0.61	600.09	19.34	1811.48	493.32	5.19	2.53
50 %	0.07	1.18	0.62	600.07	19.28	1810.51	492.44	4.82	1.22
60 %	0.08	1.37	0.64	600.05	19.20	1809.53	491.36	3.43	1.55
70 %	0.10	1.56	0.65	600.03	19.10	1808.58	490.05	2.85	1.30
80 %	0.11	1.74	0.66	600.03	18.98	1807.74	488.51	2.40	1.08
90 %	0.12	1.92	0.68	600.04	18.83	1807.15	486.74	2.32	0.59
100 %	0.14	2.10	0.69	600.09	18.67	1806.97	484.73	1.86	0.81

Table 8: The means and standard deviations calculated from Medulla Average of all subjects modelled with 2TCM. Results are shown here with 10% intervals of fixed V_B values as a ratio of V_A . The units of K_1 , k_2 and k_3 are [ml/ml] and the unit of delay is [s].

$V_B/V_A \cdot 100\%$	V_B	K_1		k_2		k_3			
		mean	sd	mean	sd	mean	sd		
0 %	0.00	1.42	0.22	3.26	0.44	0.12	0.03		
10 %	0.02	1.33	0.21	3.23	0.44	0.13	0.03		
20 %	0.05	1.24	0.21	3.16	0.44	0.14	0.04		
30 %	0.07	1.13	0.21	3.06	0.45	0.15	0.04		
40 %	0.10	1.02	0.21	2.89	0.45	0.15	0.04		
50 %	0.11	0.90	0.21	2.69	0.44	0.16	0.04		
60 %	0.14	0.79	0.21	2.47	0.45	0.17	0.05		
70 %	0.17	0.67	0.21	2.18	0.50	0.17	0.05		
80 %	0.19	0.57	0.20	1.92	0.60	0.17	0.04		
90 %	0.22	0.47	0.19	1.60	0.62	0.16	0.02		
100 %	0.24	0.39	0.18	1.31	0.62	0.13	0.05		
		delay		AIC		RMSE		RSE of delay	
		mean	sd	mean	sd	mean	sd	mean	sd
0 %	0.00	-1.29	0.99	587.05	15.64	1465.42	317.55	210.31	83.89
10 %	0.02	-0.29	1.07	587.01	15.61	1464.42	316.61	8.80	2.90
20 %	0.05	0.62	1.15	586.87	15.58	1461.22	315.46	4.20	1.19
30 %	0.07	1.49	1.19	586.73	15.47	1457.83	313.34	3.09	1.44
40 %	0.10	2.25	1.25	586.93	15.19	1460.87	309.50	2.21	0.96
50 %	0.11	3.05	1.54	587.75	14.99	1478.07	307.90	1.42	0.79
60 %	0.14	3.88	1.82	588.14	15.76	1490.76	326.60	1.05	0.67
70 %	0.17	4.52	1.96	589.50	16.22	1524.23	345.70	1.06	0.32
80 %	0.19	5.25	2.01	592.43	16.68	1596.58	379.01	0.98	0.24
90 %	0.22	5.67	1.92	597.16	18.55	1729.67	478.87	0.86	0.15
100 %	0.24	6.10	1.76	603.09	23.27	1940.32	718.14	0.71	0.27

The tables 6 and 7 would suggest that all the values behave similarly with 2TCM than with 1TCM in Left Cortex and Right Cortex when the V_B value increases. It seems that the k_3 value in 2TCM does not vary significantly between fittings of different V_B values in Left Cortex or Right Cortex. The smallest AIC and RMSE values seem to be in the fittings with V_B being 100% of V_A with both 1TCM and 2TCM. This can suggest that the most suitable V_B percentage could be 100% for Left Cortex and Right Cortex with both 1TCM and 2TCM.

The values for Medulla Average with 2TCM in Table 8 look similar to the values with 1TCM in Table 5 apart from the goodness-of-fit measures, AIC and RMSE. The smallest AIC and RMSE values seem both to be in the fitting with V_B being 30% of V_A . This can suggest that the most suitable V_B percentage could be 30% for Medulla Average with 2TCM.

In every ROI, the AIC scores and RMSE values with 2TCM seem to be significantly lower than with 1TCM. This could mean that 2TCM is a better model than 1TCM to depict renal plasma flow.

Let us take a look at the AIC scores with all subjects in order to see more accurately how they behave. Figures 13, 14 and 15 show the AIC scores of all subjects with 1TCM and 2TCM in different ROIs.

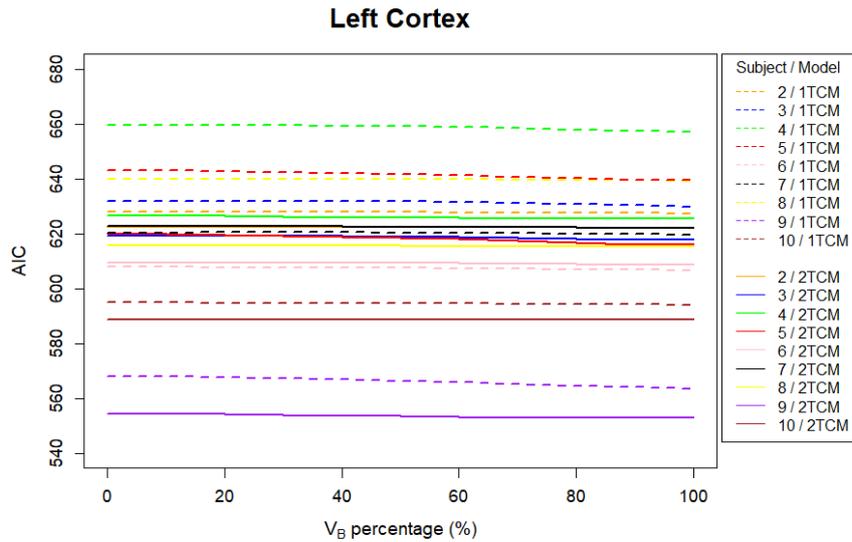


Figure 13: AIC scores of every subject with 1TCM and 2TCM in left cortex.

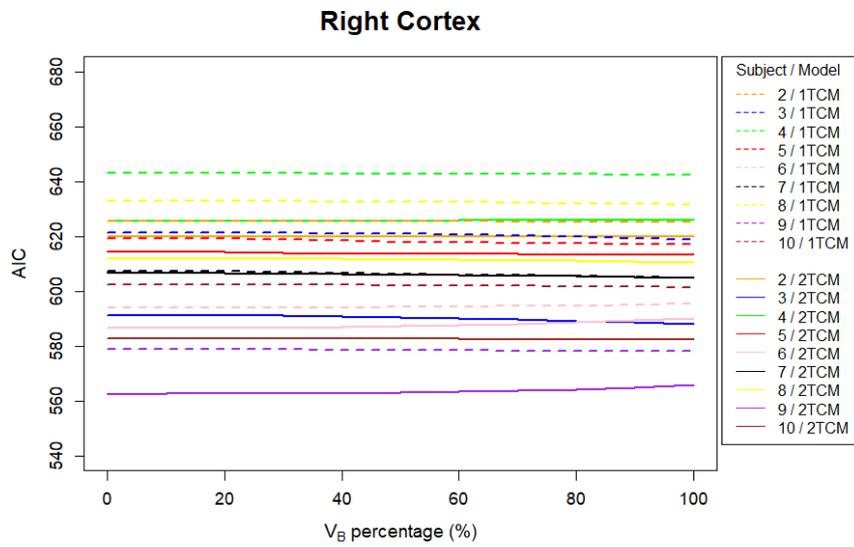


Figure 14: AIC scores of every subject with 1TCM and 2TCM in the right cortex.

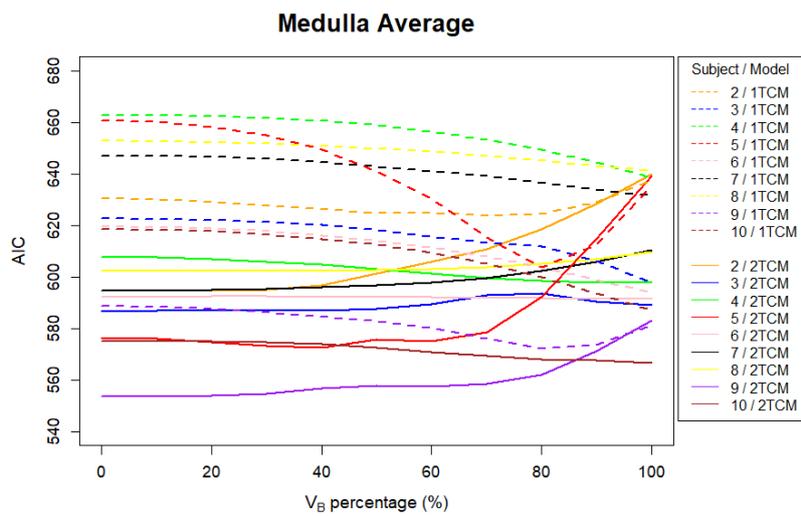


Figure 15: AIC scores of every subject with 1TCM and 2TCM in the medulla.

Clearly the AIC scores are smaller with 2TCM than with 1TCM when observing the subjects one at a time. This could suggest that 2TCM is a better model for all the ROIs. The AIC scores do not seem to fluctuate significantly in Left Cortex and Right Cortex when V_B value changes. The AIC scores in Medulla Average seem to change drastically at around 70% and over it. The AIC scores seem to vary between subjects even with the same model. Figure 16 shows the average AIC scores from all 21 fittings.

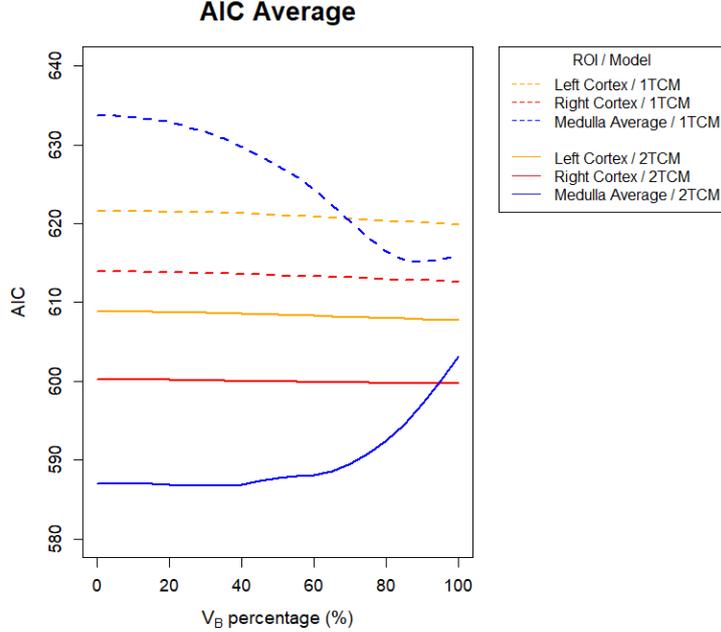


Figure 16: The average of AIC scores from all 21 fittings with respect to V_B percentage out of V_A .

According to Figure 16, the best AIC scores in Left Cortex and Right Cortex seem to be when the V_B value is around 100% of V_A with both 1TCM and 2TCM. For Medulla Average, the smallest AIC scores seem to be around 90% of V_A for 1TCM and around 20-40% of V_A for 2TCM.

The AIC scores of each subject were tested for normality with Shapiro-Wilk test. After that the subjects were paired with each other in paired t-test or paired Wilcoxon signed rank test depending on the result of the Shapiro-Wilk's test. The chosen significance level α was determined to be 0.05. If the p-value of both subject's distributions in Shapiro-Wilk's test were above the significance level of $\alpha = 0.05$, then the distributions were noted as "normal", and the paired t-test was used to test the differences of the distributions. If the p-value of either or both subject's distributions in Shapiro-Wilk's test were below the significance level of $\alpha = 0.05$, then the distributions were noted as "not normal", and the paired Wilcoxon signed rank was used to test the differences of the distributions. When the p-value for either paired t-test or Wilcoxon signed rank test were above the significance level $\alpha = 0.05$, the distributions were noted to be "not significantly different", and otherwise "significantly different". Figures 17 and 18 show the noted normality and distribution differences for each pair of subjects with 1TCM and 2TCM respectively.

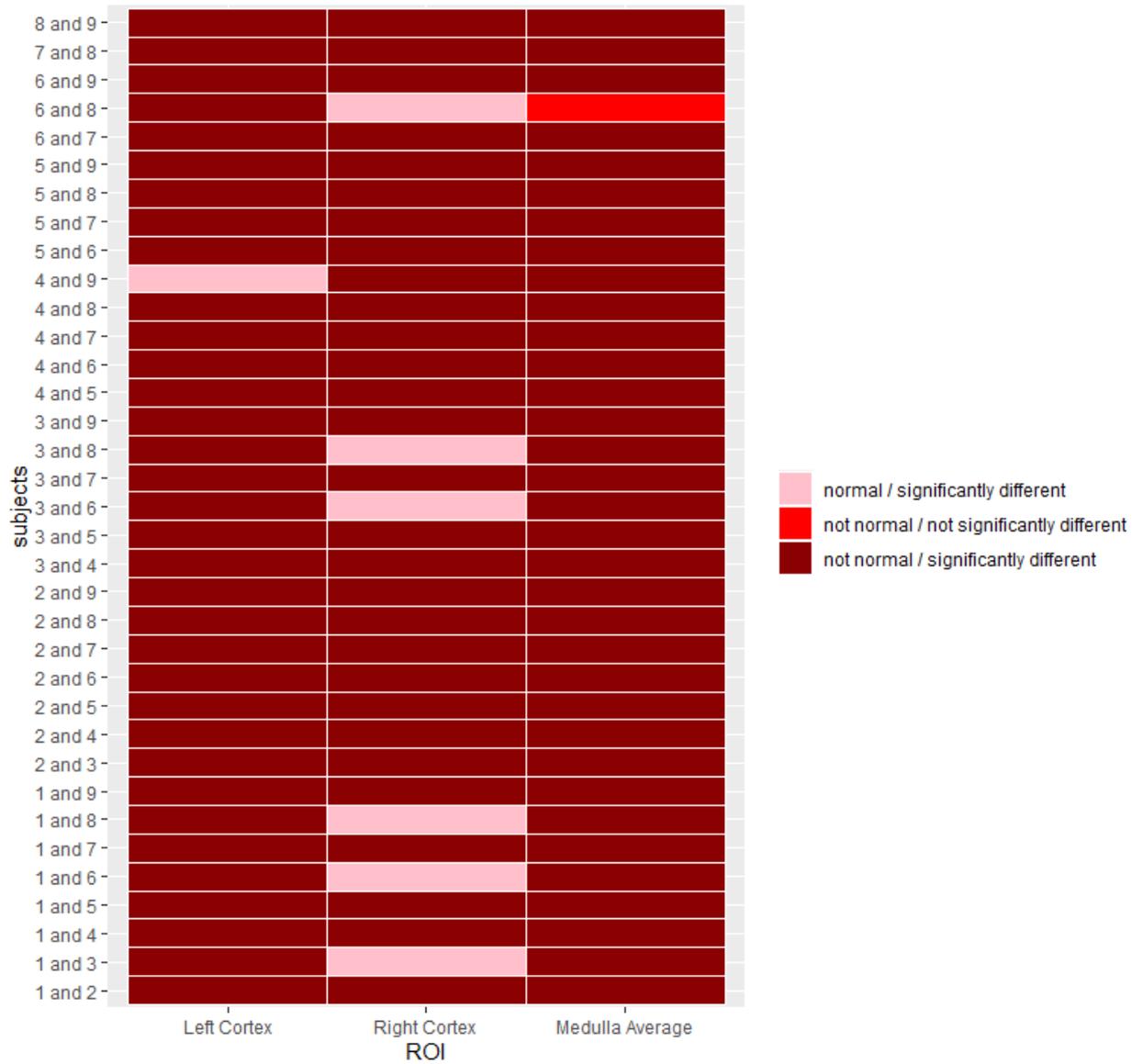


Figure 17: AIC score results from Shapiro-Wilk's test and either paired t-test or paired Wilcoxon signed rank test with 1TCM. The first part shows if the AIC scores were both normally distributed or not. "not normal" means that at least one of the two subjects was not normally distributed according to Shapiro-Wilk's test. The second part tells if the distributions of AIC scores of the two subjects were significantly different.

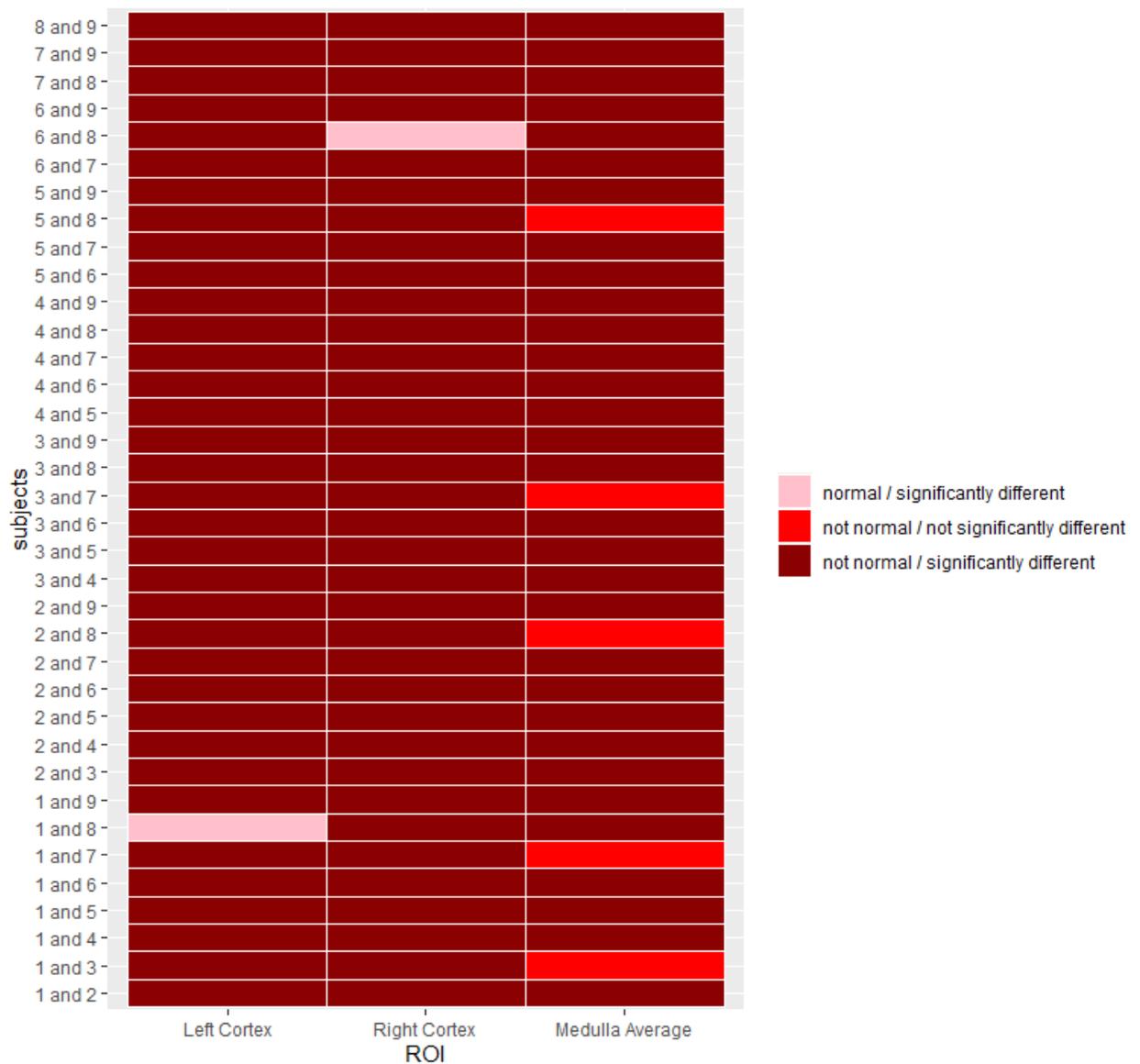


Figure 18: AIC score results from Shapiro-Wilk's test and either paired t-test or paired Wilcoxon signed rank test with 2TCM. The first part shows if the AIC scores were both normally distributed or not. "not normal" means that at least one of the two subjects was not normal. The second part tells if the distributions of AIC scores are significantly different between the two subjects.

Most of the subjects' AIC scores seemed not to be normally distributed with either 1TCM or 2TCM according to the results of Shapiro-Wilk's test. However, there seems to be more normally distributed AIC scores in the Left Cortex and Right Cortex than in the Medulla Average with both 1TCM and 2TCM. The distributions of AIC scores, with both 1TCM and 2TCM, seemed to differ for most of the subject pairs. On the other hand, it looks like Medulla Average has more distributions not significantly different with 2TCM than with 1TCM.

The relationship between delay and V_B values is one of the main goals of interest in this study. In the Figures 19, 20 and 21 the delays and V_B values of all the subjects are presented within the respective ROIs. Note that the delay boundaries in the fittings were -8 and 8 seconds as prescribed in Section 6.

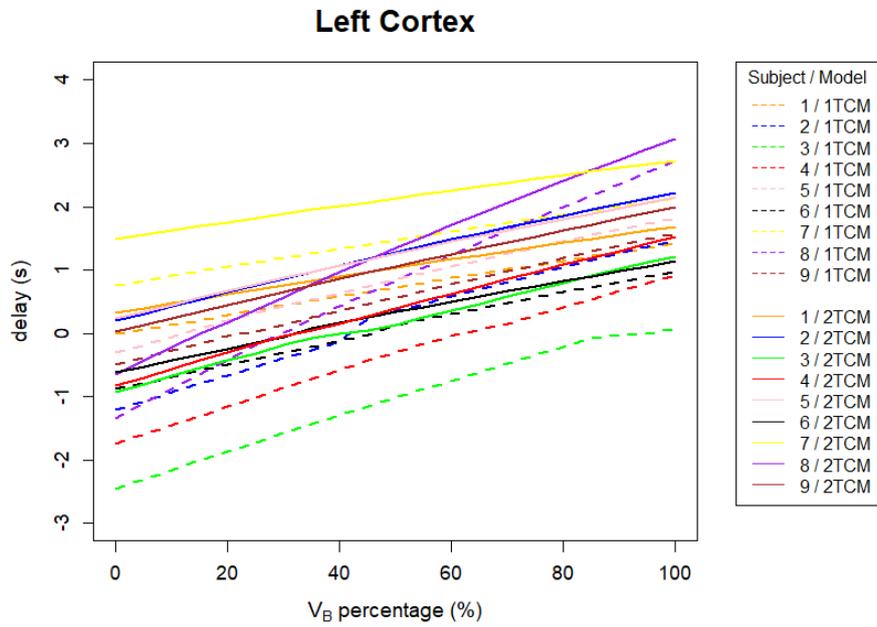


Figure 19: Delay and V_B comparison for all subjects with 1TCM and 2TCM in Left Cortex.

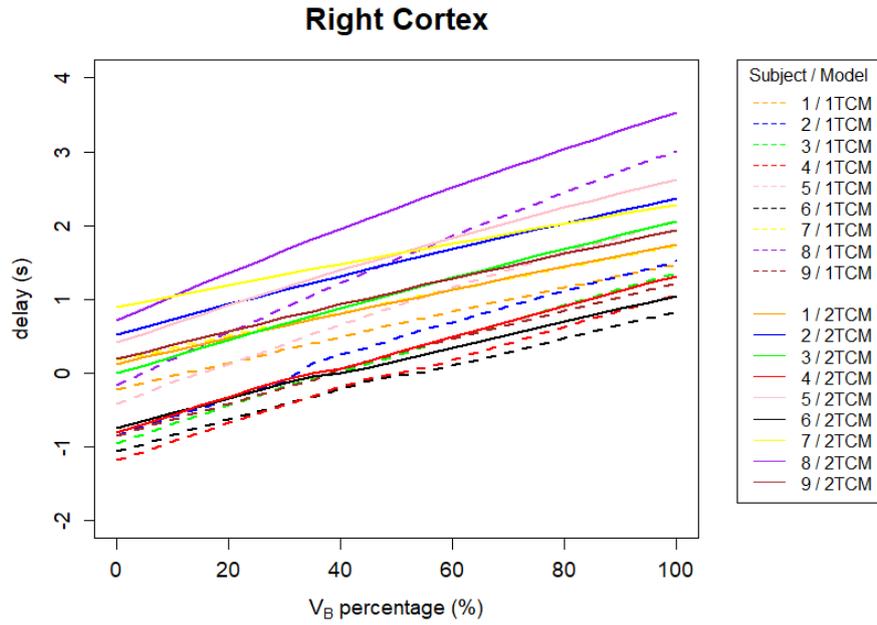


Figure 20: Delay and V_B comparison for all subjects with 1TCM and 2TCM in Right Cortex.

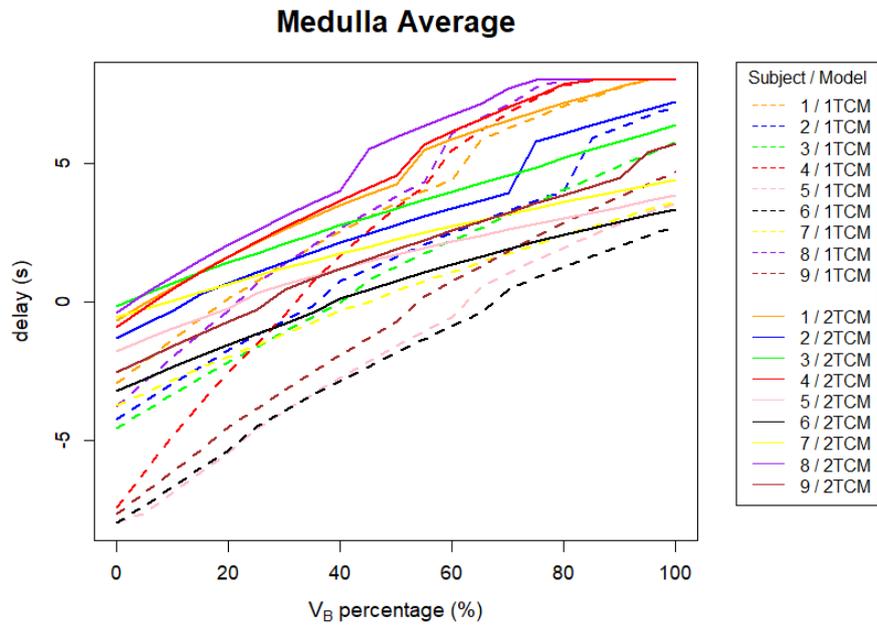


Figure 21: Delay and V_B comparison for all subjects with 1TCM and 2TCM in Medulla Average.

The delay increases with the V_B value in all the ROIs and for all the subjects. There seems to be a clear connection between delay and V_B . The delay in fittings with 1TCM are smaller than with 2TCM in all ROIs for every subject.

In order to examine the relationship between delay and V_B value even further, a correlation was measured with Kendall's τ . Figure 22 shows the Kendall's τ coefficients between delay and V_B with both 1TCM and 2TCM in each ROI.

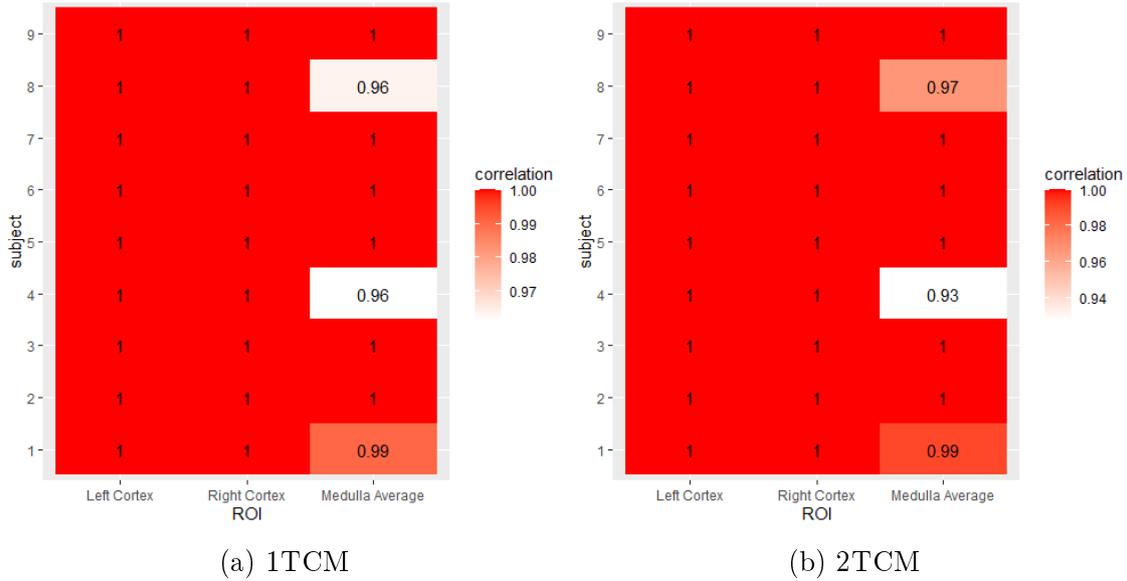


Figure 22: Kendall's τ coefficients between delay and V_B value with 1TCM and 2TCM in each ROI. All p-values from the correlation test were below significance level of 0.01 so all the correlations are statistically significant.

All the p-values in the correlation test were statistically significant with significance level of 0.01. There seems to be perfect correlation between delay and V_B value in the Left Cortex and Right Cortex for every subject and with both 1TCM and 2TCM since the Kendall's τ coefficient is 1 in all cases. The correlations in Medulla Average seem to vary more between subjects, but all Kendall's τ coefficients are still close to 1 so there seems to be strong correlation between delay and V_B value.

An overall correlation was also calculated with the mean of delays from all subjects. The result was that Kendall's τ coefficients were all 1, which seems to indicate strong correlation between delay and V_B value. The overall correlation tests for all ROIs and both 1TCM and 2TCM were all statistically significant with significance level of 0.01.

In order to get an average of delay for every ROI the fitting results were gathered and the fittings with best AIC scores were pinpointed. There were many cases that the AIC scores didn't fluctuate much between the V_B values, for example in the Left Cortex and Right Cortex. So instead of choosing the delay, the V_B value and RSE of delay from the fitting with the minimum AIC score, the delays and V_B values were gathered from a fitting with the average V_B value out of all fittings at maximum 1 AIC score point away from the smallest AIC score in the ROI in question. The results of this procedure are presented in the following tables.

Table 9: Average V_B percentages from fittings at maximum 1 AIC score point away from the smallest AIC score. ("LC" = Left Cortex, "RC" = Right Cortex and "MA" = Medulla Average)

subject	1TCM			2TCM		
	LC	RC	MA	LC	RC	MA
1	50 %	50 %	75 %	50 %	50 %	15 %
2	90 %	90 %	100 %	85 %	95 %	25 %
3	90 %	50 %	100 %	65 %	50 %	90 %
4	95 %	50 %	85 %	70 %	30 %	10 %
5	70 %	40 %	100 %	50 %	25 %	50 %
6	50 %	85 %	100 %	50 %	80 %	15 %
7	50 %	80 %	100 %	50 %	80 %	30 %
8	90 %	80 %	80 %	95 %	65 %	40 %
9	50 %	50 %	100 %	50 %	50 %	100 %
average	71%	64%	93%	63 %	58 %	42 %

In Table 9, fittings of lowest AIC scores with 1TCM give larger V_B values than with 2TCM.

Table 10: Average delays in seconds from fittings at maximum 1 AIC score point away from the smallest AIC score. ("LC" = Left Cortex, "RC" = Right Cortex and "MA" = Medulla Average)

subject	1TCM			2TCM		
	LC	RC	MA	LC	RC	MA
1	0.73	0.67	6.68	1.03	0.97	1.09
2	1.26	1.32	7.07	1.95	2.28	1.07
3	-0.02	0.24	5.76	0.47	1.09	5.79
4	2.54	1.54	8.00	2.07	1.66	0.92
5	1.26	0.66	8.00	1.26	1.04	5.95
6	0.14	0.57	2.73	0.34	0.70	-1.94
7	1.47	1.43	3.61	2.14	2.03	1.21
8	0.67	0.63	7.84	1.42	0.60	3.66
9	0.57	0.27	4.66	1.05	1.11	5.72
average	0.96	0.81	6.04	1.30	1.28	2.61

We can see from Table 10, that fittings of lowest AIC scores with 1TCM give larger delay values for Left Cortex and Right Cortex and smaller delay values for Medulla Average than with 2TCM.

Table 11: Average RSE of delay in seconds from fittings at maximum 1 AIC score point away from the smallest AIC score. ("LC" = Left Cortex, "RC" = Right Cortex and "MA" = Medulla Average)

subject	1TCM			2TCM		
	LC	RC	MA	LC	RC	MA
1	5.75	4.98	0.86	5.66	4.87	4.22
2	2.80	2.87	0.56	2.48	1.63	3.12
3	3.61	7.84	1.09	5.17	6.54	0.99
4	1.22	3.34	0.78	1.56	4.80	7.73
5	4.36	5.90	0.97	6.70	8.83	2.79
6	6.00	2.59	1.15	6.59	2.95	5.16
7	6.75	3.89	1.19	4.71	0.00	3.70
8	2.75	2.59	0.52	1.88	3.27	0.84
9	6.54	8.95	0.86	6.19	6.86	0.99
average	4.42	4.77	0.89	4.55	4.42	3.28

Table 11 shows that RSEs of delay in Left Cortex and Right Cortex are not so different between 1TCM or 2TCM. However, RSE of delay is smaller with 1TCM than with 2TCM for Medulla Average.

These average V_B values and delays from Tables 9 and 10 were fixed in new fittings for 1TCM and 2TCM with irreversible factor. The fitted values from these fittings are shown in Tables 12 and 13.

Table 12: Fittings with fixed delay and V_B value with 1TCM. ("LC" = Left Cortex, "RC" = Right Cortex and "MA" = Medulla Average)

	delay	V_B	% from V_A	K_1		k_2	
				mean	sd	mean	sd
LC	0.96	0.10	71 %	3.07	0.65	3.65	0.72
RC	0.81	0.09	64 %	3.00	0.38	3.80	0.38
MA	6.04	0.22	93 %	0.35	0.14	0.85	0.35
				AIC		RMSE	
				mean	sd	mean	sd
LC				620.16	0.21	2790.24	1039.97
RC				612.61	17.92	2387.26	631.06
MA				622.51	22.87	2833.80	915.77

Compared to the minimum AIC scores in Tables 3, 4 and 5, the AIC scores in Table 12 are bigger or about equal for all ROIs with 1TCM. The RMSE values also increased in all ROIs compared to the fittings with fixed V_B values.

Table 13: Fittings with fixed delay and V_B value with 2TCM. ("LC" = Left Cortex, "RC" = Right Cortex and "MA" = Medulla Average)

	delay	V_B	% from V_A	K_1		k_2		k_3	
				mean	sd	mean	sd	mean	sd
LC	1.30	0.09	63 %	3.39	0.69	4.17	0.77	0.0290	0.01
RC	1.28	0.08	58 %	3.32	0.40	4.37	0.45	0.0282	0.01
MA	2.61	0.10	42 %	1.04	0.28	3.00	0.65	0.1559	0.04
				AIC		RMSE			
				mean	sd	mean	sd		
LC				608.47	22.66	2138.27	564.58		
RC				600.60	18.57	1876.66	495.72		
MA				589.56	14.58	1564.28	322.36		

Compared to the minimum AIC scores in Tables 6, 7 and 8, the AIC scores in Table 13 are bigger or equal for all ROIs with 2TCM. The RMSE values increased in all ROIs compared to the fittings with fixed V_B values. This might suggest that the method of determining the average delays and V_B values from the range of multiple fittings according to small differences in AIC scores and fixing the average values in the new fittings might not yield significant information.

The V_B values with smallest AIC scores were also deduced from Tables 3, 4, 5, 6, 7 and 8 and from Figure 16. To test the validity of these fittings and to find the proper delays and V_B values, the fittings were done again with fixed delays that matched the ones from fittings with smallest AIC scores and RMSE values in Tables 3, 4, 5, 6, 7 and 8. The V_B value was allowed to vary freely with other variables and only the delays were fixed. Tables 14 and 15 show the comparison with the earlier fittings with fixed V_B values that had the smallest AIC scores and the new fittings with fixed delays derived from the earlier fittings.

Table 14: Comparison of fittings with 1TCM. The earlier fittings had fixed V_B values and the new fittings used the fitted delays from earlier fittings as fixed values.

	delay		V_B					
delay fixed	mean	sd	mean	sd	% from V_A			
Left Cortex	1.46		0.15	0.06	111 %			
Right Cortex	1.58		0.15	0.04	106 %			
Medulla Average	5.18		0.23	0.04	94 %			
V_B fixed								
Left Cortex	1.46	0.72	0.14		100 %			
Right Cortex	1.58	0.61	0.14		100 %			
Medulla Average	5.18	2.26	0.22		90 %			
	K_1		k_2		AIC		RMSE	
delay fixed	mean	sd	mean	sd	mean	sd	mean	sd
Left Cortex	2.74	0.73	3.45	0.79	617.28	25.49	2643.63	893.35
Right Cortex	2.68	0.31	3.66	0.31	610.24	18.74	2311.14	642.40
Medulla Average	0.33	0.07	0.81	0.27	611.18	21.81	2372.39	715.76
V_B fixed								
Left Cortex	2.86	0.67	3.56	0.78	619.77	26.55	2674.11	938.85
Right Cortex	2.75	0.34	3.72	0.34	612.79	18.69	2327.92	638.89
Medulla Average	0.36	0.13	0.85	0.32	615.09	22.94	2457.84	808.29

Table 14 shows that the AIC scores and RMSE values are lower in the new fittings with fixed delays with 1TCM. However, the fittings give V_B values higher than V_A in Left Cortex and Right Cortex.

Table 15: Comparison of fittings with 2TCM with irreversible factor. The earlier fittings had fixed V_B values and the new fittings used the fitted delays from earlier fittings as fixed values.

	delay		V_B		K_1			
delay fixed	mean	sd	mean	sd	% from V_A		mean	sd
Left Cortex	1.97		0.15	0.05	108 %		3.00	0.69
Right Cortex	2.10		0.14	0.05	101 %		3.00	0.33
Medulla Average	1.49		0.08	0.03	32 %		1.11	0.18
V_B fixed								
Left Cortex	1.97	0.62	0.14		100 %		3.10	0.68
Right Cortex	2.10	0.69	0.14		100 %		3.00	0.33
Medulla Average	1.49	1.76	0.07		30 %		1.13	0.21
	k_2		k_3		AIC		RMSE	
delay fixed	mean	sd	mean	sd	mean	sd	mean	sd
Left Cortex	3.97	0.79	0.03	0.02	605.22	21.68	2026.79	511.23
Right Cortex	4.28	0.47	0.03	0.01	597.37	19.32	1792.49	495.51
Medulla Average	3.01	0.44	0.14	0.04	584.90	15.97	1463.76	323.25
V_B fixed								
Left Cortex	4.05	0.81	0.03	0.01	607.73	21.79	2040.42	516.20
Right Cortex	4.28	0.47	0.03	0.01	600.09	18.67	1806.97	484.73
Medulla Average	3.06	0.45	0.15	0.04	586.73	15.47	1457.83	313.34

Table 15 shows that the AIC scores and RMSE values are lower in the new fittings with fixed delays with 2TCM. The only exception is Medulla Average for which the RMSE value increases in the fittings with fixed delays. Here again, the V_B values in Left Cortex and Right Cortex exceed the V_A value.

9 Conclusions

2TCM with irreversible factor seems to be more adequate model than 1TCM to model any ROI used in this study. This is clear according to AIC scores and RMSE values. The configuration with irreversible factor where the tracer gets trapped in the second tissue compartment models Left Cortex, Right Cortex and Medulla Average better than the conventional model.

The most suitable percentages of blood volume V_B seemed to be 100% out of V_A for Left Cortex and Right Cortex in both models and about 90% and 30% out of V_A for Medulla Average with 1TCM and 2TCM with irreversible factor respectively.

There seemed to be clear relationship between delay and V_B value with both 1TCM and 2TCM with irreversible factor: delay increased when V_B value increased. High correlation was also found in a test with Kendall's τ coefficient.

The fittings with average delays and V_B values had bigger AIC scores than the fittings with fixed V_B values. Therefore, they did not give significant information about the proper delays and V_B values.

The corresponding delays from the fittings with the most suitable V_B values would be 1,46 s for Left Cortex, 1,58 s for Right Cortex and 5,18 s for Medulla Average with 1TCM. The equivalent delays with 2TCM with irreversible factor would be 1,97 s for Left Cortex, 2,10 s for Right Cortex and 1,49 s for Medulla Average.

References

- [1] Akaike, H. *Information theory and an extension of the maximum likelihood principle*
- [2] Fox, H., Bolton, B. *Mathematics for Engineering and Technologists*, 2002
- [3] Gelman, A. *Data Analysis Using Regression and Multilevel / Hierarchical Models*, 2007
- [4] Heino, R. *Laplace-muunnos ja sen soveltaminen differentiaaliyhtälöiden teoriaan*, Master's thesis, 2014
- [5] Tadashi Inaba, Masato Yamashita, Yoshio Kawase, Hisamitsu Nakahashi, Hiroki Watanabe *Quantitative Measurement of Renal Plasma Flow by Positron Emission Tomography with Oxygen-15 Water*, Tohoku Journal, Vol. 159, 1989
- [6] Hidehiro Iida, Iwao Kanno, Akira Takahashi, Shuichi Miura, Matsutaro Murakami, Kazuhiro Takahashi, Yukihiro Ono, Fumio Shishido, Atsushi Inugami, Noriaki Tomura, Shuichi Higano, Hideaki Fujita, Hiroshi Sasaki, Hiroyuki Nakamichi, Shigenori Mizusawa, Yasushi Kondo, Kazuo Uemura *Measurement of Absolute Myocardial Blood Flow with $H_2^{15}O$ and Dynamic Positron-Emission Tomography*, Circulation, Vol. 78, 1988
- [7] Khalil, M. *Basic Science of PET Imaging*, 2017
- [8] Matheson, G.J. *kinfitr: Reproducible PET Pharmacokinetic Modelling in R*, 2019, <https://doi.org/10.1101/755751>
- [9] Mathews, J., Howell, R. *Complex Analysis for Mathematics and Engineering, 3rd edition*, 1996
- [10] Oikonen, V. *The website of Turku PET Centre* <http://www.turkupetcentre.net/petanalysis/index.html>
- [11] Shapiro, S., Wilk, M. *An Analysis of Variance Test for Normality*, Biometrika, Vol. 52, 1965
- [12] Suominen, H. *Yleistettyyn lokeromalliin perustuva spektraalianalyysi positroniemissiotomografia-mallintamisessa*, Master's thesis, 2005
- [13] Thomas, N. *Renal Nursing*, Fourth Edition, John Wiley & Sons Incorporated, 2014
- [14] Tuisku, J. *Aktivaation aloitusajankohdan tilastollinen suunnittelu aivojen PET-aktivaatiotutkimuksissa*, Master's thesis, 2012
- [15] Wilcox, R. *Understanding and applying basic statistical methods using R*, 2017

10 Appendix

Here is the programming code with which the fittings and the results in this thesis have been calculated. The programming was done with programming language R using `kinfitr` package [8].

```
remotes::install_github("mathesong/kinfitr")
install.packages("tidyverse")
library(kinfitr)
library(tidyverse)
library(knitr)
library(cowplot)
library(mgcv)
install.packages("ggforce")
library(ggforce)
install.packages("Metrics")
library(Metrics)
install.packages("xlsx")
library(xlsx)
install.packages("stats")
library(stats)

modified_onetcm <- function(t_tac, tac, input, weights = NULL, inpshift = NULL, vB = NULL,
                           frameStartEnd = NULL, K1.start = 0.1, K1.lower = 0.0001, K1.upper = 0.5,
                           k2.start = 0.1, k2.lower = 0.0001, k2.upper = 0.5,
                           inpshift.start = 0, inpshift.lower = -0.5, inpshift.upper = 0.5,
                           vB.start = 0.05, vB.lower = 0.01, vB.upper = 0.1, multstart_iter = 1,
                           multstart_lower = NULL, multstart_upper = NULL, printvals = F) {

  # Tidying
  modeldata <- tidyinput_art(t_tac, tac, weights, frameStartEnd)
  modeldata <- as.list(modeldata)
  modeldata$input <- input
  # Parameters
  start <- c(K1 = K1.start, k2 = k2.start, inpshift = inpshift.start, vB = vB.start)
  lower <- c(K1 = K1.lower, k2 = k2.lower, inpshift = inpshift.lower, vB = vB.lower)
  upper <- c(K1 = K1.upper, k2 = k2.upper, inpshift = inpshift.upper, vB = vB.upper)

  vB_fitted <- T
  if (!is.null(vB)) {
    vB_fitted <- F
    start[which(names(start) == "vB")] <- vB
    lower[which(names(lower) == "vB")] <- vB
    upper[which(names(upper) == "vB")] <- vB
  }
  multstart_pars <- fix_multstartpars(start, lower, upper, multstart_iter,
                                     multstart_lower, multstart_upper)
  multstart_upper <- multstart_pars$multstart_upper
  multstart_lower <- multstart_pars$multstart_lower

  # Solution - Fitting the Delay
  if (is.null(inpshift)) {
    inpshift_fitted <- T
    if (prod(multstart_iter) == 1) {
      output <- minpack.lm::nlsLM(
        tac ~ my_onetcmDelay_model(t_tac, input, K1, k2, inpshift, vB),
        data = modeldata, start = start, lower = lower, upper = upper, weights = weights,
        control = minpack.lm::nls.lm.control(maxiter = 200), trace = printvals)
    } else {
      output <- nls.multstart::nls_multstart(
        tac ~ my_onetcmDelay_model(t_tac, input, K1, k2, inpshift, vB), data = modeldata,
        supp_errors = "Y", start_lower = multstart_lower, start_upper = multstart_upper,
        iter = multstart_iter, convergence_count = FALSE,
        lower = lower, upper = upper, modelweights = weights)
    }
  }
  # Check for parameters hitting limits
  limcheck_u <- purrr::map2_lgl(round(upper, 3), round(coef(output), 3), identical)
  limcheck_l <- purrr::map2_lgl(round(lower, 3), round(coef(output), 3), identical)
  limcheck <- limcheck_u + limcheck_l
}
```

```

limcheck <- limcheck==1
if(any(limcheck)) {
  warning(paste0("\nFitted parameters are hitting upper or lower limit bounds. Consider\n",
    "either modifying the upper and lower limit boundaries, or else using\n",
    "multstart when fitting the model (see the function documentation).\n") )
}
# Output
if (inpshift_fitted == T) {
  newvals <- shift_timings(modeldata$t_tac, modeldata$tac, modeldata$input,
    as.numeric(coef(output)[["inpshift"]]))
  tacs <- data.frame(Time = newvals$t_tac, Target = newvals$tac,
    Target_fitted = as.numeric(fitted(output)))
  input <- newvals$input
} else {
  tacs <- data.frame(Time = newvals$t_tac, Target = newvals$tac,
    Target_fitted = as.numeric(fitted(output)))
  input <- newvals$input
}
par <- as.data.frame(as.list(coef(output)))

if (inpshift_fitted == F) par$inpshift <- inpshift
par.se <- par
par.se[1,] <- purrr::map_dbl(names(par), ~ get_se(output, .x))
names(par.se) <- paste0(names(par.se), ".se")
par$Vt <- par$K1 / par$k2
par.se$Vt.se <- get_se(output, "K1/k2")

out <- list(par = par, par.se = par.se, fit = output, tacs = tacs, input = input,
  weights = modeldata$weights, inpshift_fitted = inpshift_fitted,
  vB_fitted = vB_fitted, model = "itcm")
class(out) <- c("itcm", "kinfit")
return(out)
}

modified_onetcm_fitDelay_model <- function(t_tac, input, K1, k2, inpshift, vB) {
  # Using ones instead of tac as don't need it
  newvals <- shift_timings(t_tac, rep(1, length(t_tac)), input, inpshift)
  t_tac <- newvals$t_tac
  aif <- newvals$input$AIF
  interptime <- newvals$input$Time
  step <- interptime[2] - interptime[1]
  a <- K1 * exp(-k2 * interptime)
  b <- aif

  i_outtac <- kinfit_convolve(a, b, step)
  # Correction for vB
  i_outtac <- i_outtac + vB * aif
  outtac <- pracma::interp1(interptime, i_outtac, t_tac)
  return(outtac)
}

modified_twotcm <- function(t_tac, tac, input, weights = NULL, inpshift = NULL, vB = NULL,
  frameStartEnd = NULL, K1.start = 0.1, K1.lower = 0.0001, K1.upper = 0.5,
  k2.start = 0.1, k2.lower = 0.0001, k2.upper = 0.5, k3.start = 0.1,
  k3.lower = 0.0001, k3.upper = 0.5, k4.start = 0.1, k4.lower = 0.0001,
  k4.upper = 0.5, inpshift.start = 0, inpshift.lower = -0.5,
  inpshift.upper = 0.5, vB.start = 0.05, vB.lower = 0.01, vB.upper = 0.1,
  multstart_iter = 1, multstart_lower = NULL, multstart_upper = NULL,
  printvals = F) {
  # Tidying
  tidyinput <- tidyinput_art(t_tac, tac, weights, frameStartEnd)
  modeldata <- list(t_tac = tidyinput$t_tac, tac = tidyinput$tac,
    weights = tidyinput$weights, input = input)
  # Parameters
  start <- c(K1 = K1.start, k2 = k2.start, k3 = k3.start, k4 = k4.start,
    inpshift = inpshift.start, vB = vB.start)
  lower <- c(K1 = K1.lower, k2 = k2.lower, k3 = k3.lower, k4 = k4.lower,
    inpshift = inpshift.lower, vB = vB.lower)
  upper <- c(K1 = K1.upper, k2 = k2.upper, k3 = k3.upper, k4 = k4.upper,
    inpshift = inpshift.upper, vB = vB.upper)

```

```

vB_fitted <- T
if (!is.null(vB)) {
  vB_fitted <- F
  start[which(names(start) == "vB")] <- vB
  lower[which(names(lower) == "vB")] <- vB
  upper[which(names(upper) == "vB")] <- vB
}
multstart_pars <- fix_multstartpars(start, lower, upper, multstart_iter,
                                   multstart_lower, multstart_upper)
multstart_upper <- multstart_pars$multstart_upper
multstart_lower <- multstart_pars$multstart_lower

# Solution - Fitting the Delay
if (is.null(inpshift)) {
  inpshift_fitted <- T
  if (prod(multstart_iter) == 1) {
    output <- minpack.lm::nlsLM(
      tac ~ my_twotcm_fitDelay_model(t_tac, input, K1, k2, k3, k4, inpshift, vB),
      data = modeldata, start = start, lower = lower, upper = upper, weights = weights,
      control = minpack.lm::nls.lm.control(maxiter = 200), trace = printvals)
  } else {
    output <- nls.multstart::nls_multstart(
      tac ~ my_twotcm_fitDelay_model(t_tac, input, K1, k2, k3, k4, inpshift, vB),
      data = modeldata, supp_errors = "Y", start_lower = multstart_lower,
      start_upper = multstart_upper, iter = multstart_iter, convergence_count = FALSE,
      lower = lower, upper = upper, modelweights = weights)
  }
}
# Check for parameters hitting limits
limcheck_u <- purrr::map2_lgl(round(upper,3), round(coef(output),3), identical)
limcheck_l <- purrr::map2_lgl(round(lower,3), round(coef(output),3), identical)
limcheck <- limcheck_u + limcheck_l
limcheck <- limcheck==1

if(any(limcheck)) {
  warning(
    paste0(
      "\nFitted parameters are hitting upper or lower limit bounds. Consider\n",
      "either modifying the upper and lower limit boundaries, or else using\n",
      "multstart when fitting the model (see the function documentation).\n") )
}
# Output
if (inpshift_fitted == T) {
  newvals <- shift_timings(modeldata$t_tac, modeldata$tac, modeldata$input,
                          as.numeric(coef(output)[["inpshift"]]))
  tacs <- data.frame(Time = newvals$t_tac, Target = newvals$tac,
                    Target_fitted = as.numeric(fitted(output)))
  input <- newvals$input
} else {
  tacs <- data.frame(Time = newvals$t_tac, Target = newvals$tac,
                    Target_fitted = as.numeric(fitted(output)))
  input <- newvals$input
}
par <- as.data.frame(as.list(coef(output)))
if (inpshift_fitted == F) par$inpshift <- inpshift
par.se <- par
par.se[1,] <- purrr::map_dbl(names(par), ~ get_se(output, .x))
names(par.se) <- paste0(names(par.se), ".se")
par$Vt <- (par$K1 / par$k2) * (1 + par$k3 / par$k4)
par.se$Vt.se <- get_se(output, "(K1/k2)*_*(1+(k3/k4))")

out <- list(par = par, par.se = par.se, fit = output, tacs = tacs, input = input,
           weights = modeldata$weights, inpshift_fitted = inpshift_fitted,
           vB_fitted = vB_fitted, model = "2tcm")
class(out) <- c("2tcm", "kinfit")
return(out)
}

modified_twotcm_fitDelay_model <- function(t_tac, input, K1, k2, k3, k4, inpshift, vB) {

```

```

# Using ones instead of tac as don't need it
newvals <- shift_timings(t_tac, rep(1, length(t_tac)), input, inpshift)
t_tac <- newvals$t_tac
aif <- newvals$input$AIF
interptime <- newvals$input$Time
step <- interptime[2] - interptime[1]

delta <- sqrt((k2 + k3 + k4)^2 - 4 * k2 * k4)
th1 <- (k2 + k3 + k4 + delta) / 2
th2 <- (k2 + k3 + k4 - delta) / 2
ph1 <- K1 * (th1 - k3 - k4) / delta
ph2 <- K1 * (th2 - k3 - k4) / (-delta)

a <- ph1 * exp(-th1 * interptime) + ph2 * exp(-th2 * interptime)
b <- aif
i_outtac <- kinfit_convolve(a, b, step)
# Correction for vB
i_outtac <- i_outtac + vB * aif
outtac <- pracma::interp1(interptime, i_outtac, t_tac)
return(outtac)
}

#the calculated VB values from C150 images for left cortex,
#right cortex and medulla average (from Excel)
vb1<-c(0.113342919, 0.119335195,(0.264526748+0.273717841+0.257051541+0.256186976+
0.244070492)/5)

#the VB percentages
vbco<-seq(from = 0, to=1, by = 0.05)
#measured activities
ROI_act<-list(as.numeric(subject1_act$Cortex_left),as.numeric(subject1_act$Cortex_right),
mean(c(as.numeric(subject1_act$Medulla_1),as.numeric(subject1_act$Medulla_2),
as.numeric(subject1_act$Medulla_3),as.numeric(subject1_act$Medulla_4),
as.numeric(subject1_act$Medulla_5))))
aorta<-as.numeric(subject1_act$Aorta)
time_points<-subject1_act$.1/60
b<-blood_interp(t_blood = t,blood=rep(0,length(t)),t_plasma = t,
plasma=aorta,t_parentfrac = t,parentfrac = rep(1,length(t)))
#the fittings of 1TCM and 2TCM for all VB values
for (j in 1:length(vbco)) {
act<-list()
act_values<-list()
act2<-list()
act2_values<-list()
act_e<-list()
act2_e<-list()
for (i in 1:length(y)) {
final_vb<-vb1[i]*vbco[j]
#1TCM
fit1<-modified_onetcm(time_points,ROI_act[[i]],b,vB=final_vb,K1.start = 3,K1.upper = 10,
k2.start = 3,k2.upper = 10,inpshift.start = 5/60,
inpshift.lower = -8/60,inpshift.upper = 8/60,
multstart_iter = c(K1=3,k2=3,inpshift=1,vB=1),
multstart_lower = list(K1=0.01,k2=0.01,inpshift=-8/60,vB=final_vb),
multstart_upper = list(K1=10,k2=10,inpshift=8/60,vB=final_vb))
act[[i]]<-modified_onetcm_fitDelay_model(time_points,b,K1= fit1$par$K1,k2=fit1$par$k2,
inpshift=fit1$par$inpshift,vB=fit1$par$vB)
#RMSE
act_e[i]<-rmse(ROI_act[[i]],act[[i]])
#residual standard error
d_std<-summary(fit1$fit)$parameters[3,2]
#gather fitted and calculated values to list
act_values[[i]]<-list(k1=fit1$par$K1,k2=fit1$par$k2,vB=fit1$par$vB,
inpshift=round((fit1$par$inpshift)*60,2),
error=act_e[i],aic=AIC(fit1$fit),delay_std=round(d_std*60,2))
#2TCM
fit2<-modified_twtotcm(time_points,ROI_act[[i]],b,vB=final_vb,K1.start = 3,K1.upper = 10,
k2.start = 3,k2.upper = 10,k3.start=3,k3.upper=10,k4.start=0,
k4.lower=0,k4.upper=0,inpshift.start = 5/60,inpshift.lower = -8/60,
inpshift.upper = 8/60,
multstart_iter = c(K1=3,k2=3,k3=3,k4=1,inpshift=1,vB=1),

```

```

        multstart_lower = list(K1=0.01,k2=0.01,k3=0.01,
                               k4=0,inpshift=1/60,vB=final_vb),
        multstart_upper = list(K1=10,k2=10,k3=10,k4=0,inpshift=9/60,vB=final_vb))
act2[[i]]<-modified_twotcm_fitDelay_model(time_points,b,K1=fit2$par$K1,k2=fit2$par$k2,
                                           k3 = fit2$par$k3,k4 = fit2$par$k4,
                                           inpshift = fit2$par$inpshift,vB=fit2$par$vB)

#RMSE
act2_e[i]<-rmse(ROI_act[[i]],act2[[i]])
#standard error
d_std2<-summary(fit2$fit)$parameters[5,2]
#gather fitted and calculated values to list
act2_values[[i]]<-c(k1= fit2$par$K1, k2=fit2$par$k2,k3=fit2$par$k3,vB=fit2$par$vB,
                    inpshift=round((fit2$par$inpshift)*60,2),error=act2_e[i],
                    aic=AIC(fit2$fit),delay_std=round(d_std2*60,2))
}
value_table<-as.data.frame(rbind(act_values[[1]],act_values[[2]],act_values[[3]]),
                            row.names = c("Cortex_left","Cortex_right","Medulla_average"))
value_table2<-as.data.frame(rbind(act2_values[[1]],act2_values[[2]],act2_values[[3]]),
                             row.names = c("Cortex_left","Cortex_right","Medulla_average"))

#save in Excel file
if (j==1){write.xlsx(value_table,"VB_1(0).xlsx")
           write.xlsx(value_table2,"VB_1(0.0)2.xlsx")}
if (j==2){write.xlsx(value_table,"VB_1(0.1).xlsx")
           write.xlsx(value_table2,"VB_1(0.1)2.xlsx")}
if (j==3){write.xlsx(value_table,"VB_1(0.2).xlsx")
           write.xlsx(value_table2,"VB_1(0.2)2.xlsx")}
if (j==4){write.xlsx(value_table,"VB_1(0.3).xlsx")
           write.xlsx(value_table2,"VB_1(0.3)2.xlsx")}
if (j==5){write.xlsx(value_table,"VB_1(0.4).xlsx")
           write.xlsx(value_table2,"VB_1(0.4)2.xlsx")}
if (j==6){write.xlsx(value_table,"VB_1(0.5).xlsx")
           write.xlsx(value_table2,"VB_1(0.5)2.xlsx")}
if (j==7){write.xlsx(value_table,"VB_1(0.6).xlsx")
           write.xlsx(value_table2,"VB_1(0.6)2.xlsx")}
if (j==8){write.xlsx(value_table,"VB_1(0.7).xlsx")
           write.xlsx(value_table2,"VB_1(0.7)2.xlsx")}
if (j==9){write.xlsx(value_table,"VB_1(0.8).xlsx")
           write.xlsx(value_table2,"VB_1(0.8)2.xlsx")}
if (j==10){write.xlsx(value_table,"VB_1(0.9).xlsx")
            write.xlsx(value_table2,"VB_1(0.9)2.xlsx")}
if (j==11){write.xlsx(value_table,"VB_1(1).xlsx")
            write.xlsx(value_table2,"VB_1(1)2.xlsx")}
}

#gather all AIC scores to list (same for 2TCM)
aic_list<-aic(subject1_aic,subject2_aic,subject3_aic,subject4_aic,subject5_aic,
              subject6_aic,subject7_aic,subject8_aic,subject9_aic)

#shapiro-wilk's test (Subject 1 as an example here. Same for 2TCM)
subject1_shap_list<-list(shapiro.test(subject1_cl_aic)$p.value,
                        shapiro.test(subject1_cr_aic)$p.value,
                        shapiro.test(subject1_med_aic)$p.value)

#gather shapiro-wilk's tests for all subjects to list (same for 2TCM)
shap_list<-list(subject1_shap_list,subject2_shap_list,subject3_shap_list,
                subject4_shap_list,subject5_shap_list,subject6_shap_list,
                subject7_shap_list,subject8_shap_list,subject9_shap_list)

#Paired Student's t-tests and Wilcoxon tests for 1TCM (same for 2TCM)
result_list<-list()
result_vec<-vector()
l<-1
for (i in 1:(length(shap_list)-1)) {
  first_aic<-aic_list[[i]]
  first_shap<-shap_list[[i]]
  for (j in (i+1):length(shap_list)) {
    for (k in 1:3) {
      if(shap_list[[j]][[k]]>0.05 & first_shap[[k]]>0.05){
        paired_test<-t.test(unlist(first_aic[[k]]),
                           unlist(aic_list2[[j]][[k]]),paired=TRUE)

```

```

    if(paired_test$p.value<0.05){
      result_vec[k]<-paste0("normal_/significantly_different")
    }else{
      result_vec[k]<-paste0(
        "normal_/not_significantly_different")
    }
  }else{
    paired_test<-wilcox.test(unlist(first_aic[[k]]),
                             unlist(aic_list[[j]][[k]]),paired = TRUE)
    if(paired_test$p.value<0.05){
      result_vec[k]<-paste0(
        "not_normal_/significantly_different")
    }else{
      result_vec[k]<-paste0(
        "not_normal_/not_significantly_different")
    }
  }
}
}
result_list[[1]]<-result_vec
l<-l+1
}
}
results_table<-data.frame(matrix(unlist(result_list),nrow=length(result_list),byrow=TRUE))
new_table<-cbind(results_table,subjects=c("1_and_2","1_and_3","1_and_4","1_and_5",
                                          "1_and_6","1_and_7","1_and_8","1_and_9",
                                          "2_and_3","2_and_4","2_and_5","2_and_6",
                                          "2_and_7","2_and_8","2_and_9","3_and_4",
                                          "3_and_5","3_and_6","3_and_7","3_and_8",
                                          "3_and_9","4_and_5","4_and_6","4_and_7",
                                          "4_and_8","4_and_9","5_and_6","5_and_7",
                                          "5_and_8","5_and_9","6_and_7","6_and_8",
                                          "6_and_9","7_and_8","7_and_9","8_and_9"))
colnames(new_table)<-c("Left_Cortex","Right_Cortex","Medulla_Average","subjects")
final_table <- melt(new_table, id.var = "subjects")
ggplot(final_table,aes(variable, subjects)) + geom_tile(aes(fill = value),colour = "white")+
  xlab("ROI") + labs(fill = "") +
  scale_fill_manual(values=c("pink","red", "dark_red","black"))

#Kendall rank correlation test for delay and V_B (same for 2TCM)
corr<-vector()
corr_list<-list()
subj_list<-list(delay_subject1,delay_subject2,delay_subject3,delay_subject4,delay_subject5,
                delay_subject6,delay_subject7,delay_subject8,delay_subject9)
for (j in 1:9) {
  corr<-vector()
  for (i in 1:3) {
    #Kendall's tau coefficient
    corr_tau<-append(corr,cor.test(VB_plot,as.vector(t(subj_list[[j]][i])),
                                   method = "kendall")$estimate)

    #p-value
    corr_p<-append(corr,cor.test(VB_plot,as.vector(t(subj_list[[j]][i])),
                                   method = "kendall")$p.value)
  }
  corr_list_tau[[j]]<-corr_tau
  corr_list_p[[j]]<-corr_p
}
y <- c("1","2","3","4","5","6","7","8","9")
x <- c("Left_Cortex","Right_Cortex","Medulla_Average")
data <- expand.grid(ROI=x, subject=y)
data$correlation <- c(corr_list_k[[1]],corr_list_k[[2]],corr_list_k[[3]],
                    corr_list_k[[4]],corr_list_k[[5]],corr_list_k[[6]],
                    corr_list_k[[7]],corr_list_k[[8]],corr_list_k[[9]])
ggplot(data, aes(ROI,subject, fill=correlation )) + geom_tile() +
  geom_text(aes(label = round(correlation,2))) +
  scale_fill_gradient(low = "white",high = "red")

#Kendall's tau coefficients and p-value for overall Kendall rank correlation (same for 2TCM)
left_cortex<-mean(c(as.vector(t(subj_list[[1]][1])),as.vector(t(subj_list[[2]][1])),
                  as.vector(t(subj_list[[3]][1])),as.vector(t(subj_list[[4]][1])),
                  as.vector(t(subj_list[[5]][1])),as.vector(t(subj_list[[6]][1])),

```

```

as.vector(t(subj_list[[7]][1,])), as.vector(t(subj_list[[8]][1,])),
as.vector(t(subj_list[[9]][1,])))
right_cortex<-mean(c(as.vector(t(subj_list[[1]][2,])), as.vector(t(subj_list[[2]][2,])),
as.vector(t(subj_list[[3]][2,])), as.vector(t(subj_list[[4]][2,])),
as.vector(t(subj_list[[5]][2,])), as.vector(t(subj_list[[6]][2,])),
as.vector(t(subj_list[[7]][2,])), as.vector(t(subj_list[[8]][2,])),
as.vector(t(subj_list[[9]][2,])))
medulla_average<-mean(c(as.vector(t(subj_list[[1]][3,])), as.vector(t(subj_list[[2]][3,])),
as.vector(t(subj_list[[3]][3,])), as.vector(t(subj_list[[4]][3,])),
as.vector(t(subj_list[[5]][3,])), as.vector(t(subj_list[[6]][3,])),
as.vector(t(subj_list[[7]][3,])), as.vector(t(subj_list[[8]][3,])),
as.vector(t(subj_list[[9]][3,])))
corr_tau_cl<-cor.test(VB_plot, left_cortex, method = "kendall")$estimate
corr_tau_cr<-cor.test(VB_plot, right_cortex, method = "kendall")$estimate
corr_tau_med<-cor.test(VB_plot, medulla_average, method = "kendall")$estimate
corr_p_cl<-cor.test(VB_plot, left_cortex, method = "kendall")$p.value
corr_p_cr<-cor.test(VB_plot, right_cortex, method = "kendall")$p.value
corr_p_med<-cor.test(VB_plot, medulla_average, method = "kendall")$p.value

#The new fitting with average VB and delay values fixed

#VB values
subj<-vb1
vb_list<-c(subj[1]*0.71, subj[1]*0.63, subj[2]*0.64, subj[2]*0.58, subj[3]*0.93, subj[3]*0.42)

#delay
delay_list<-c(0.96/60, 1.3/60, 0.81/60, 1.28/60, 6.04/60, 2.61/60)

#measured activities, aorta activity, time points and input
ROI_act<-list(as.numeric(subject1_act$Cortex_left), as.numeric(subject1_act$Cortex_right),
(as.numeric(subject1_act$Medulla_1)+as.numeric(subject1_act$Medulla_2)+
as.numeric(subject1_act$Medulla_3)+as.numeric(subject1_act$Medulla_4)+
as.numeric(subject1_act$Medulla_5))/5)
aorta<-as.numeric(subject1_act$Aorta)
time_points<-subject1$...1/60
b<-blood_interp(t_blood = t, blood=rep(0, length(t)), t_plasma = t, plasma=aorta,
t_parentfrac = t, parentfrac = rep(1, length(t)))
k<-1
act<-list()
act_co<-list()
act2<-list()
act2_co<-list()
act_e<-list()
act2_e<-list()
for (i in 1:length(y)) {
final_vb<-vb_list[k]
dt<-dt_list[k]
k<-k+1
#1TCM
fit1<-modified_onetcm(time_points, ROI_act[[i]], b, vB = final_vb, inpshift = dt, K1.start = 3,
K1.upper = 10, k2.start = 3, k2.upper = 10, multstart_iter = 1)
act[[i]]<-modified_onetcm_fitDelay_model(time_points, b, K1= fit1$par$K1, k2=fit1$par$k2,
inpshift=fit1$par$inpshift, vB=fit1$par$vB)
#RMSE
act_e[i]<-rmse(ROI_act[[i]], act[[i]])
#gather fitted and calculated values to list
act_values[[i]]<-list(k1=fit1$par$K1, k2=fit1$par$k2, vB=fit1$par$vB,
inpshift=round((fit1$par$inpshift)*60, 2), error=act_e[i],
aic=AIC(fit1$fit))
final_vb<-vb_list[k]
dt<-dt_list[k]
k<-k+1
#2TCM
fit2<-modified_twotcm(time_points, ROI_act[[i]], b, vB = final_vb, inpshift = dt, K1.start = 3,
K1.upper = 10, k2.start = 2, k2.upper = 7, k3.start=1, k3.upper=5,
k4.lower=0, k4.upper=0, vB.start = 0.15, vB.lower = 0, vB.upper = 0.5,
multstart_iter = 1)
act2[[i]]<-modified_twotcm_fitDelay_model(time_points, b, K1=fit2$par$K1, k2= fit2$par$k2,

```

```

k3 = fit2$par$k3, k4 = fit2$par$k4,
inpshift = fit2$par$inpshift, vB=fit2$par$vB)

#RMSE
act2_e[i]<-rmse(ROI_act[[i]],act2[[i]])
#gather fitted and calculated values to list
act2_values[[i]]<-c(k1= fit2$par$K1, k2=fit2$par$k2, k3=fit2$par$k3, vB=fit2$par$vB,
inpshift=round((fit2$par$inpshift)*60,2),error=act2_e[i],aic=AIC(fit2$fit))
}
value_table<-as.data.frame(rbind(act_values[[1]],act_values[[2]],act_values[[3]]),
row.names = c("Cortex_left","Cortex_right","Medulla_average"))
value_table2<-as.data.frame(rbind(act2_values[[1]],act2_values[[2]],act2_values[[3]]),
row.names = c("Cortex_left","Cortex_right","Medulla_average"))
write.xlsx(table,"VB_dt_subject1.xlsx")
write.xlsx(table2,"VB_dt_subject12.xlsx")

#The new fitting with fixed delay values

#delay
delay_list<-c(1.46/60,1.97/60,1.58/60,2.10/60,5.18/60,1.49/60)

#measured activities, aorta activity, time points and input
ROI_act<-list(as.numeric(subject1_act$Cortex_left),as.numeric(subject1_act$Cortex_right),
(as.numeric(subject1_act$Medulla_1)+as.numeric(subject1_act$Medulla_2)+
as.numeric(subject1_act$Medulla_3)+as.numeric(subject1_act$Medulla_4)+
as.numeric(subject1_act$Medulla_5))/5)
aorta<-as.numeric(subject1_act$Aorta)
time_points<-subject1$...1/60
b<-blood_interp(t_blood = t,blood=rep(0,length(t)),t_plasma = t,plasma=aorta,
t_parentfrac = t,parentfrac = rep(1,length(t)))

k<-1
act<-list()
act_co<-list()
act2<-list()
act2_co<-list()
act_e<-list()
act2_e<-list()
for (i in 1:length(y)) {
dt<-dt_list[k]
k<-k+1
#1TCM
fit1<-modified_onetcm(time_points,ROI_act[[i]],b,inpshift = dt, K1.start = 3,
K1.upper = 10,k2.start = 3,k2.upper = 10,vB.start = 0.5,
vB.lower = 0,vB.upper = 1, multstart_iter = 1)
act[[i]]<-modified_onetcm_fitDelay_model(time_points, b,K1= fit1$par$K1, k2=fit1$par$k2,
inpshift=fit1$par$inpshift, vB=fit1$par$vB)

#RMSE
act_e[i]<-rmse(ROI_act[[i]],act[[i]])
#gather fitted and calculated values to list
act_values[[i]]<-list(k1=fit1$par$K1,k2=fit1$par$k2,vB=fit1$par$vB,
inpshift=round((fit1$par$inpshift)*60,2),error=act_e[i],
aic=AIC(fit1$fit))

dt<-dt_list[k]
k<-k+1
#2TCM
fit2<-modified_twotcm(time_points,ROI_act[[i]],b,inpshift = dt, K1.start = 3,
K1.upper = 10,k2.start = 2,k2.upper = 7,k3.start=1,k3.upper=5,
k4.lower=0,k4.upper=0,vB.start = 0.15,vB.lower = 0,vB.upper = 0.5,
vB.start = 0.5,vB.lower = 0,vB.upper = 1,multstart_iter = 1)
act2[[i]]<-modified_twotcm_fitDelay_model(time_points, b, K1=fit2$par$K1,k2= fit2$par$k2,
k3 = fit2$par$k3, k4 = fit2$par$k4,
inpshift = fit2$par$inpshift, vB=fit2$par$vB)

#RMSE
act2_e[i]<-rmse(ROI_act[[i]],act2[[i]])
#gather fitted and calculated values to list
act2_values[[i]]<-c(k1= fit2$par$K1, k2=fit2$par$k2, k3=fit2$par$k3, vB=fit2$par$vB,
inpshift=round((fit2$par$inpshift)*60,2),error=act2_e[i],aic=AIC(fit2$fit))
}
value_table<-as.data.frame(rbind(act_values[[1]],act_values[[2]],act_values[[3]]),
row.names = c("Cortex_left","Cortex_right","Medulla_average"))

```

```
value_table2<-as.data.frame(rbind(act2_values[[1]],act2_values[[2]],act2_values[[3]]),
                             row.names = c("Cortex_left","Cortex_right","Medulla_average"))
write.xlsx(table,"dt_subject1.xlsx")
write.xlsx(table2,"dt_subject12.xlsx")
```