

# [11C]PBR-28 Test-Retest Analysis: Ratio-Based Methods Evaluation

## Contents

<b>Aims</b>	<b>2</b>
<b>Libraries</b>	<b>2</b>
CRAN libraries . . . . .	2
Non-CRAN libraries . . . . .	2
<b>Demographic Data</b>	<b>2</b>
Summary Statistics . . . . .	3
<b>TACs and Blood Data</b>	<b>3</b>
<b>Kinetic Modelling</b>	<b>4</b>
Fitting of the Delay and Blood Volume Fraction . . . . .	4
Rearrangement of the Data into Long Format . . . . .	4
Define functions for fitting the models . . . . .	4
Fit all Kinetic Models . . . . .	5
<b>Calculate Ratio Outcomes</b>	<b>6</b>
<b>Evaluate Spread and Test-Retest Metrics</b>	<b>6</b>
HABs and MABs together . . . . .	6
Separated by Genotype . . . . .	7
HABs . . . . .	8
MABs . . . . .	8
Article Summary . . . . .	9
<b>Correlations with <math>V_T</math></b>	<b>10</b>
Plot of Ratio Associations with $V_T$ . . . . .	11
Plot Of Alternative Quantification Methods Associations with $V_T$ . . . . .	13
<b>Interregional Correlation</b>	<b>17</b>
<b>Principal Components Analysis</b>	<b>18</b>
PCA: All Regions . . . . .	19
PCA: Only FC, WB and CBL . . . . .	19
<b>Plotting of 2TCM Model Fits</b>	<b>20</b>
Included Participants . . . . .	20
Excluded Measurement . . . . .	43
<b>vB Checking</b>	<b>44</b>
<b>Session Info</b>	<b>46</b>

## Aims

The aims of this study were to evaluate the test-retest reliability, and validity (assessed by the relationship to  $V_T$ ) of the Standardised Uptake Value Ratio (SUVR) and Distribution Volume Ratio (DVR) of [11C]PBR28 in the frontal cortex (FC), using the whole brain (WB) and cerebellum (CBL) as reference regions (i.e. denominators). We also aim to examine the degree of association of binding estimates between regions of the brain.

## Libraries

### CRAN libraries

First, the libraries for the analysis and plotting are loaded.

```
library(tidyverse)
library(stringr)
library(corrplot)
library(grid)
library(gridExtra)
library(RColorBrewer)
library(psych)
library(readxl)
library(pracma)
```

### Non-CRAN libraries

The libraries above can be installed from CRAN. Those which cannot are installed as follows:

```
# install.packages("devtools") # If you do not already have devtools
# devtools::install_github("mathesong/kinfitr", ref="v0.2.0")
# devtools::install_github("mathesong/granviller")
# devtools::install_github("mvuorre/vmisc")
# devtools::install_github('Rapporter/pander')

library(kinfitr)
library(vmisc)
library(pander)
```

## Demographic Data

Here, the demographic data is loaded in.

```
demog <- read_excel('../RawData/TrT_chemistry_demograph.xlsx') %>%
  select(Subjname=Akronym, Gender=Sex, Age, Genotype,
         PET_same_day, `MBq PET1`, `MBq PET2`, bodyMass=Weight_kg) %>%
  gather(PETNo, injRad, contains('MBq')) %>%
  mutate(PETNo = as.numeric(str_match(PETNo, '\\\\d'))) %>%
  mutate(PET = paste(Subjname, PETNo, sep='_'))
```

## Summary Statistics

Below are presented some summary statistics of the demographic data.

```
demog %>%
  select(Age, InjectedRadioactivity = injRad) %>%
  describe() %>%
  pandoc.table(digits=3, caption = "Summary Statistics", split.tables=Inf)
```

Table 1: Summary Statistics

	vars	n	mean	sd	median	trimmedmad	min	max	range	skew	kurtosis	se
Age	1	24	23.9	2.99	24	23.8	3.71	20	29	9	0.226	-1.32
InjectedRadioactivity	2	24	395	51.7	403	401	51.9	248	462	214	-	0.722

```
counts <- demog %>%
  filter(PETNo==1)

pandoc.table(table(counts$Gender, counts$Genotype), caption = "Gender and Genotype")
```

Table 2: Gender and Genotype

	HAB	MAB
F	2	4
M	4	2

## TACs and Blood Data

First, we must read in the TAC and blood data. It should be noted that the blood data is already dispersion corrected, and the plasma data is already metabolite corrected, thus the plasma fraction is set to 100% in the input data frame. It should also be noted that we set all negative values of both blood and plasma radioactivity concentrations to zero.

```
tacdat <- read_csv('../RawData/tacdata.csv') %>%
  group_by(PET) %>%
  nest(.key = 'tacs')

blooddat <- read_csv('../RawData/blooddata.csv') %>%
  mutate(Cbl.disp.corr = ifelse(Cbl.disp.corr < 0, 0, Cbl.disp.corr),
         Cpl..metabcorr. = ifelse(Cpl..metabcorr. < 0, 0, Cpl..metabcorr.)) %>%
  group_by(PET) %>%
  nest(.key='blooddata') %>%
  mutate(input = map(blooddata, ~blood_interp(
    t_blood = .x$ABSS.sec/60, blood=.x$Cbl.disp.corr,
    t_plasma=.x$ABSS.sec/60, plasma=.x$Cpl..metabcorr.,
    t_parentfrac = 1, parentfrac=1 ) ))

dat <- inner_join(tacdat, blooddat) %>%
  separate(PET, c("Subjname", "PETNo"), sep='_', remove = F, convert=T)
```

Here, the TACs and blood data are merged with the demographic data.

```
datdf <- dat %>%
  inner_join(demog) %>%
  arrange(PET)
```

## Kinetic Modelling

Kinetic modelling is performed using the kinfitr R package.

### Fitting of the Delay and Blood Volume Fraction

Here, the delay and blood bolume fraction are fitted using the whole brain ROI using 2TCM.

```
datdf <- datdf %>%
  group_by(PET) %>%
  mutate(WB_delay = map2(tacs, input,
    ~twotcm(t_tac = .x$Times/60, tac = .x$WB, input = .y,
      frameStartEnd = c(1,33), inpshift.upper = 1))) %>%
  ungroup()
```

### Rearrangement of the Data into Long Format

Here, the data is transformed from wide format (where each row represents one PET measurement) into long format (where each row represents one region of each PET measurement)

```
tacs <- datdf %>%
  select(PET, tacs, Subjname, PETNo) %>%
  unnest() %>%
  gather(Region, TAC, -(PET:Weights), -(StartTime:Duration)) %>%
  group_by(PET, Subjname, PETNo, Region) %>%
  nest() %>%
  rename(tacs=data)

longdat <- datdf %>%
  select(PET, Subjname, PETNo, input, WB_delay, bodyMass, injRad) %>%
  inner_join(tacs)
```

### Define functions for fitting the models

Here we define functions for fitting each of the models.

```
# Total SUV
calcSUV <- function(tacs, bodymass, injRad) {
  SUV(t_tac = tacs$Times/60,
       tac = tacs$TAC*0.037,      # To kBq - because of kg
       injRad = injRad,
       bodymass = bodymass,
       frameStartEnd=c(1,33))
}
```

```

# 40-60 Minute SUV
calcSUV_4060 <- function(SUVout) {

  interptime = seq(SUVout$tacs$Time[1], rev(SUVout$tacs$Time)[1], by=1/60)
  interceptac = interp1(x = SUVout$tacs$Time, SUVout$tacs$SUV,
                        xi = interptime, method = 'linear')
  step=interptime[2] - interptime[1]

  SUVdf <- data.frame(interptime, interceptac) %>%
    filter(interptime > 40 & interptime <= 60)

  out <- mean(SUVdf$interceptac)
  return(out)
}

# MA1 using the fitted delay and vB from delayFit
fitma1 <- function(tacs, input, delayFit) {
  ma1(t_tac = tacs$Times/60, tac = tacs$TAC, input = input, tstarIncludedFrames = 6,
       inpshift = delayFit$par$inpshift, frameStartEnd=c(1,33), weights=tacs$Weights,
       vB=delayFit$par$vB)
}

# 2TCM using the fitted delay and vB from delayFit
fit2tcm <- function(tacs, input, delayFit) {
  twotcm(t_tac = tacs$Times/60, tac = tacs$TAC, input = input,
         inpshift = delayFit$par$inpshift, vB=delayFit$par$vB,
         frameStartEnd=c(1,33), weights=tacs$Weights)
}

```

## Fit all Kinetic Models

Here, all kinetic models are fitted, and the  $V_T$  values are extracted and added to the data frame.

```

longdat <- longdat %>%
  # SUV Total
  mutate(suvout_tot = pmap(list(tacs, bodyMass, injRad), calcSUV)) %>%
  mutate(SUV_tot = map_dbl(suvout_tot, c('par', 'intSUV'))) %>%

  # SUV 40-60 Minutes
  mutate(SUV_4060 = map_dbl(suvout_tot, calcSUV_4060)) %>%

  # MA1
  mutate(fit_ma1 = pmap(list(tacs, input, WB_delay), fitma1)) %>%
  mutate(Vt_ma1 = map_dbl(fit_ma1, c('par', 'Vt'))) %>%

  # 2TCM using fitted vB and delay
  mutate(fit_2tcm= pmap(list(tacs, input, WB_delay), fit2tcm)) %>%
  mutate(Vt_2tcm = map_dbl(fit_2tcm, c('par', 'Vt')))

```

## Calculate Ratio Outcomes

Here we calculate the Ratio Outcomes for SUV<sub>Total</sub>, SUV<sub>40-60</sub>, 2TCM and MA1.

```
trtdata <- longdat %>%
  select(Subjectname, PETNo, Region, SUV_tot, SUV_4060, Vt_ma1, Vt_2tcm) %>%
  gather(Measure, Value, -Subjectname, -PETNo, -Region) %>%
  spread(Region, Value) %>%
  mutate(FC_CBL = FC/CBL, FC_WB=FC/WB)
```

## Evaluate Spread and Test-Retest Metrics

Here we evaluate the spread and test-retest performance for all measures. Due to the influence of genotype, there are systematic differences between individuals. There are high-affinity binders (HABs) and medium-affinity binders (MABs) in this analysis. Thus, test-retest analyses are performed separately for HABs and MABs. Ratio-based outcome measures should account for genotype. Thus, we also performed test-retest analysis on all measurements together.

Note that SEM, the standard error of measurement, can be presented as an estimated standard error around each estimate (i.e. in the same units as the outcome), or as a percentage of the mean outcome like an estimate of the within-individual coefficient of variation. Here, due to differences in the scale of the outcome measures, it is presented as a percentage. This is why it is transformed from the original output.

### HABs and MABs together

```
trtout <- trtdata %>%
  gather(Region, Binding, -(Subjectname:Measure)) %>%
  filter(Subjectname != 'uqis') %>%
  spread(PETNo, Binding, drop = T) %>%
  group_by(Region, Measure) %>%
  do(trt = granviller::trt(.`1`, .`2`)$tidy) %>%
  ungroup() %>%
  unnest() %>%
  select(-se, -skew, -kurtosis, -md, -avgpercchange) %>%
  mutate('SEM %' = sem/mean*100,
        'COV %' = cov*100) %>%
  select(-sem, -cov) %>%
  arrange(Measure, Region) %>%
  add_column(Genotype='All', .after='Measure')

pandoc.table(trtout, digits=2, split.tables=Inf,
             caption='Test-Retest Analysis for HABs and MABs together')
```

Table 3: Test-Retest Analysis for HABs and MABs together

Region	Measure	Genotype	mean	sd	icc	aapd	SEM %	COV %
CBL	SUV_4060	All	0.98	0.3	0.9	10	9.4	30
FC	SUV_4060	All	0.92	0.27	0.88	13	10	30
FC_CBL	SUV_4060	All	0.95	0.059	0.63	4.4	3.8	6.2
FC_WB	SUV_4060	All	1	0.04	0.89	1.5	1.3	3.9
STR	SUV_4060	All	0.89	0.27	0.84	16	12	31

Region	Measure	Genotype	mean	sd	icc	aapd	SEM %	COV %
TC	SUV_4060	All	0.93	0.27	0.89	11	9.6	29
THA	SUV_4060	All	1.2	0.37	0.87	15	11	32
WB	SUV_4060	All	0.9	0.24	0.86	13	10	27
CBL	SUV_tot	All	80	17	0.83	11	8.8	21
FC	SUV_tot	All	78	17	0.8	12	9.5	21
FC_CBL	SUV_tot	All	0.98	0.045	0.77	2.5	2.2	4.6
FC_WB	SUV_tot	All	1.1	0.039	0.9	1.3	1.1	3.6
STR	SUV_tot	All	76	17	0.77	14	11	22
TC	SUV_tot	All	75	15	0.79	12	9.3	20
THA	SUV_tot	All	92	20	0.74	14	11	21
WB	SUV_tot	All	72	14	0.75	13	9.7	19
CBL	Vt_2tcm	All	3	1.7	0.93	21	15	55
FC	Vt_2tcm	All	2.9	1.5	0.93	19	14	53
FC_CBL	Vt_2tcm	All	0.98	0.07	0.54	4.7	4.9	7.1
FC_WB	Vt_2tcm	All	1	0.037	0.52	3	2.5	3.6
STR	Vt_2tcm	All	2.8	1.5	0.93	19	14	53
TC	Vt_2tcm	All	2.9	1.6	0.94	19	13	54
THA	Vt_2tcm	All	3.8	2.3	0.93	23	16	60
WB	Vt_2tcm	All	2.8	1.5	0.93	19	14	53
CBL	Vt_ma1	All	3.4	1.8	0.94	18	13	53
FC	Vt_ma1	All	3.1	1.6	0.93	18	14	51
FC_CBL	Vt_ma1	All	0.93	0.074	0.62	6.1	4.9	7.9
FC_WB	Vt_ma1	All	1	0.035	0.86	1.6	1.3	3.4
STR	Vt_ma1	All	3	1.6	0.93	18	14	52
TC	Vt_ma1	All	3.2	1.6	0.94	17	12	52
THA	Vt_ma1	All	4.1	2.4	0.94	21	15	58
WB	Vt_ma1	All	3.1	1.6	0.93	18	14	51

## Separated by Genotype

```

trtdata$Genotype <- ifelse(trtdata$Subjname %in% demog$Subjname[demog$Genotype == 'HAB'],
                           yes = 'HAB', no = 'MAB')

trtout_genotypes <- trtdata %>%
  gather(Region, Binding, -(Subjname:Measure), -Genotype) %>%
  filter(Subjname != 'uqis') %>%
  spread(PETNo, Binding) %>%
  group_by(Region, Measure, Genotype) %>%
  do(trt = granviller::trt(. $`1`, . $`2`)$tidy) %>%
  ungroup() %>%
  unnest() %>%
  select(-se, -skew, -kurtosis, -md, -avgpercchange) %>%
  mutate('SEM %' = sem / mean * 100,
        'COV %' = cov * 100) %>%
  select(-sem, -cov) %>%
  arrange(Measure, Region)

```

## HABs

```
hab_trtout <- filter(trtout_genotypes, Genotype=='HAB')
pandoc.table(hab_trtout, digits=2, split.tables=Inf,
              caption='Test-Retest Analysis for HABs')
```

Table 4: Test-Retest Analysis for HABs

Region	Measure	Genotype	mean	sd	icc	aapd	SEM %	COV %
CBL	SUV_4060	HAB	1.1	0.27	0.8	13	11	24
FC	SUV_4060	HAB	1.1	0.24	0.76	13	11	22
FC_CBL	SUV_4060	HAB	0.94	0.071	0.85	3.2	2.9	7.5
FC_WB	SUV_4060	HAB	1	0.019	0.6	1.3	1.2	1.9
STR	SUV_4060	HAB	1	0.24	0.74	15	12	23
TC	SUV_4060	HAB	1.1	0.23	0.77	13	10	22
THA	SUV_4060	HAB	1.4	0.32	0.78	14	11	24
WB	SUV_4060	HAB	1	0.22	0.78	12	10	21
CBL	SUV_tot	HAB	89	15	0.69	11	9.1	16
FC	SUV_tot	HAB	87	12	0.65	9.9	8.4	14
FC_CBL	SUV_tot	HAB	0.99	0.056	0.96	1.3	1.1	5.7
FC_WB	SUV_tot	HAB	1.1	0.02	0.7	1.2	0.97	1.8
STR	SUV_tot	HAB	85	13	0.62	11	9.1	15
TC	SUV_tot	HAB	83	13	0.63	11	9.3	15
THA	SUV_tot	HAB	100	16	0.71	9.6	8.6	16
WB	SUV_tot	HAB	79	12	0.68	10	8.4	15
CBL	Vt_2tcm	HAB	4	1.8	0.91	19	14	44
FC	Vt_2tcm	HAB	3.9	1.6	0.89	21	14	42
FC_CBL	Vt_2tcm	HAB	0.97	0.079	0.87	3.4	2.9	8.1
FC_WB	Vt_2tcm	HAB	1.1	0.031	0.33	3.1	2.4	3
STR	Vt_2tcm	HAB	3.6	1.6	0.92	19	13	45
TC	Vt_2tcm	HAB	3.9	1.7	0.92	18	12	44
THA	Vt_2tcm	HAB	5.1	2.5	0.91	24	15	49
WB	Vt_2tcm	HAB	3.7	1.6	0.91	20	14	44
CBL	Vt_ma1	HAB	4.5	1.9	0.92	17	12	43
FC	Vt_ma1	HAB	4.1	1.7	0.9	20	13	42
FC_CBL	Vt_ma1	HAB	0.92	0.078	0.79	4.8	3.9	8.5
FC_WB	Vt_ma1	HAB	1	0.026	0.79	1.6	1.2	2.6
STR	Vt_ma1	HAB	3.9	1.7	0.91	22	13	43
TC	Vt_ma1	HAB	4.1	1.8	0.93	17	11	43
THA	Vt_ma1	HAB	5.4	2.6	0.91	24	14	48
WB	Vt_ma1	HAB	4	1.7	0.91	20	13	43

## MABs

```
mab_trtout <- filter(trtout_genotypes, Genotype=='MAB')
pandoc.table(mab_trtout, digits=2, split.tables=Inf,
              caption='Test-Retest Analysis for MABs')
```

Table 5: Test-Retest Analysis for MABs

Region	Measure	Genotype	mean	sd	icc	aapd	SEM %	COV %
CBL	SUV_4060	MAB	0.84	0.25	0.95	7.3	6.3	30
FC	SUV_4060	MAB	0.8	0.24	0.91	13	9.1	31
FC_CBL	SUV_4060	MAB	0.95	0.05	0.32	5.4	4.3	5.2
FC_WB	SUV_4060	MAB	1	0.041	0.89	1.6	1.3	4
STR	SUV_4060	MAB	0.77	0.24	0.87	16	11	31
TC	SUV_4060	MAB	0.8	0.24	0.93	9.9	7.7	30
THA	SUV_4060	MAB	1	0.36	0.9	16	11	35
WB	SUV_4060	MAB	0.79	0.22	0.87	13	9.7	27
CBL	SUV_tot	MAB	72	16	0.86	11	8	21
FC	SUV_tot	MAB	71	16	0.8	14	10	23
FC_CBL	SUV_tot	MAB	0.98	0.037	0.45	3.5	2.8	3.8
FC_WB	SUV_tot	MAB	1.1	0.038	0.89	1.4	1.2	3.6
STR	SUV_tot	MAB	69	17	0.77	17	12	25
TC	SUV_tot	MAB	69	15	0.83	12	8.9	22
THA	SUV_tot	MAB	86	21	0.73	17	13	24
WB	SUV_tot	MAB	67	14	0.74	15	11	21
CBL	Vt_2tcm	MAB	2.2	1	0.9	22	15	48
FC	Vt_2tcm	MAB	2.2	0.99	0.93	17	12	46
FC_CBL	Vt_2tcm	MAB	0.99	0.065	0.17	5.9	6	6.6
FC_WB	Vt_2tcm	MAB	1	0.038	0.55	2.9	2.5	3.7
STR	Vt_2tcm	MAB	2	0.8	0.86	18	15	40
TC	Vt_2tcm	MAB	2.1	1	0.92	21	13	47
THA	Vt_2tcm	MAB	2.8	1.5	0.93	22	15	54
WB	Vt_2tcm	MAB	2.1	0.94	0.92	18	13	45
CBL	Vt_ma1	MAB	2.5	1.1	0.92	19	12	43
FC	Vt_ma1	MAB	2.3	0.95	0.92	16	12	41
FC_CBL	Vt_ma1	MAB	0.94	0.071	0.48	7.1	5.4	7.5
FC_WB	Vt_ma1	MAB	1	0.04	0.89	1.6	1.3	3.9
STR	Vt_ma1	MAB	2.3	0.95	0.9	16	13	42
TC	Vt_ma1	MAB	2.3	0.99	0.93	16	12	42
THA	Vt_ma1	MAB	3	1.5	0.93	19	13	51
WB	Vt_ma1	MAB	2.3	0.91	0.91	16	12	40

## Article Summary

```

summary_trtout <- bind_rows(list(trtout, hab_trtout, mab_trtout)) %>%
  arrange(rev(Measure), Region, Genotype) %>%
  filter(Region %in% c("FC", "FC_WB", "FC_CBL")) %>%
  filter(Measure %in% c("Vt_2tcm", "SUV_4060")) %>%
  #separate(Region, c("Region", "Denominator"), sep = '_')
  mutate(Denominator = str_match(Region, '_(.*)')[,2]) %>%
  select(Measure, Genotype, Denominator=Denominator, Mean=mean, `COV %` ,
    ICC=icc, VAR=aapd, `SEM %`) %>%
  mutate(Measure = ifelse(Measure=='Vt_2tcm', 'V~T-2TCM~', 'SUV~40-60min~')) %>%
  filter(! (Denominator=='-' & Genotype=='All')) %>%
  replace_na(list(Denominator='-'))
}

pandoc.table(summary_trtout, digits=2, split.tables=Inf,

```

```
caption='Test-Retest Table for Article')
```

Table 6: Test-Retest Table for Article

Measure	Genotype	Denominator	Mean	COV %	ICC	VAR	SEM %
V <sub>T-2TCM</sub>	HAB	-	3.9	42	0.89	21	14
V <sub>T-2TCM</sub>	MAB	-	2.2	46	0.93	17	12
V <sub>T-2TCM</sub>	All	CBL	0.98	7.1	0.54	4.7	4.9
V <sub>T-2TCM</sub>	HAB	CBL	0.97	8.1	0.87	3.4	2.9
V <sub>T-2TCM</sub>	MAB	CBL	0.99	6.6	0.17	5.9	6
V <sub>T-2TCM</sub>	All	WB	1	3.6	0.52	3	2.5
V <sub>T-2TCM</sub>	HAB	WB	1.1	3	0.33	3.1	2.4
V <sub>T-2TCM</sub>	MAB	WB	1	3.7	0.55	2.9	2.5
SUV <sub>40-60min</sub>	HAB	-	1.1	22	0.76	13	11
SUV <sub>40-60min</sub>	MAB	-	0.8	31	0.91	13	9.1
SUV <sub>40-60min</sub>	All	CBL	0.95	6.2	0.63	4.4	3.8
SUV <sub>40-60min</sub>	HAB	CBL	0.94	7.5	0.85	3.2	2.9
SUV <sub>40-60min</sub>	MAB	CBL	0.95	5.2	0.32	5.4	4.3
SUV <sub>40-60min</sub>	All	WB	1	3.9	0.89	1.5	1.3
SUV <sub>40-60min</sub>	HAB	WB	1	1.9	0.6	1.3	1.2
SUV <sub>40-60min</sub>	MAB	WB	1	4	0.89	1.6	1.3

## Correlations with V<sub>T</sub>

Here, the data is arranged for assessment of the correlation with V<sub>T</sub>

```
plotdat <- trtdata %>%
  select(-CBL, -(STR:WB)) %>%
  filter(Measure != 'SUV_tot') %>%
  gather(Region, Value, -(Subjname:Measure), -Genotype) %>%
  #mutate(Region = str_replace(Region, 'FC', replacement = '')) %>%
  unite(Outcome, Measure, Region) %>%
  spread(Outcome, Value) %>%
  mutate(PET = paste(Subjname, PETNo, sep='_')) %>%
  filter(PET != 'uqis_2')
```

Here, correlations with 2TCM V<sub>T</sub> are calculated.

```
corout <- plotdat %>%
  gather(Measure, Binding, -(Subjname:Genotype), -Vt_2tcm_FC) %>%
  group_by(Measure, Genotype) %>%
  summarise('R^2'=cor(Binding, Vt_2tcm_FC)^2) %>%
  arrange(Measure, Genotype) %>%
  ungroup() %>%
  mutate(Measure = str_replace(string=Measure, pattern='_FC', replacement='')) %>%
  mutate(Measure = str_replace(string=Measure, pattern='_4060', replacement='')) %>%
  mutate(Measure = str_replace(string=Measure, pattern='~', replacement='~')) %>%
  mutate(Measure = str_replace(string=Measure, pattern='CBL', replacement='CBL~')) %>%
  mutate(Measure = str_replace(string=Measure, pattern='WB', replacement='WB~')) %>%
  mutate(Measure = str_replace(string=Measure, pattern='Vt~2tcm_', replacement='DVR~')) %>%
  filter(!str_detect(Measure, 'ma1')) %>%
  mutate(Measure = str_replace(string=Measure, pattern='Vt~ma1', replacement='V~T-MA1~'))
```

```
pandoc.table(corout, digits=2, split.tables=Inf, caption="Correlations with V~T~")
```

Table 7: Correlations with  $V_T$

Measure	Genotype	$R^2$
SUV	HAB	0.64
SUV	MAB	0.86
$SUV_{CBL}$	HAB	0.0046
$SUV_{CBL}$	MAB	0.0034
$SUV_{WB}$	HAB	0.0061
$SUV_{WB}$	MAB	0.33
$DVR_{CBL}$	HAB	0.022
$DVR_{CBL}$	MAB	0.034
$DVR_{WB}$	HAB	0.3
$DVR_{WB}$	MAB	0.0056
$V_{T-MA1}$	HAB	0.99
$V_{T-MA1}$	MAB	0.99

## Plot of Ratio Associations with $V_T$

```
longplotdat <- plotdat %>%
  gather(Measure, Binding, -PET, -(Subjname:Genotype), -Vt_2tcm_FC) %>%
  group_by(Measure) %>%
  nest() %>%
  mutate(outplot = map(data,
    ~ggplot(.x, aes(x=Vt_2tcm_FC, y=Binding, colour=Genotype)) +
      geom_point() + theme_blog() +
      geom_line(aes(group=Subjname), linetype="dotted") +
      theme(axis.title.y=element_blank(),
            plot.background = element_rect(fill = "white")) +
      scale_color_brewer(palette = 'Set1') +
      geom_smooth(method="lm", se=F)))

# SUVWB
a <- longplotdat$outplot[[which(longplotdat$Measure=='SUV_4060_FC_WB')]] +
  expand_limits(x=0,y=c(0.95,1.1)) +
  labs(x=expression(V[T]),
       y=expression(SUVR[WB])) +
  theme(axis.title.x=element_blank())

# SUVCBL
b <- longplotdat$outplot[[which(longplotdat$Measure=='SUV_4060_FC_CBL')]] +
  expand_limits(x=0,y=c(0.8,1.1)) +
  labs(x=expression(V[T]),
       y=expression(SUVR[CBL])) +
  theme(axis.title.x=element_blank())

# DVRWB
c <- longplotdat$outplot[[which(longplotdat$Measure=='Vt_2tcm_FC_WB')]] +
  expand_limits(x=0,y=c(0.95,1.1)) +
  labs(x=expression(paste('Frontal Cortex ',V[T])),
```

```

y=expression(DVR[WB]))
```

```

# DVRCBL
d <- longplotdat$outplot[[which(longplotdat$Measure=='Vt_2tcm_FC_CBL')]] +
  expand_limits(x=0,y=c(0.8,1.1)) +
  labs(x=expression(paste('Frontal Cortex ',V[T])),
```

```

  y=expression(DVR[CBL]))
```

```

# Text Labels
e = textGrob(label = 'SUVR')
f = textGrob(label = 'DVR')
g = textGrob(label = 'Whole Brain')
h = textGrob(label = 'Cerebellum')
```

```

# Legend
g_legend<-function(a.gplot){
  tmp <- ggplot_gtable(ggplot_build(a.gplot))
  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
  legend <- tmp$grobs[[leg]]
  legend
}
```

```

a2 <- ggplot(plotdat, aes(x=Vt_2tcm_FC, y=SUV_4060_FC_WB, colour=Genotype)) +
  geom_point() + theme_bw() +
  scale_color_brewer('Genotype', palette = 'Set1')
```

```

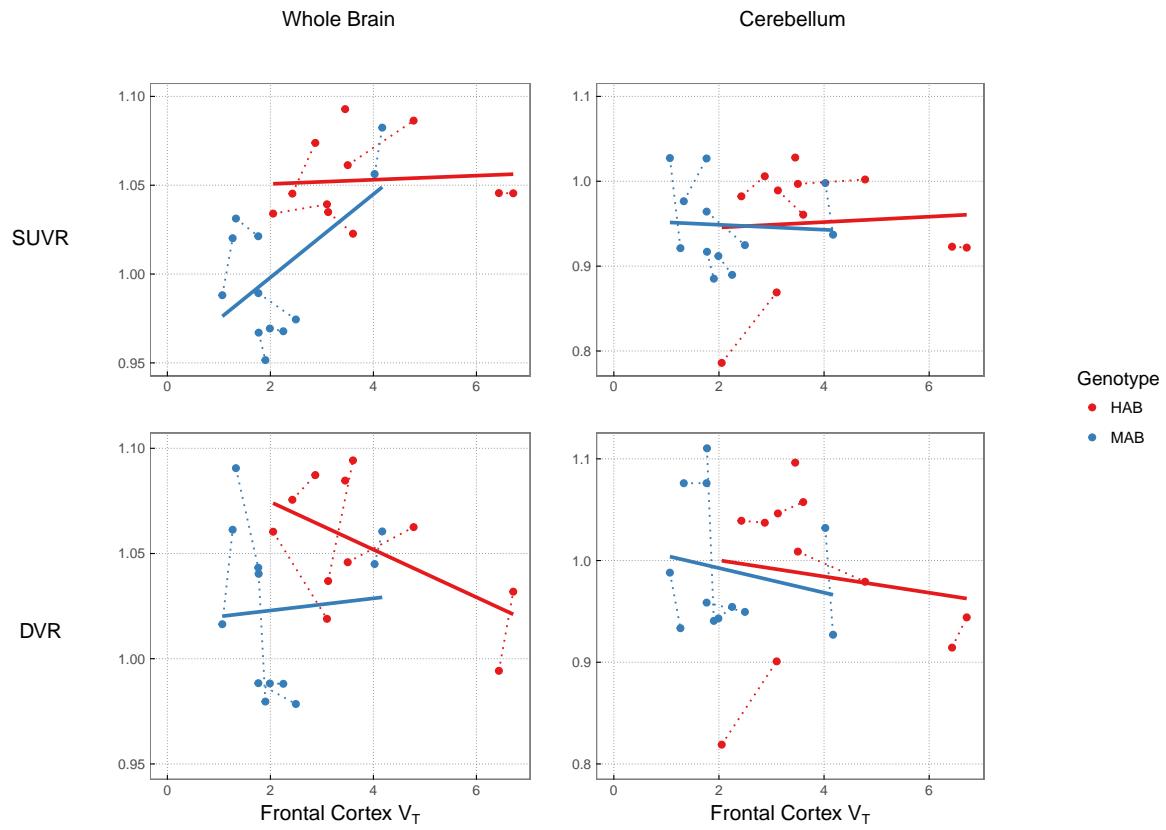
genlegend <- g_legend(a2)
```

```

layout <- rbind(c(NA, 7, 8, 9),
                c(1, 2, 3, 9),
                c(4, 5, 6, 9))
```

```

grid.arrange(grobs=list(e,a,b,f,c,d,g,h, genlegend),
             layout_matrix=layout, widths=c(1,4,4,2), heights=c(1,4,5))
```



## Plot Of Alternative Quantification Methods Associations with $V_T$

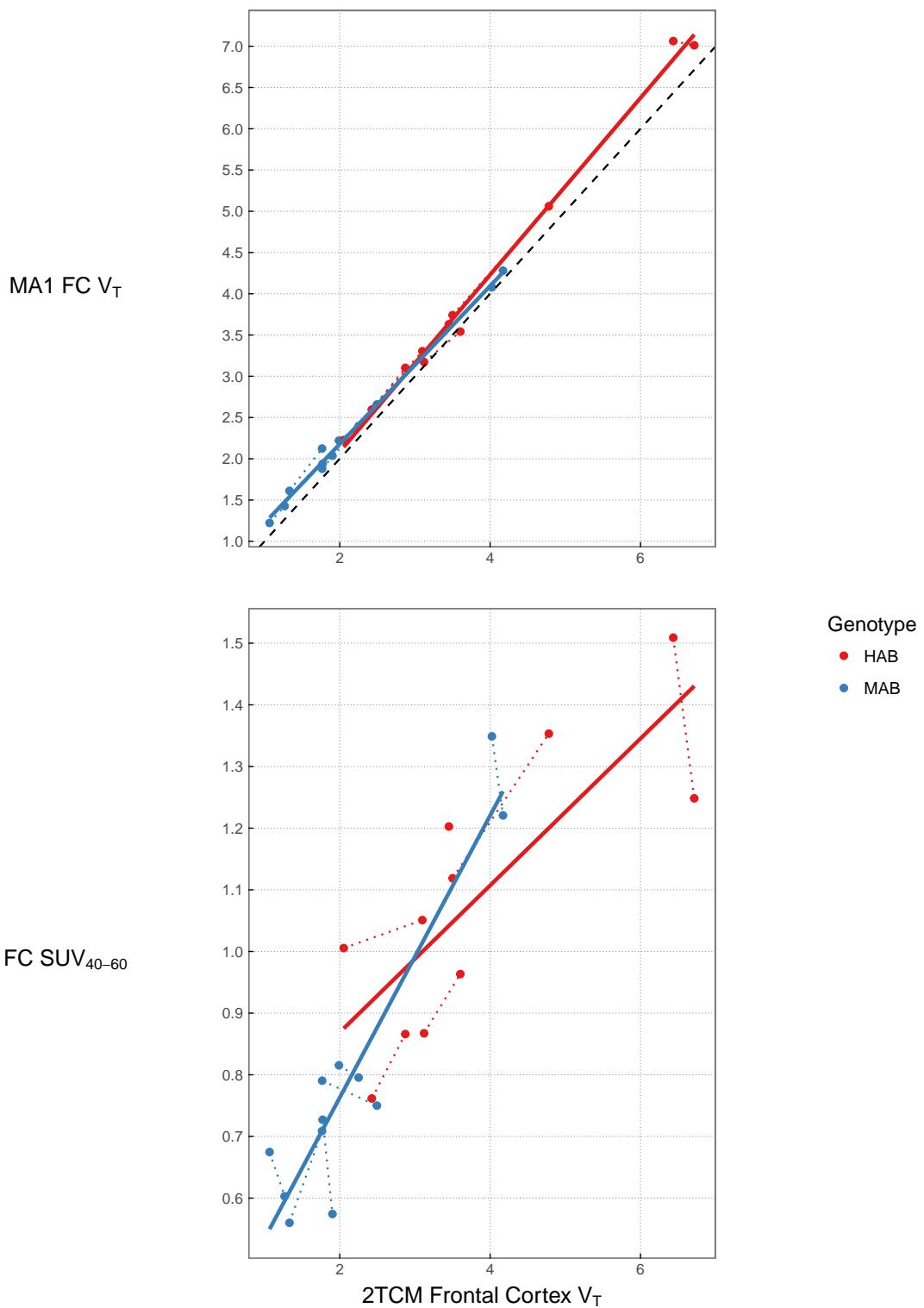
```
# MA1
Vta <- longplotdat$outplot[[which(longplotdat$Measure=='Vt_ma1_FC')]] +
  labs(x=expression(paste('2TCM Frontal Cortex ',V[T])),
       y=expression(paste('MA1 Frontal Cortex ',V[T]))) +
  theme(axis.title.x=element_blank()) +
  scale_y_continuous(breaks=seq(1,8,by=0.5)) +
  geom_abline(slope = 1, linetype='dashed')

Vtb <- longplotdat$outplot[[which(longplotdat$Measure=='SUV_4060_FC')]] +
  labs(x=expression(paste('2TCM Frontal Cortex ',V[T])),
       y=expression(paste('Frontal Cortex ',SUVV[40-60]))) +
  scale_y_continuous(breaks=seq(0.5,1.6,by=0.1))

# Text
alabel <- textGrob(label = expression(paste('MA1 FC ',V[T])))
blabel <- textGrob(label = expression(paste('FC ',SUV[40-60])))

layout2 <- rbind(c(3, 1, 5),
                  c(4, 2, 5))

grid.arrange(grobs=list(Vta, Vtb, alabel, blabel, genlegend),
             layout_matrix=layout2, widths=c(2,4,2), heights=c(4,5))
```



```
longdat <- longdat %>%
  inner_join(select(demog, PET, Genotype))
```

```
## Joining, by = "PET"
```

```

# SUV
SUV_genotype <- ggplot(plotdat, aes(x=Vt_2tcm_FC, y=SUV_4060_FC, colour=Genotype)) +
  geom_point() + theme_blog() +
  geom_line(aes(group=Subjname), linetype="dotted") +
  labs(x=expression(paste('2TCM ', V[T])),
       y=expression(paste(' ', SUV[40-60]))),
  title='Frontal Cortex',
  subtitle='Coloured by Genotype') +
  theme(plot.background = element_rect(fill = "white")) +
  scale_y_continuous(breaks=seq(0.5,1.6,by=0.1)) +
  scale_color_brewer(palette = 'Set1') + geom_smooth(method="lm", se=F) +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))

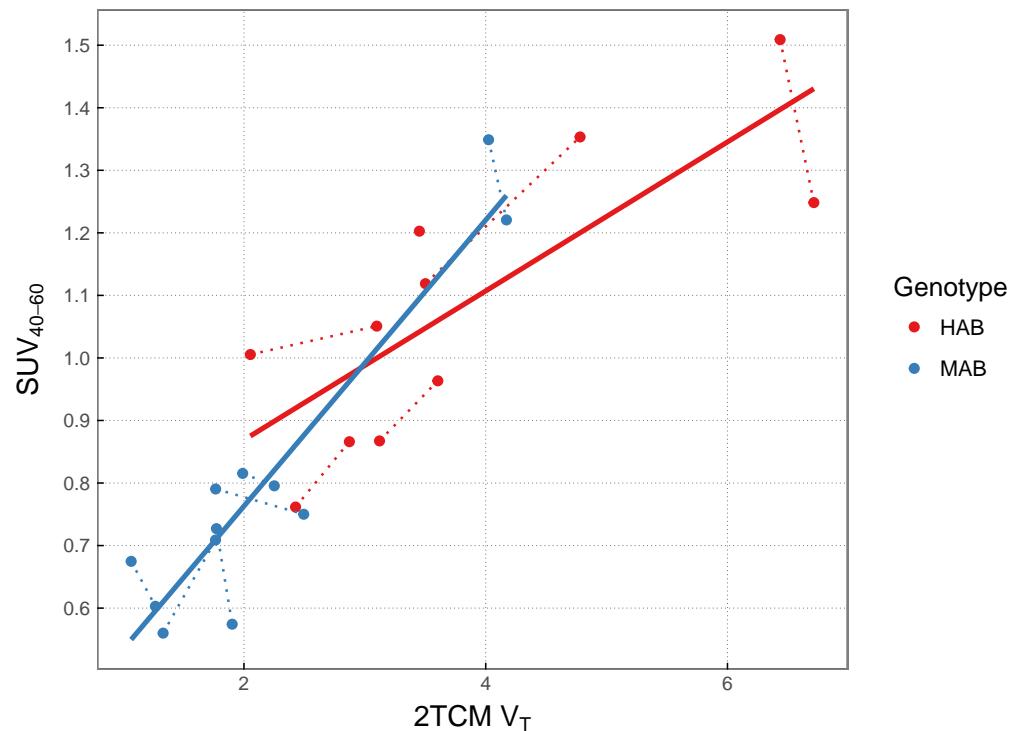
SUV_subject <- longdat %>%
  filter(PET != 'uqis_2') %>%
  ggplot(aes(x=Vt_2tcm, y=SUV_4060, colour=Subjname, shape=Genotype, linetype=Genotype)) +
  geom_point() + theme_blog() +
  labs(x=expression(paste('2TCM ', V[T])),
       y=expression(paste(' ', SUV[40-60]))),
  title='All Regions',
  subtitle='Coloured by Participant') +
  theme(plot.background = element_rect(fill = "white")) +
  geom_smooth(method="lm", se=F) +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))

subjleg <- g_legend( SUV_subject +
  theme_bw() +
  scale_colour_discrete(guide = FALSE) +
  geom_smooth(colour="black", fill=NA) )

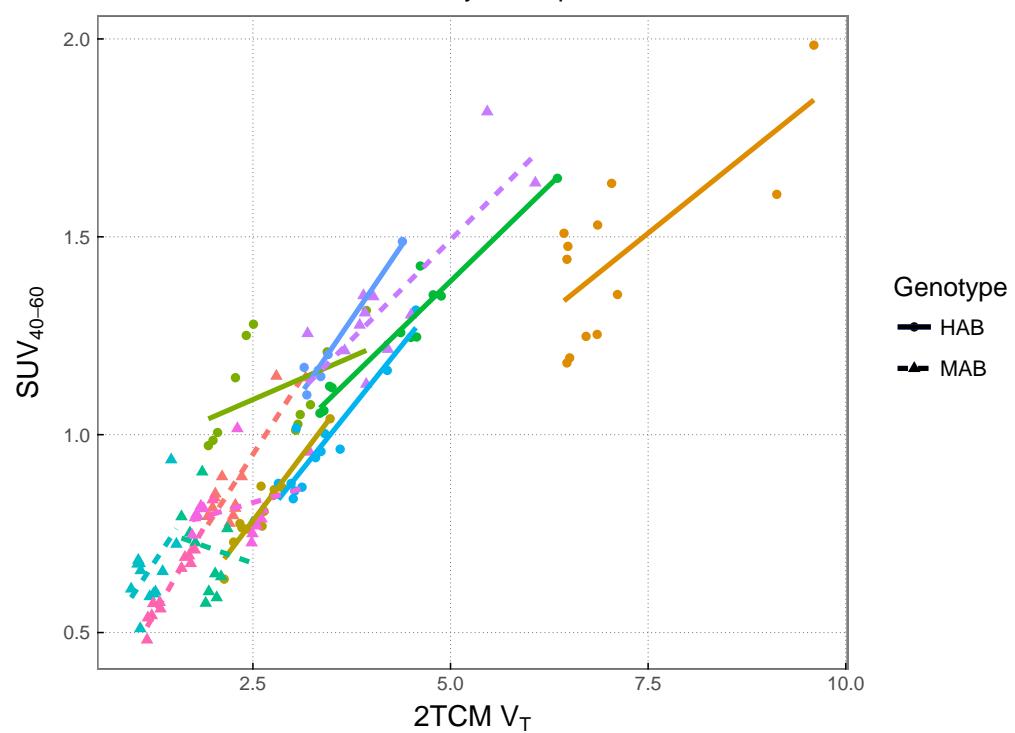
## `geom_smooth()` using method = 'loess'
grid.arrange(grobs=list(SUV_genotype, genlegend, SUV_subject, subjleg), ncol=2, heights=c(12,12), width=

```

**Frontal Cortex**  
Coloured by Genotype



**All Regions**  
Coloured by Participant



## Interregional Correlation

Here the interregional correlations for V<sub>T</sub> and for SUV are assessed

```
VT_2tcm <- longdat %>%
  select(PET, Genotype, Region, Vt_2tcm) %>%
  spread(Region, Vt_2tcm)

SUV_4060 <- longdat %>%
  select(PET, Genotype, Region, SUV_4060) %>%
  spread(Region, SUV_4060)

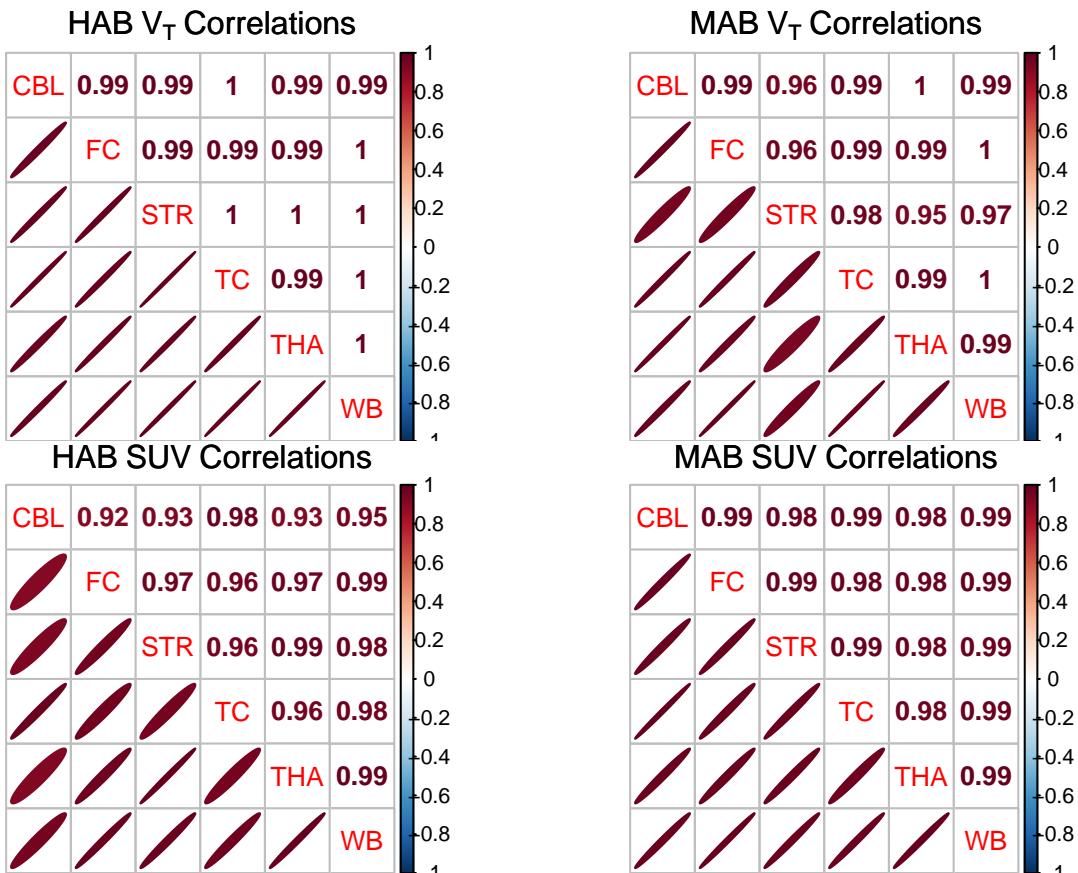
col2 <- colorRampPalette(rev(c("#67001F", "#B2182B", "#D6604D", "#F4A582", "#FDDBC7",
  "#FFFFFF", "#D1E5F0", "#92C5DE", "#4393C3", "#2166AC", "#053061")))

par(mfrow=c(2,2))
VT_2tcm %>%
  filter(Genotype=='HAB') %>%
  select(CBL:WB) %>%
  cor() %>%
  corrplot.mixed(lower='ellipse', upper='number',
    col=col2(200), diag='n',
    number.digits = 2, title=expression(HAB ~ V[T] ~ Correlations),
    mar=c(0,0,1,0))

VT_2tcm %>%
  filter(Genotype=='MAB') %>%
  select(CBL:WB) %>%
  cor() %>%
  corrplot.mixed(lower='ellipse', upper='number',
    col=col2(200), diag='n',
    number.digits = 2, , title=expression(MAB ~ V[T] ~ Correlations),
    mar=c(0,0,1,0))

SUV_4060 %>%
  filter(Genotype=='HAB') %>%
  select(CBL:WB) %>%
  cor() %>%
  corrplot.mixed(lower='ellipse', upper='number',
    col=col2(200), diag='n',
    number.digits = 2, title=expression(HAB ~ SUV ~ Correlations),
    mar=c(0,0,1,0))

SUV_4060 %>%
  filter(Genotype=='MAB') %>%
  select(CBL:WB) %>%
  cor() %>%
  corrplot.mixed(lower='ellipse', upper='number',
    col=col2(200), diag='n',
    number.digits = 2, title=expression(MAB ~ SUV ~ Correlations),
    mar=c(0,0,1,0))
```



#### Erratum Note:

Please note that there are some small discrepancies between figure 1 in the paper and the figure above, because an older version of figure 1 was accidentally submitted for publication. The submitted figure is an older version of the same figure produced using a previous version of *kinfitr* with a slightly different interpolation routine. This occurred at the only non-reproducible step of this analysis, namely an over-complicated email chain. We are in the process of transitioning to a collaborative, end-to-end reproducible analysis workflow, which will minimise the possibility of these errors in future.

## Principal Components Analysis

Principal components analysis (PCA) is performed to assess the cumulative explained variance. Note that scaling (z transformation) is performed both within genotype and region. Thus HABs and MABs can be combined in the same PCA analysis.

```
pcadat <- longdat %>%
  inner_join(select(datdf, PET, Genotype)) %>%
  select(PET, Subjname, PETNo, Genotype, Region, Vt_2tcm) %>%
  group_by(Region, Genotype) %>%
  mutate(Vt_2tcm.z = scale(Vt_2tcm)) %>%
  ungroup() %>%
  select(PET, Subjname, PETNo, Genotype, Region, Vt_2tcm.z) %>%
  filter(PET != 'uqis_2')
```

## PCA: All Regions

```
pca1 <- pcadat %>%
  filter(PETNo==1) %>%
  spread(Region, Vt_2tcm.z) %>%
  select(CBL:WB) %>%
  prcomp() %>%
  summary()

pandoc.table(pca1$importance, digits=3, caption = "PCA for PET1: All Regions")
```

Table 8: PCA for PET1: All Regions

	PC1	PC2	PC3	PC4	PC5	PC6
<b>Standard deviation</b>	2.37	0.226	0.122	0.0832	0.0294	0.0168
<b>Proportion of Variance</b>	0.987	0.00899	0.0026	0.00122	0.00015	5e-05
<b>Cumulative Proportion</b>	0.987	0.996	0.999	1	1	1

```
pca2 <- pcadat %>%
  filter(PETNo==2) %>%
  spread(Region, Vt_2tcm.z) %>%
  select(CBL:WB) %>%
  prcomp() %>%
  summary()

pandoc.table(pca2$importance, digits=3, caption = "PCA for PET2: All Regions", split.tables=Inf)
```

Table 9: PCA for PET2: All Regions

	PC1	PC2	PC3	PC4	PC5	PC6
<b>Standard deviation</b>	2.49	0.143	0.0851	0.0762	0.0408	0.0281
<b>Proportion of Variance</b>	0.994	0.00329	0.00116	0.00093	0.00027	0.00013
<b>Cumulative Proportion</b>	0.994	0.998	0.999	1	1	1

## PCA: Only FC, WB and CBL

This analysis is included to assess the PCA for only the target and the two reference regions in case the former analysis explained much of the total variability due to there being many target ROIs.

```
pca1 <- pcadat %>%
  filter(PETNo==1) %>%
  spread(Region, Vt_2tcm.z) %>%
  select(FC, CBL, WB) %>%
  prcomp() %>%
  summary()

pandoc.table(pca1$importance, digits=3, caption = "PCA for PET1: FC, WB and CBL")
```

Table 10: PCA for PET1: FC, WB and CBL

	PC1	PC2	PC3
<b>Standard deviation</b>	1.68	0.119	0.0248
<b>Proportion of Variance</b>	0.995	0.00499	0.00022
<b>Cumulative Proportion</b>	0.995	1	1

```
pca2 <- pcadat %>%
  filter(PETNo==2) %>%
  spread(Region, Vt_2tcm.z) %>%
  select(FC, CBL, WB) %>%
  prcomp() %>%
  summary()

pandoc.table(pca2$importance, digits=3, caption = "PCA for PET2: FC, WB and CBL")
```

Table 11: PCA for PET2: FC, WB and CBL

	PC1	PC2	PC3
<b>Standard deviation</b>	1.76	0.0805	0.0533
<b>Proportion of Variance</b>	0.997	0.00208	0.00091
<b>Cumulative Proportion</b>	0.997	0.999	1

## Plotting of 2TCM Model Fits

### Included Participants

```
longdat_showFits <- longdat %>%
  filter(Region %in% c('FC', 'WB', 'CBL'))

delayFits = map(longdat_showFits$WB_delay[longdat_showFits$Region=='WB'],
  ~plot_inptac_fit(.x) + ggtitle('Delay'))

allFits_2tcm = map2(longdat_showFits$fit_2tcm, longdat_showFits$Region,
  ~plot_kinfit(.x, roiname=.y))

PETs <- unique(longdat_showFits$PET)

delayFits <- data.frame(PET = PETs) %>%
  mutate(Fit = delayFits,
    Plot = 'Delay')

allFits <- data.frame(PET = rep(PETs, each=3)) %>%
  mutate(Fit = allFits_2tcm,
    Plot = 'Fit',
    PET = as.character(PET)) %>%
  bind_rows(delayFits) %>%
  arrange(PET, Plot)
```

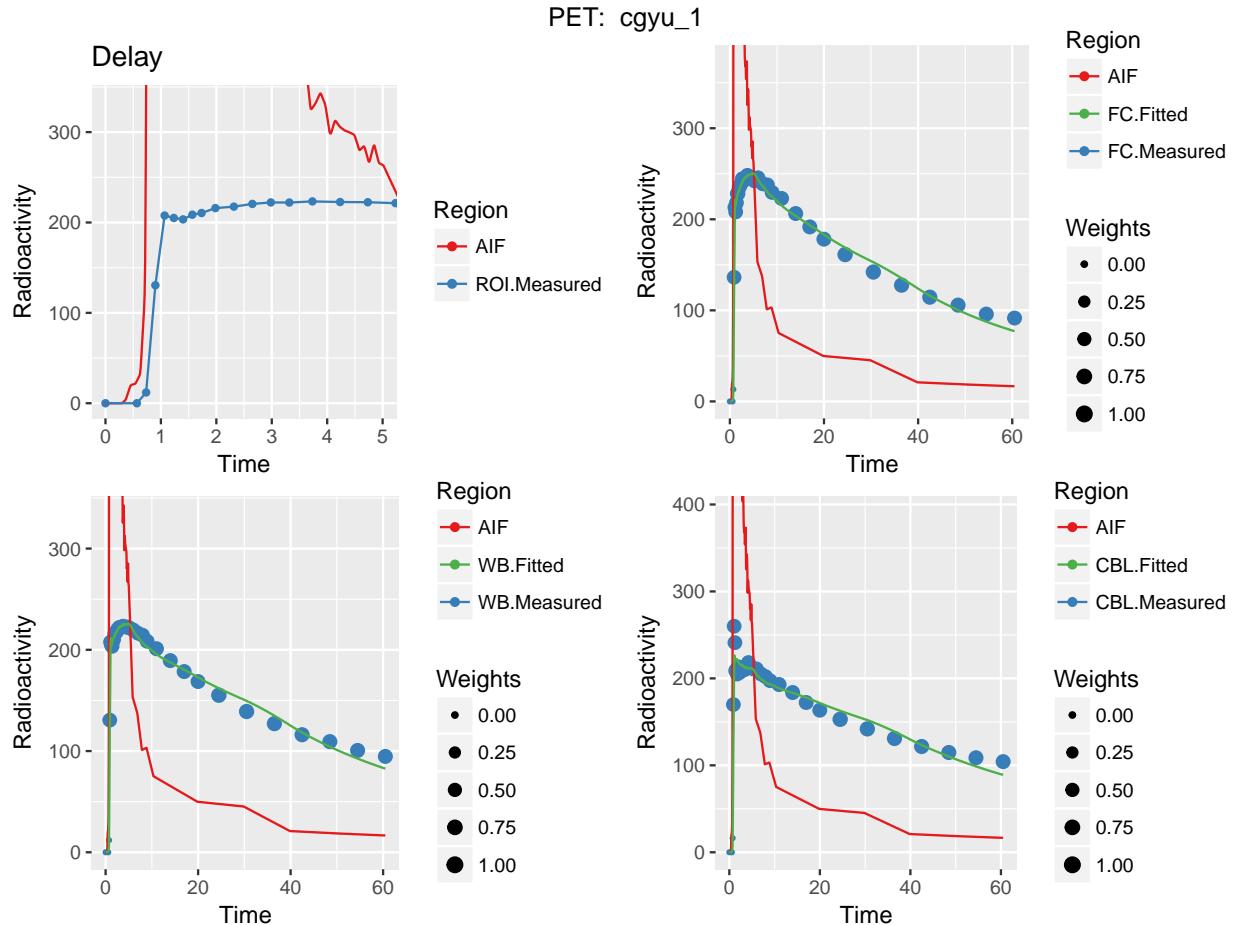
```

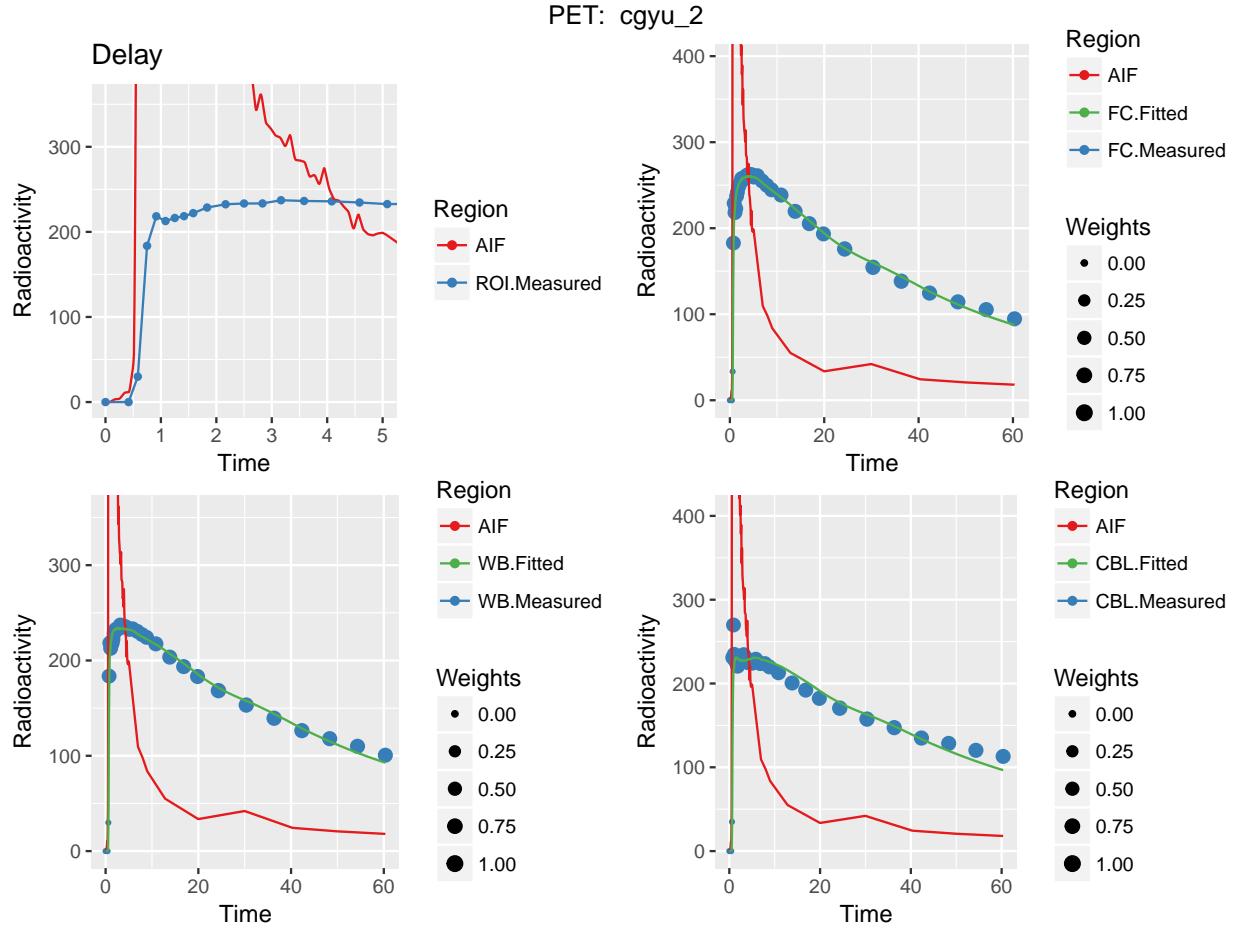
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
allFits_excluded <- allFits %>%
  filter(grepl(PET, pattern='uqis_2'))

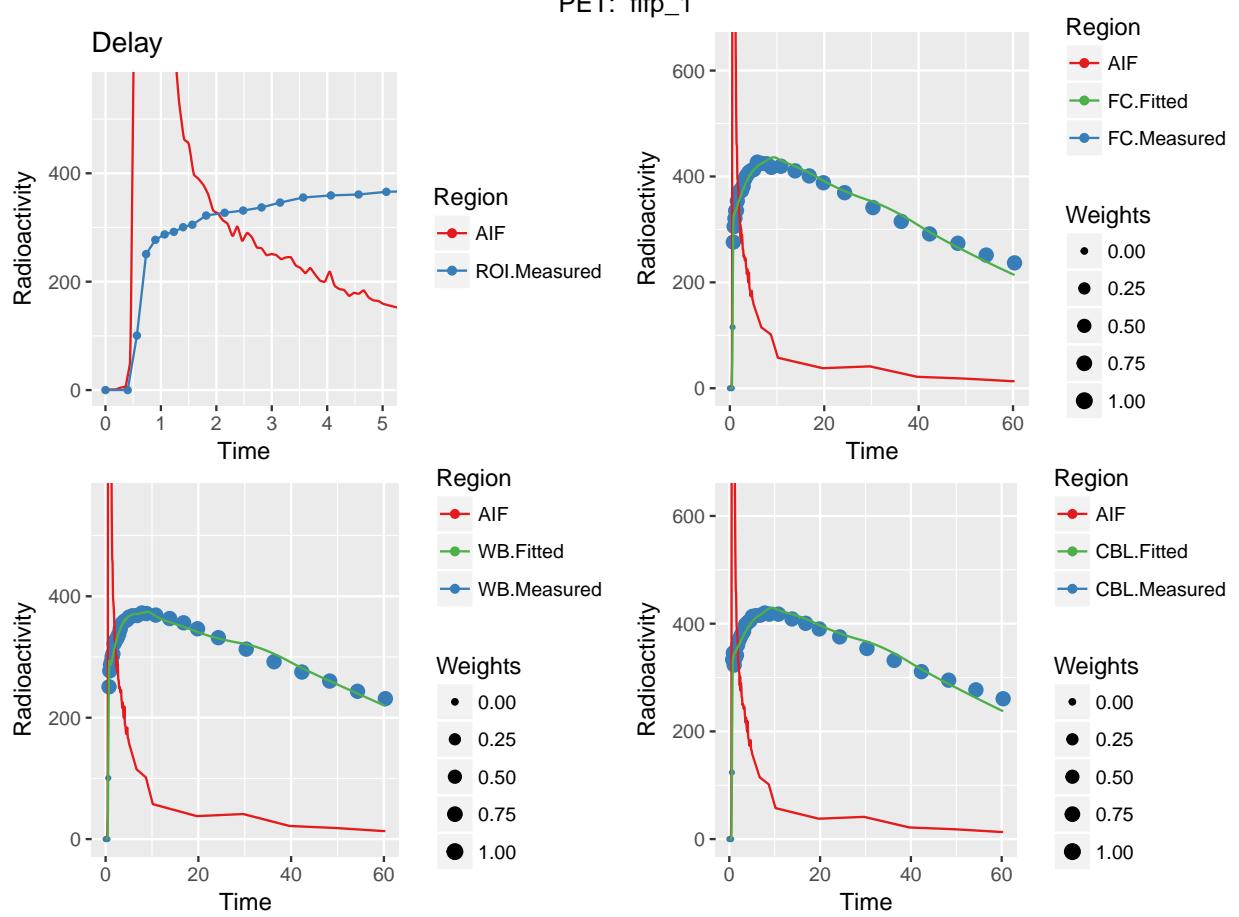
allFits <- allFits %>%
  filter(!grepl(PET, pattern='uqis_2'))

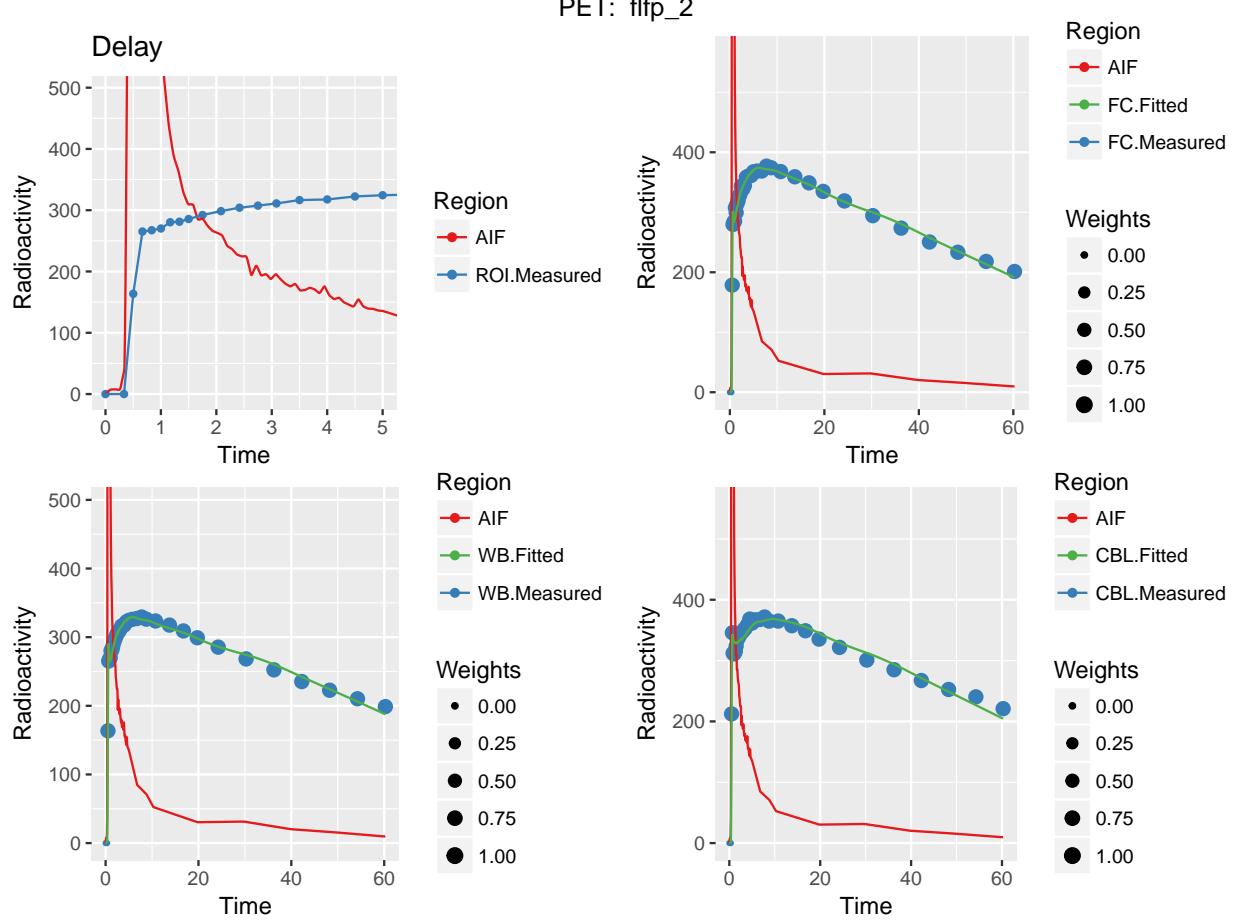
fitLabels <- unique(allFits$PET)
marrangeGrob(allFits$Fit, nrow=2, ncol=2, top=quote(paste('PET: ', PETs[g])))

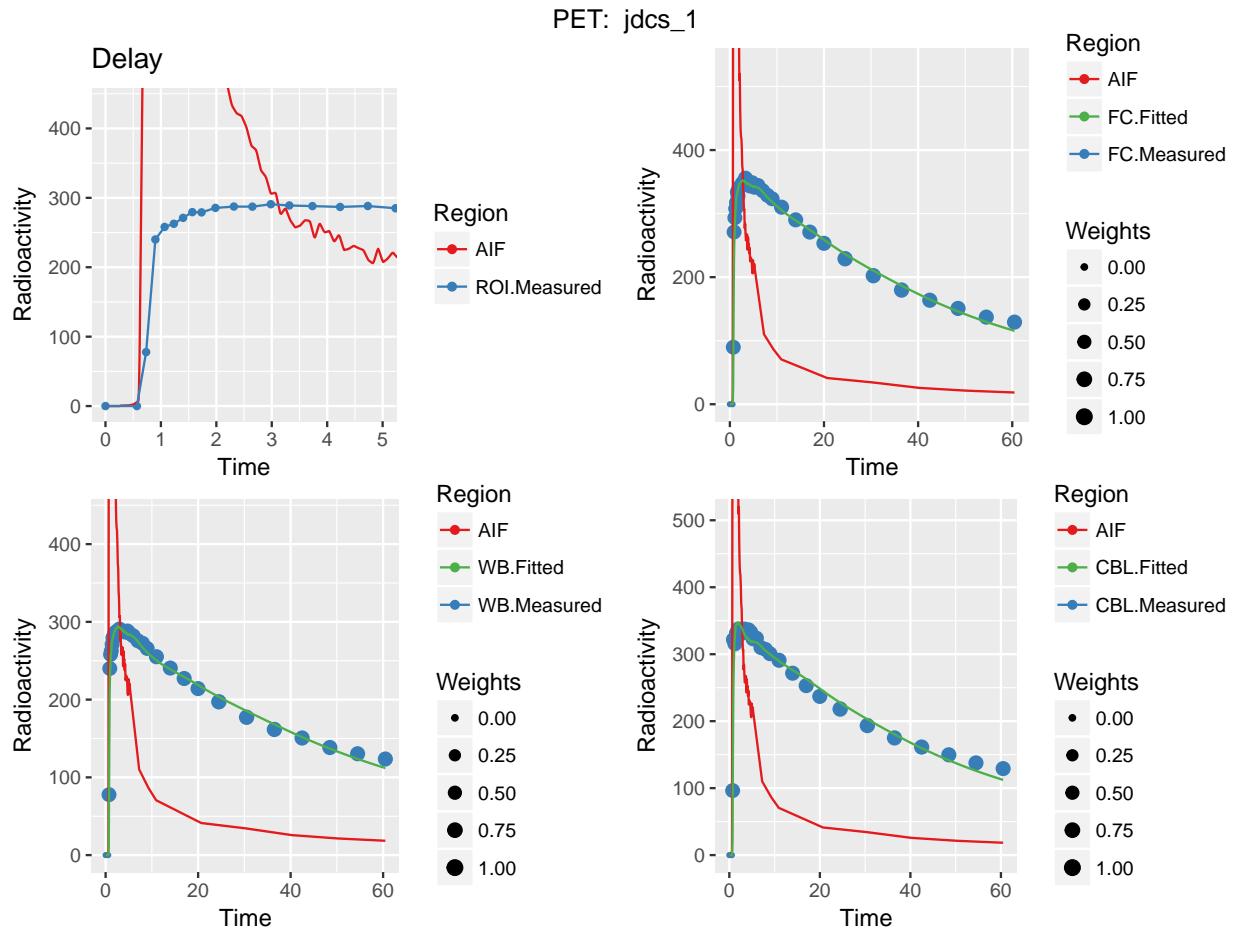
```

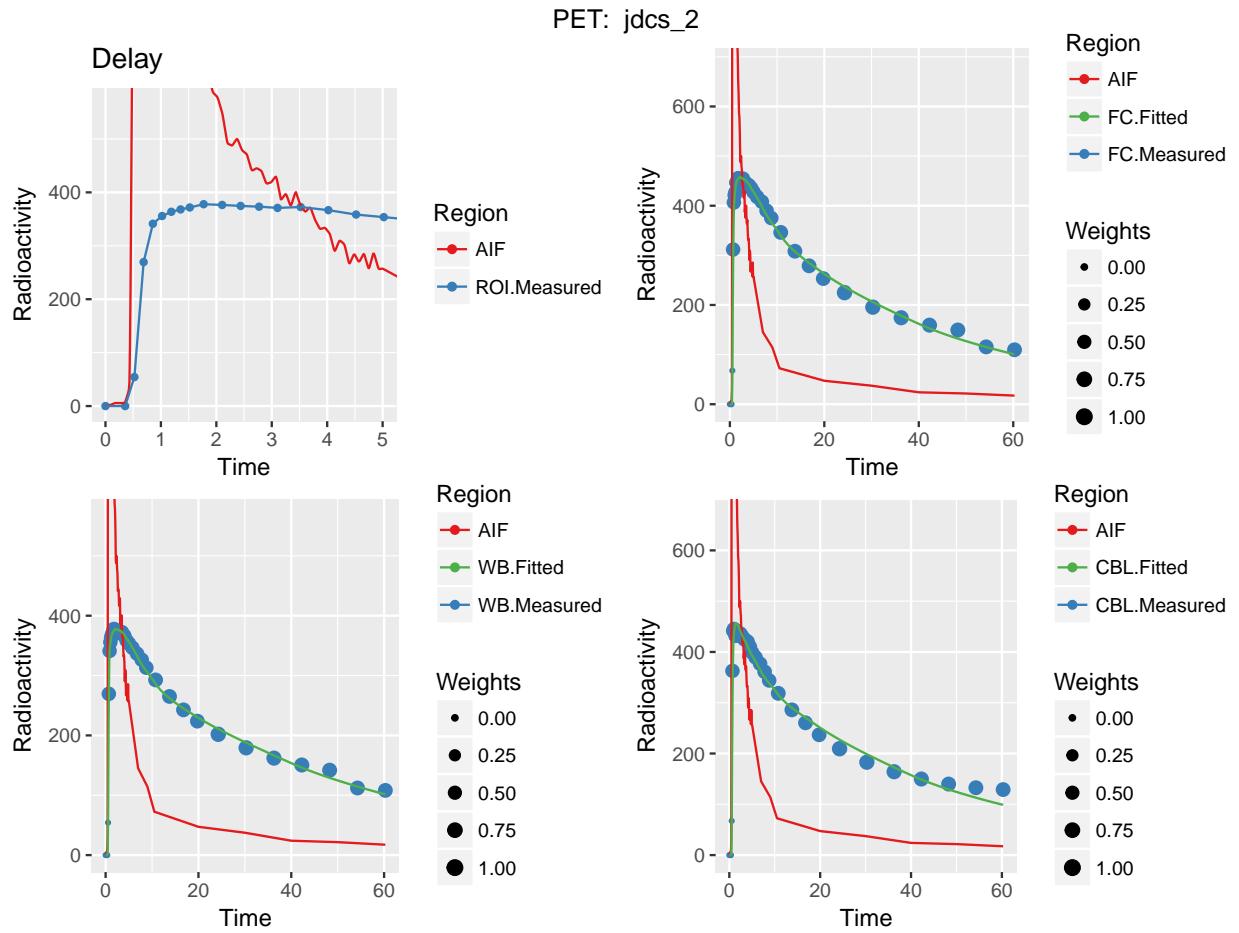


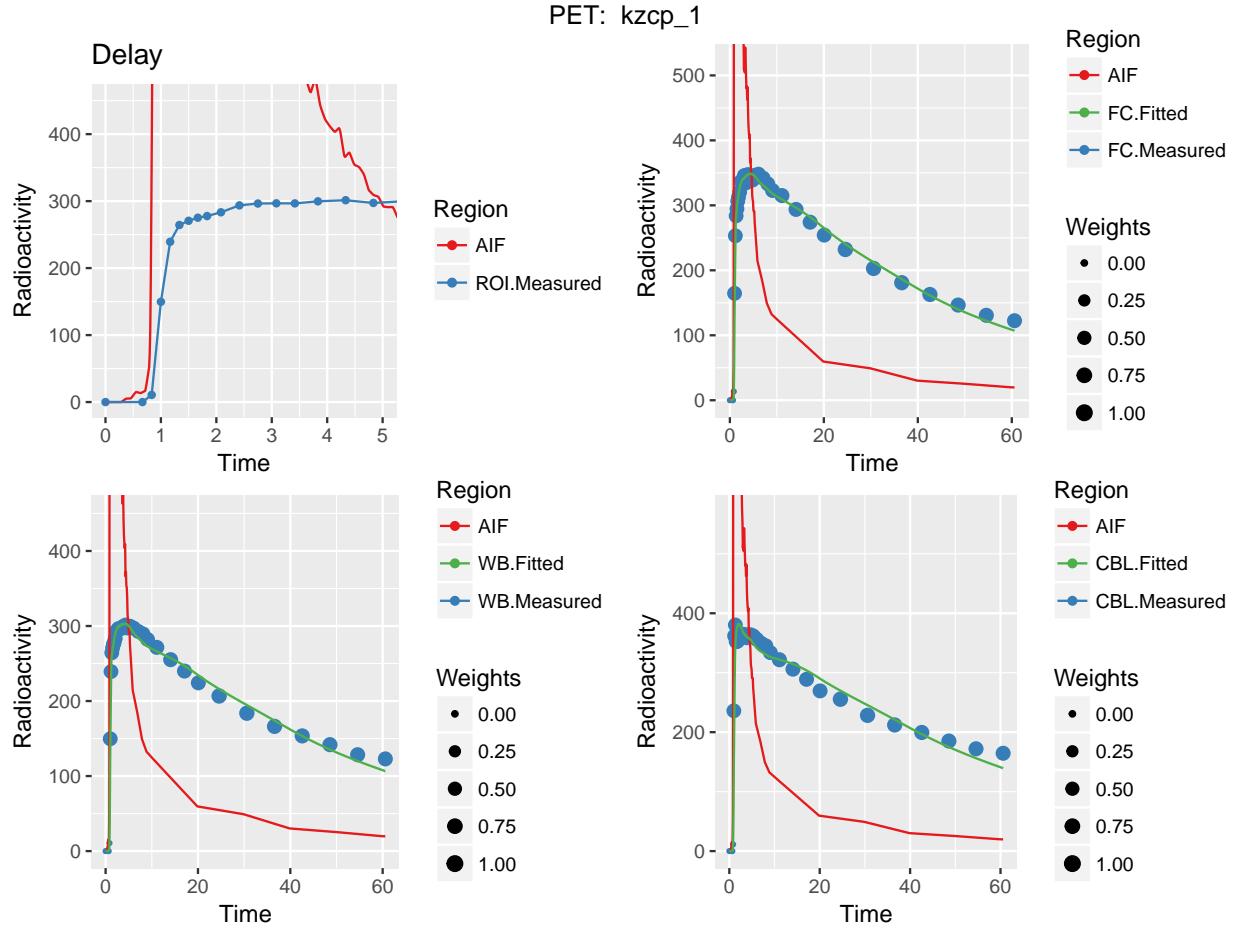


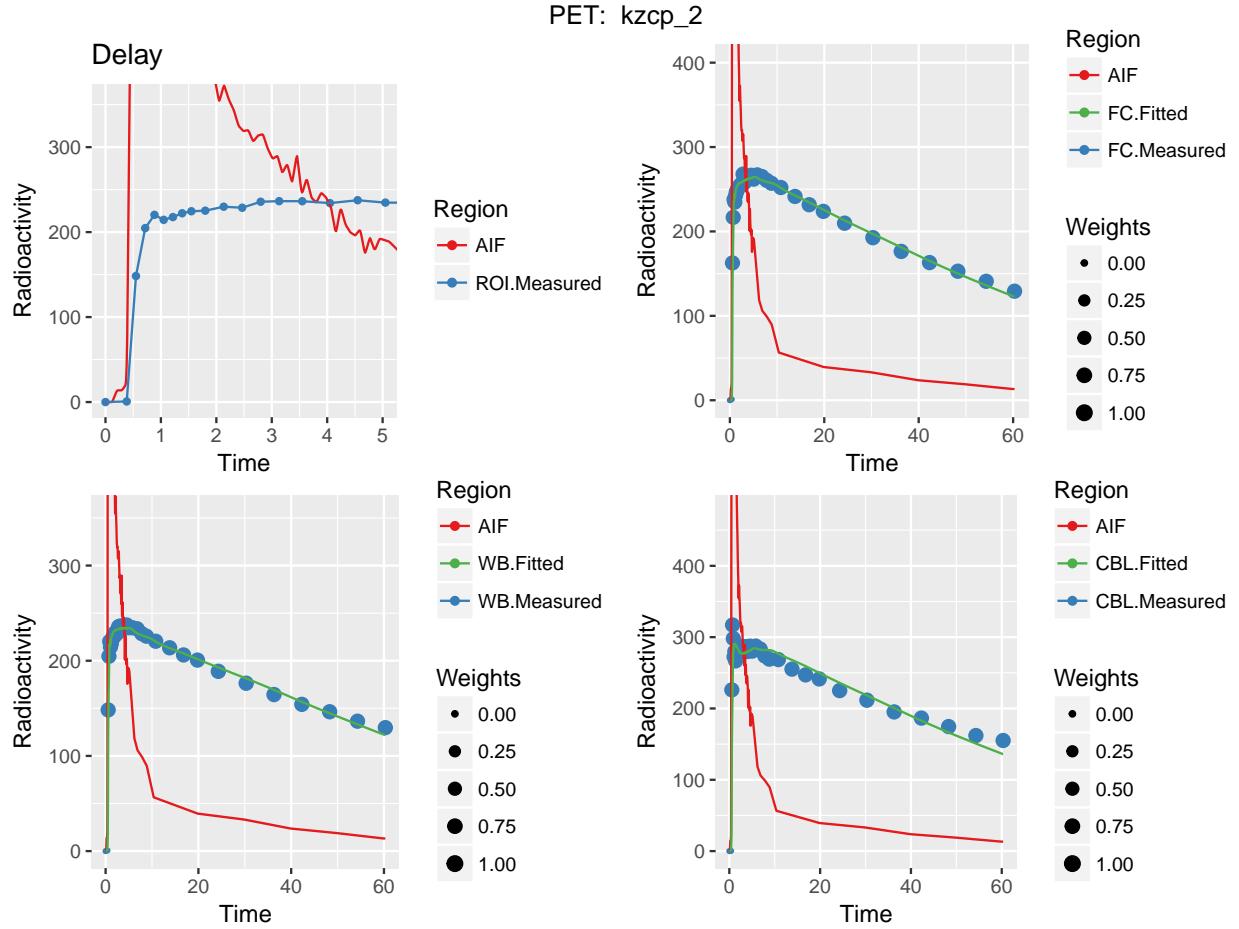


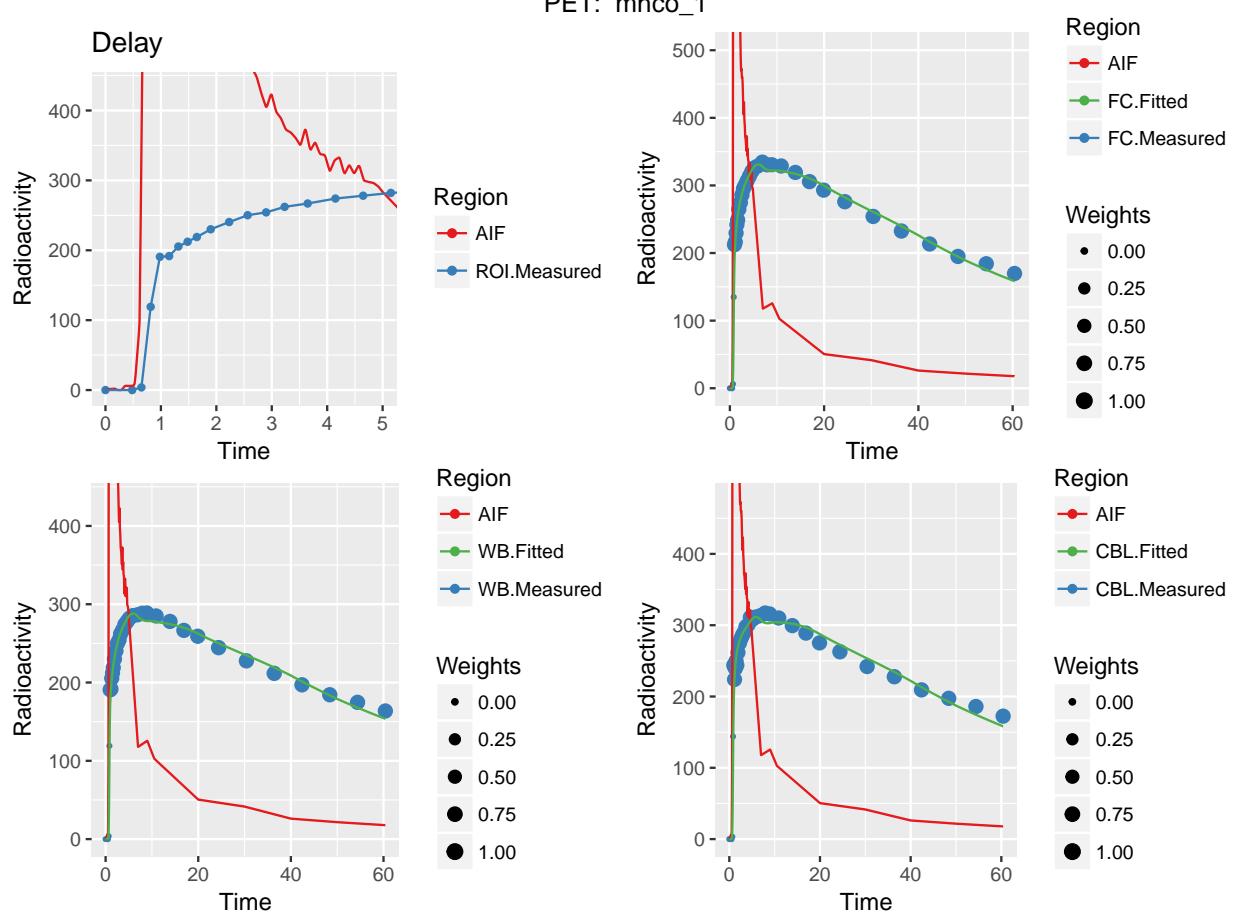


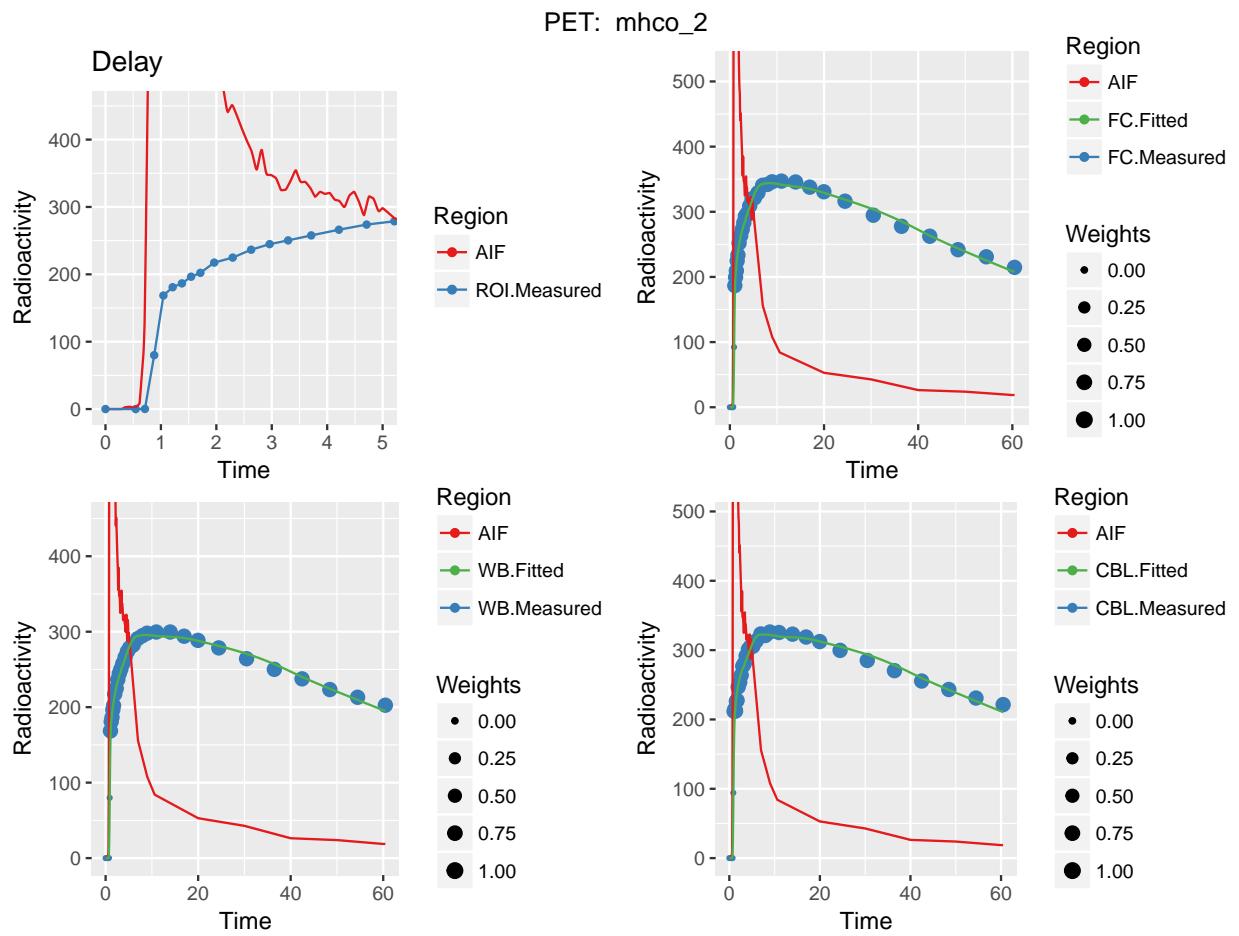


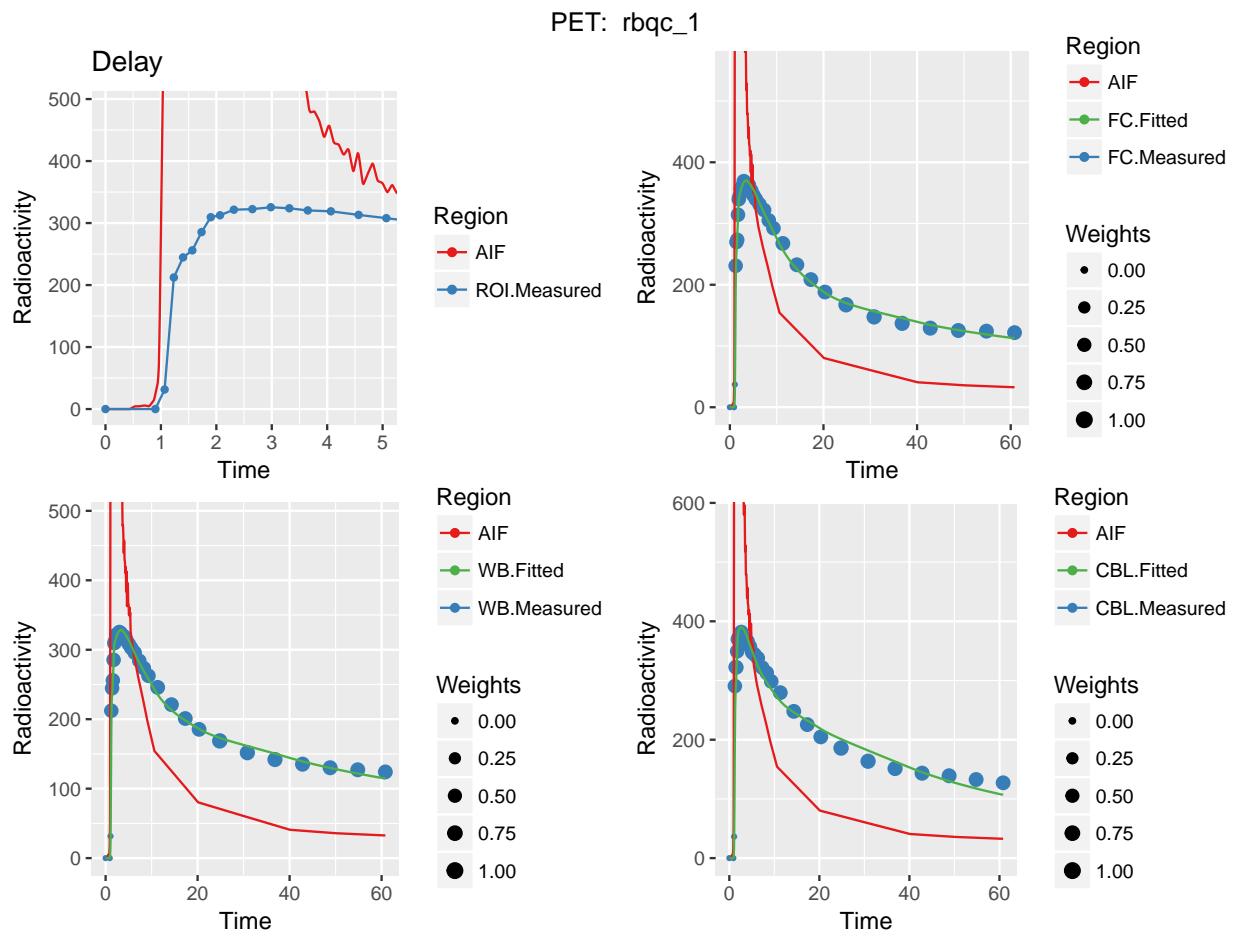


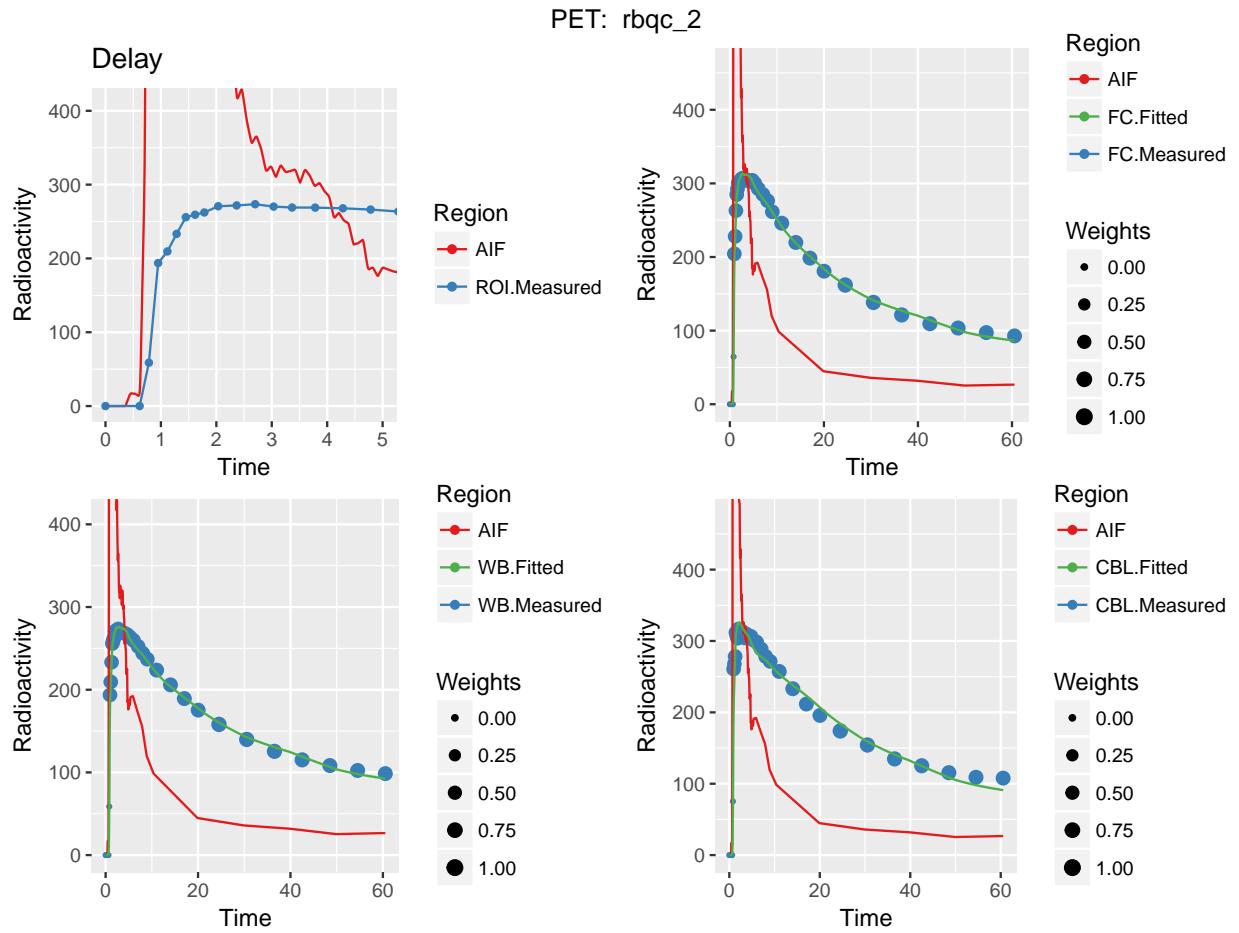


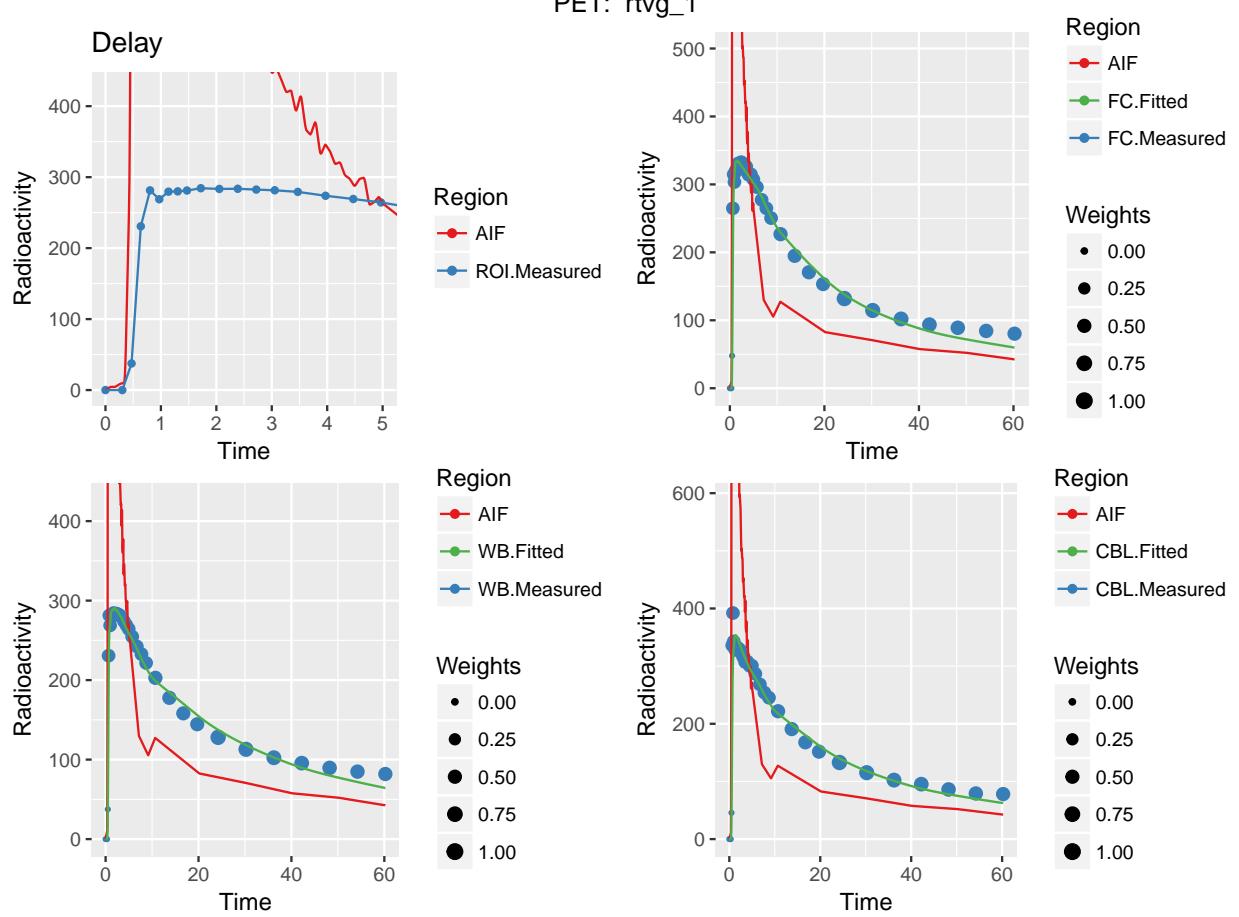


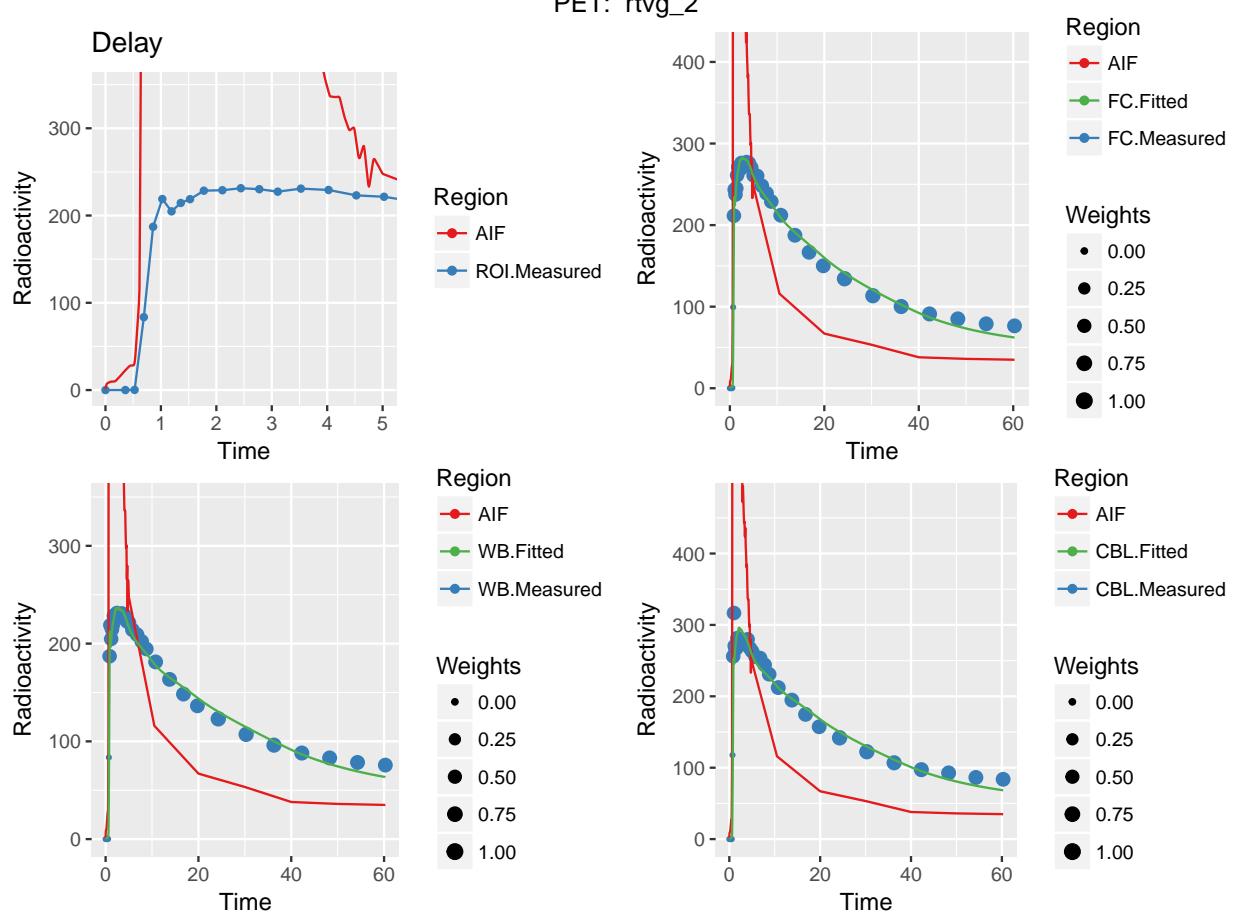


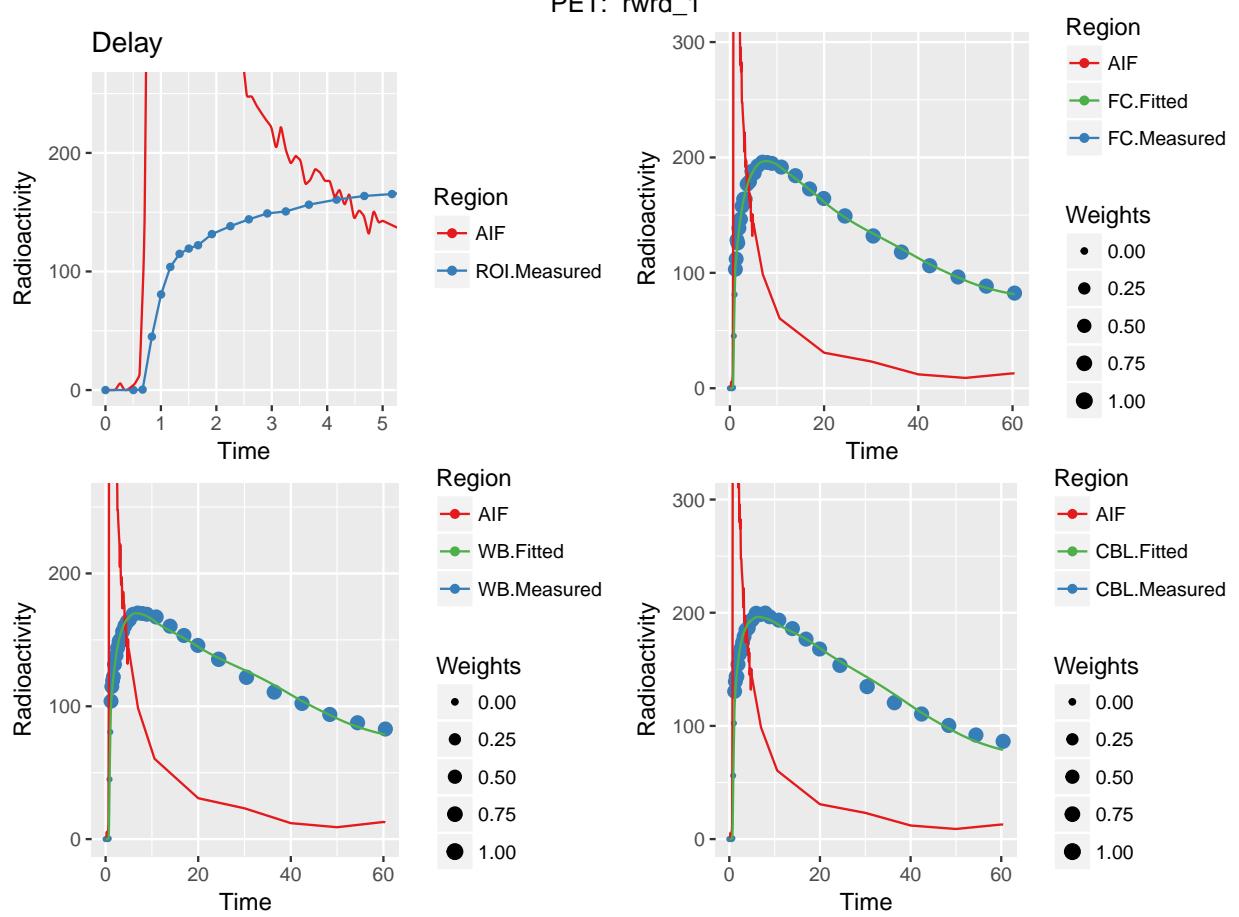


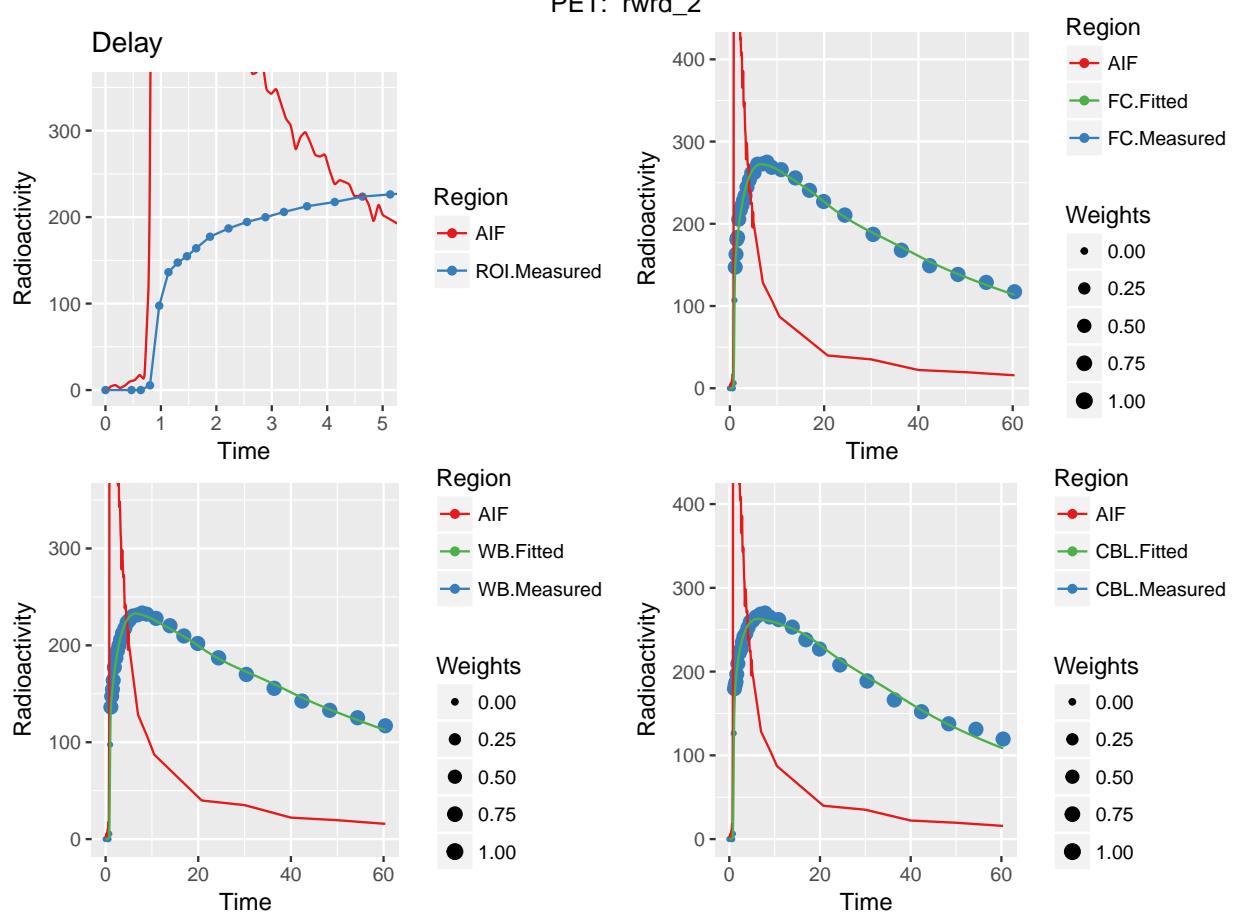


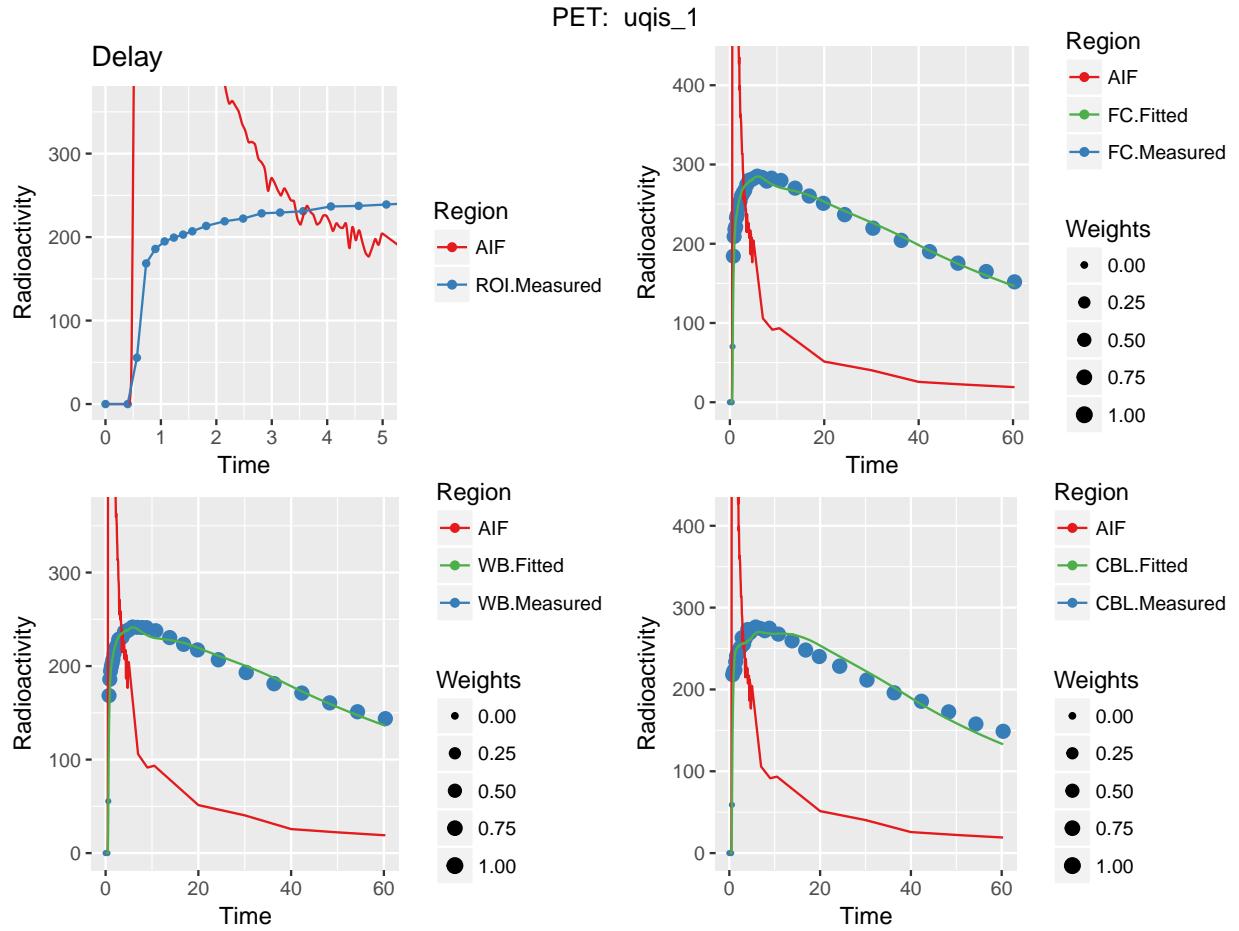


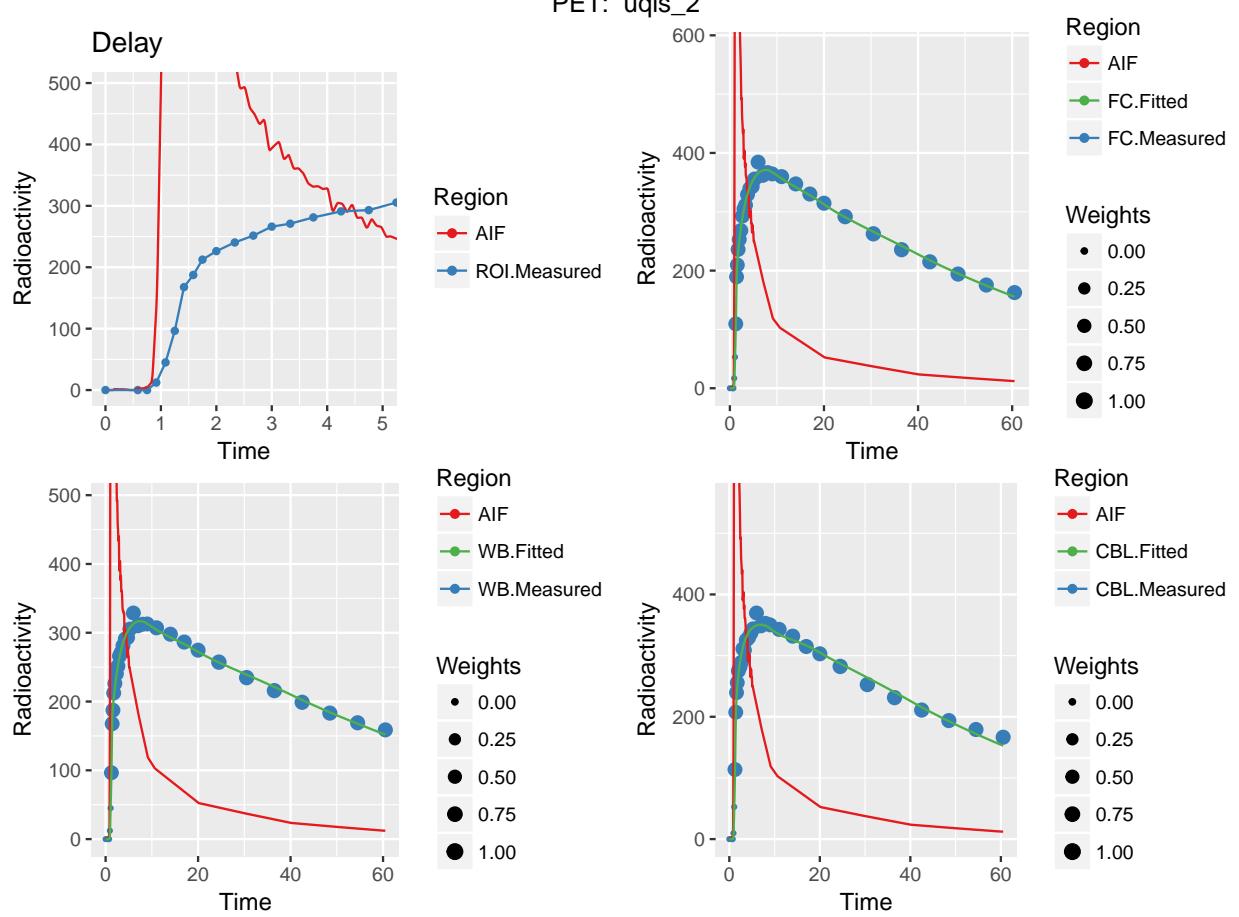


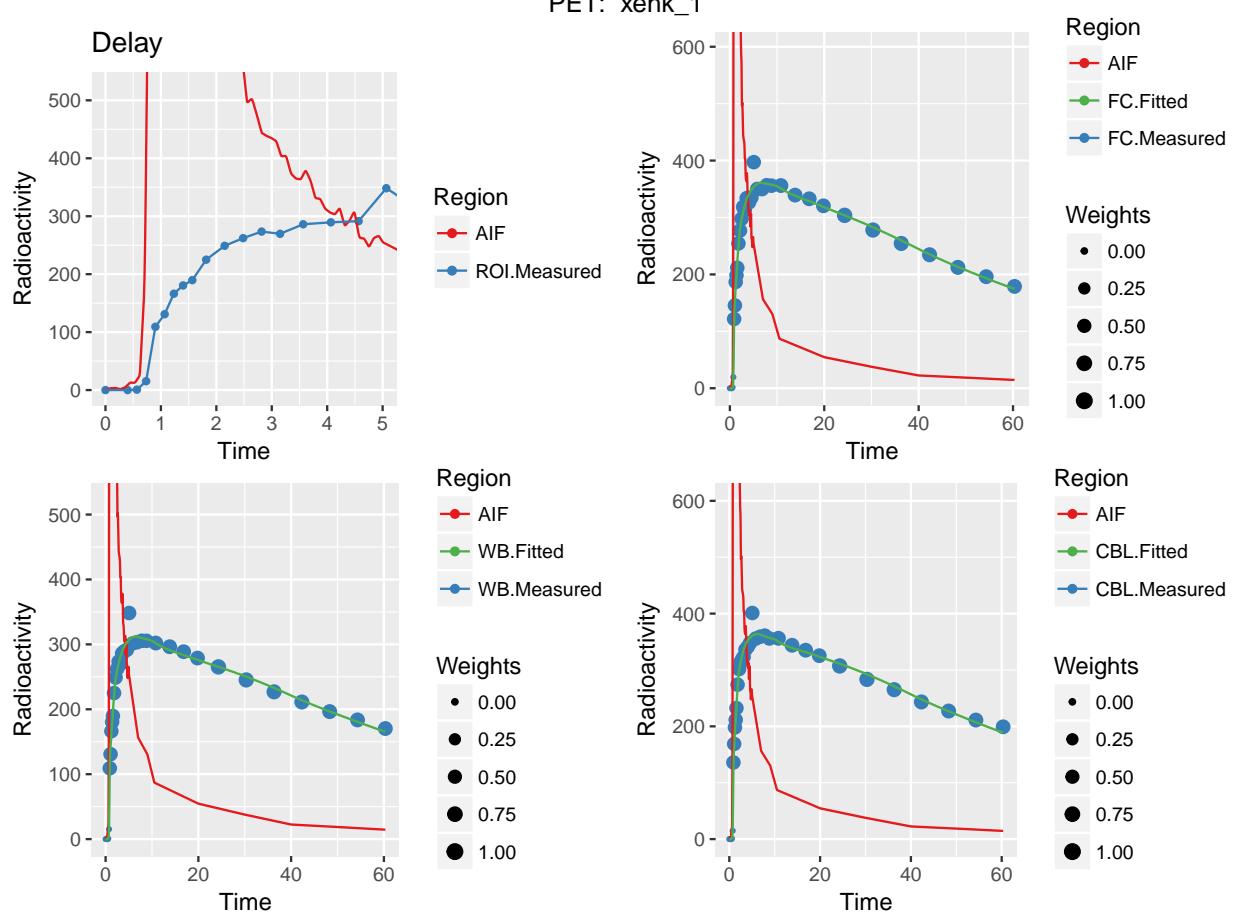


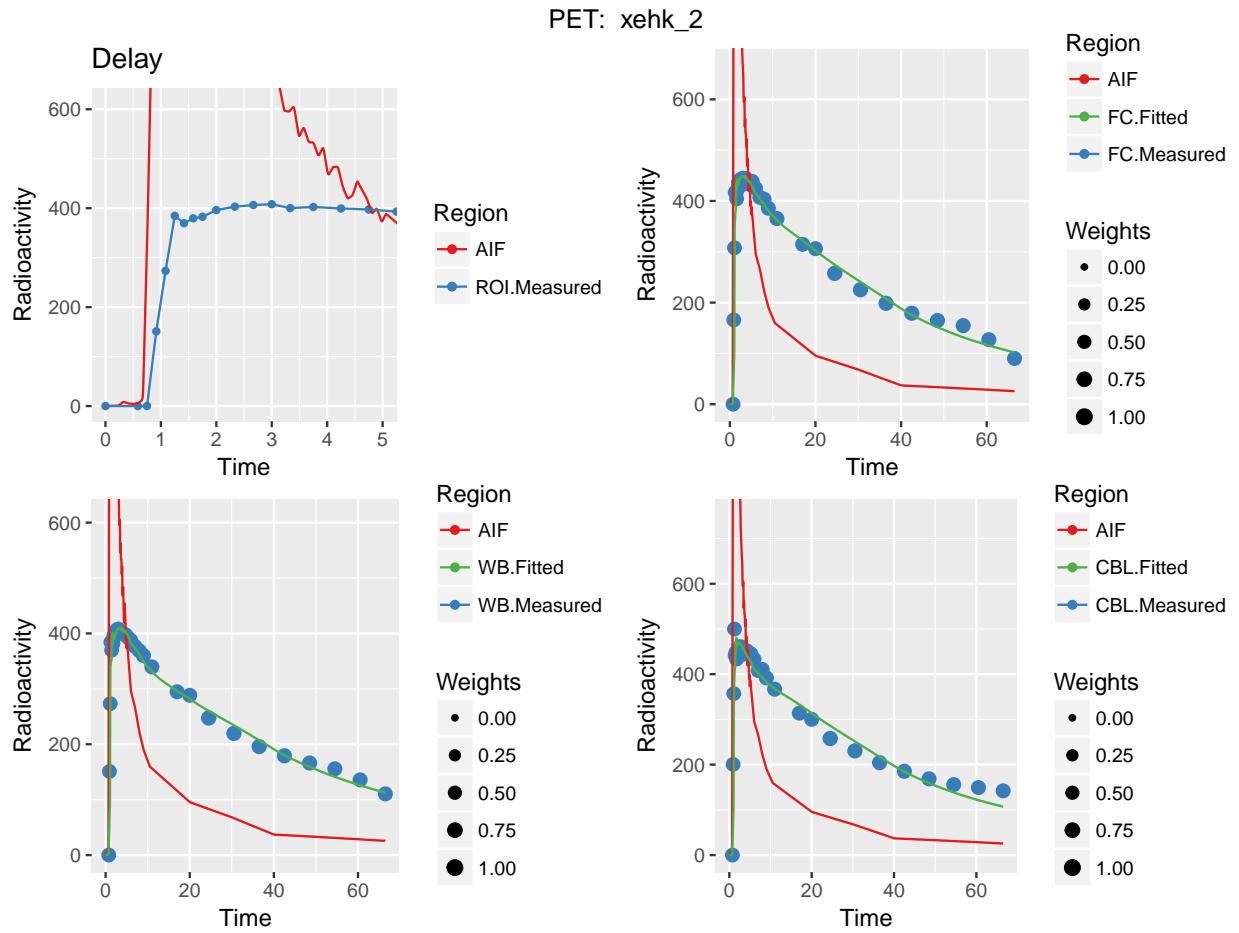


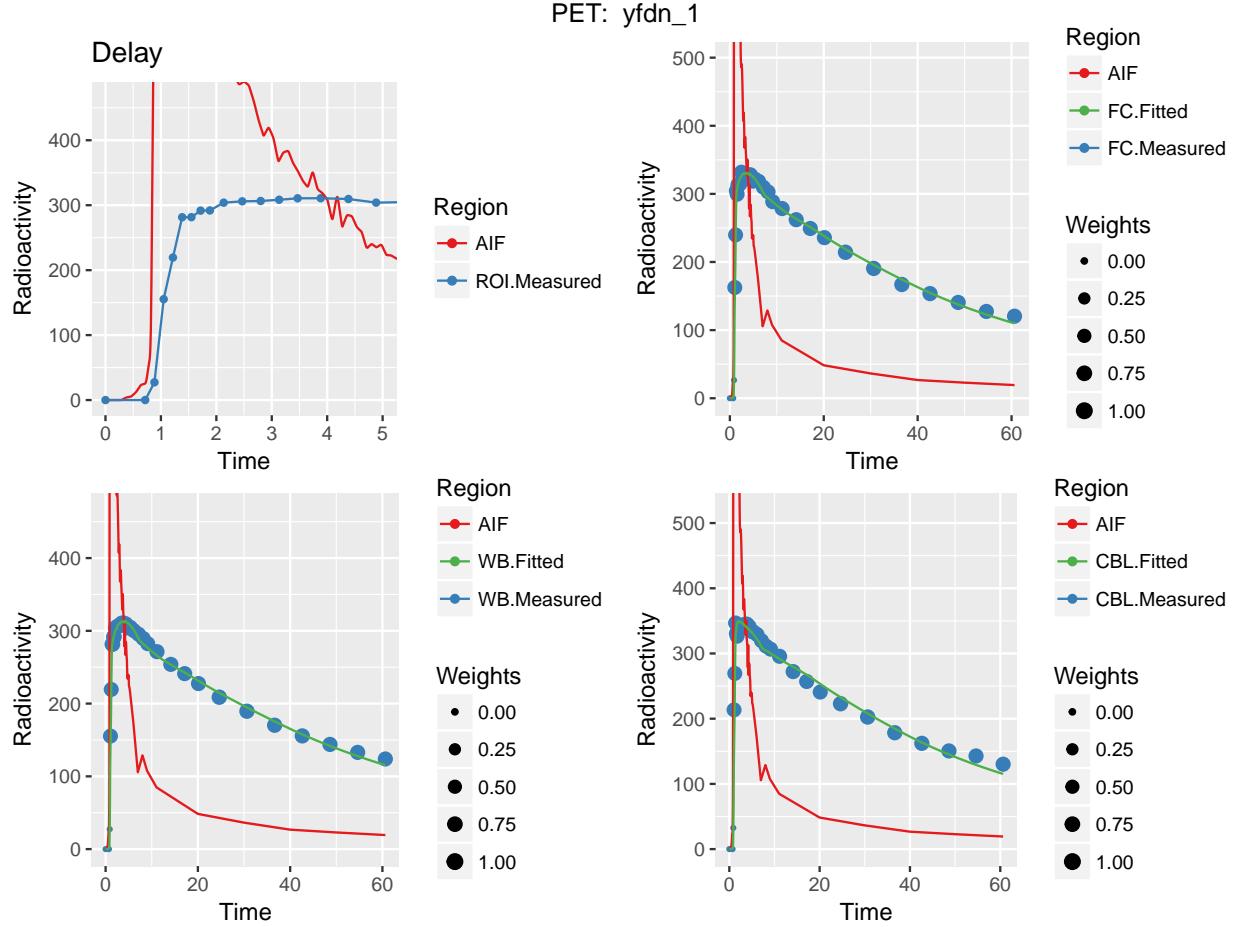


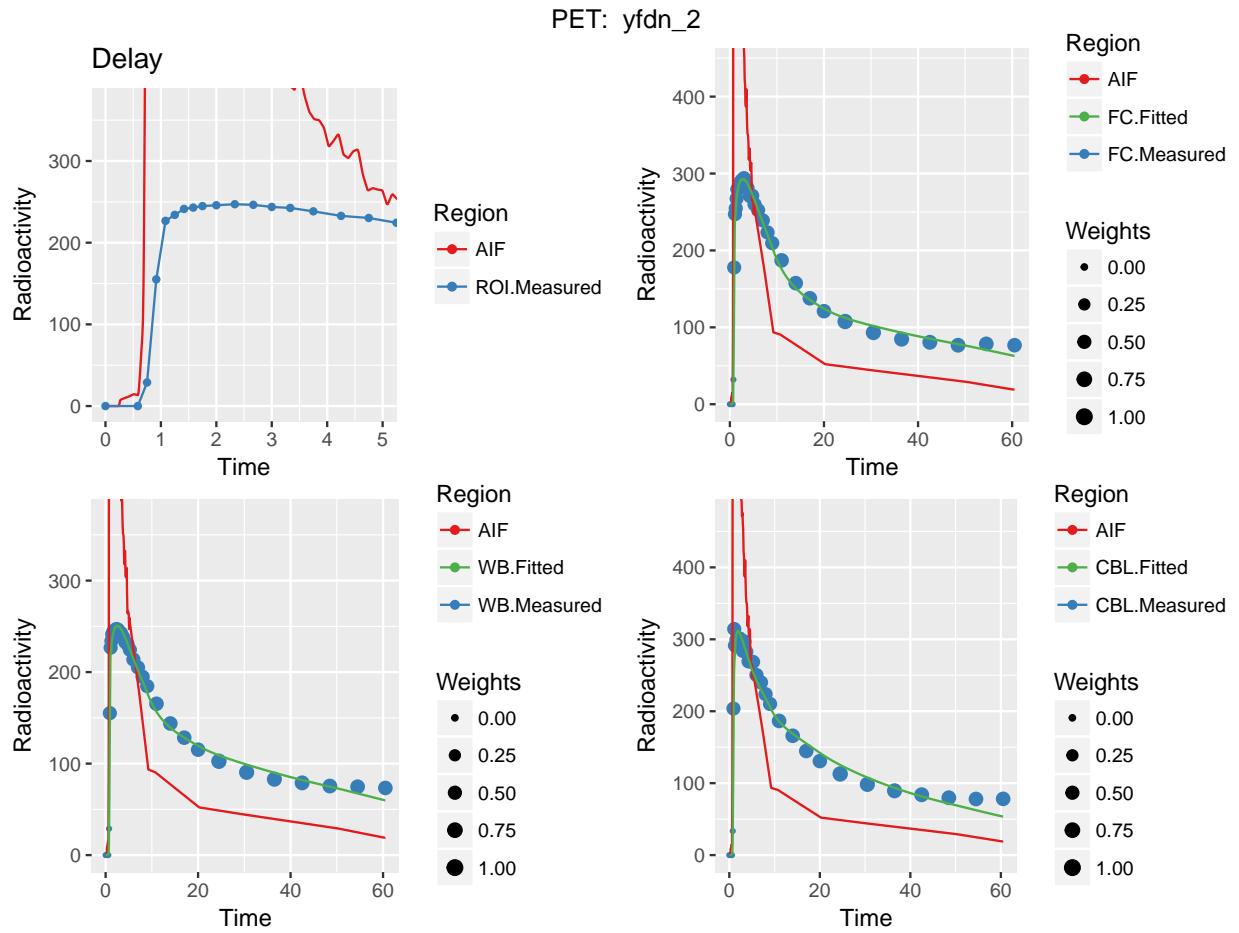


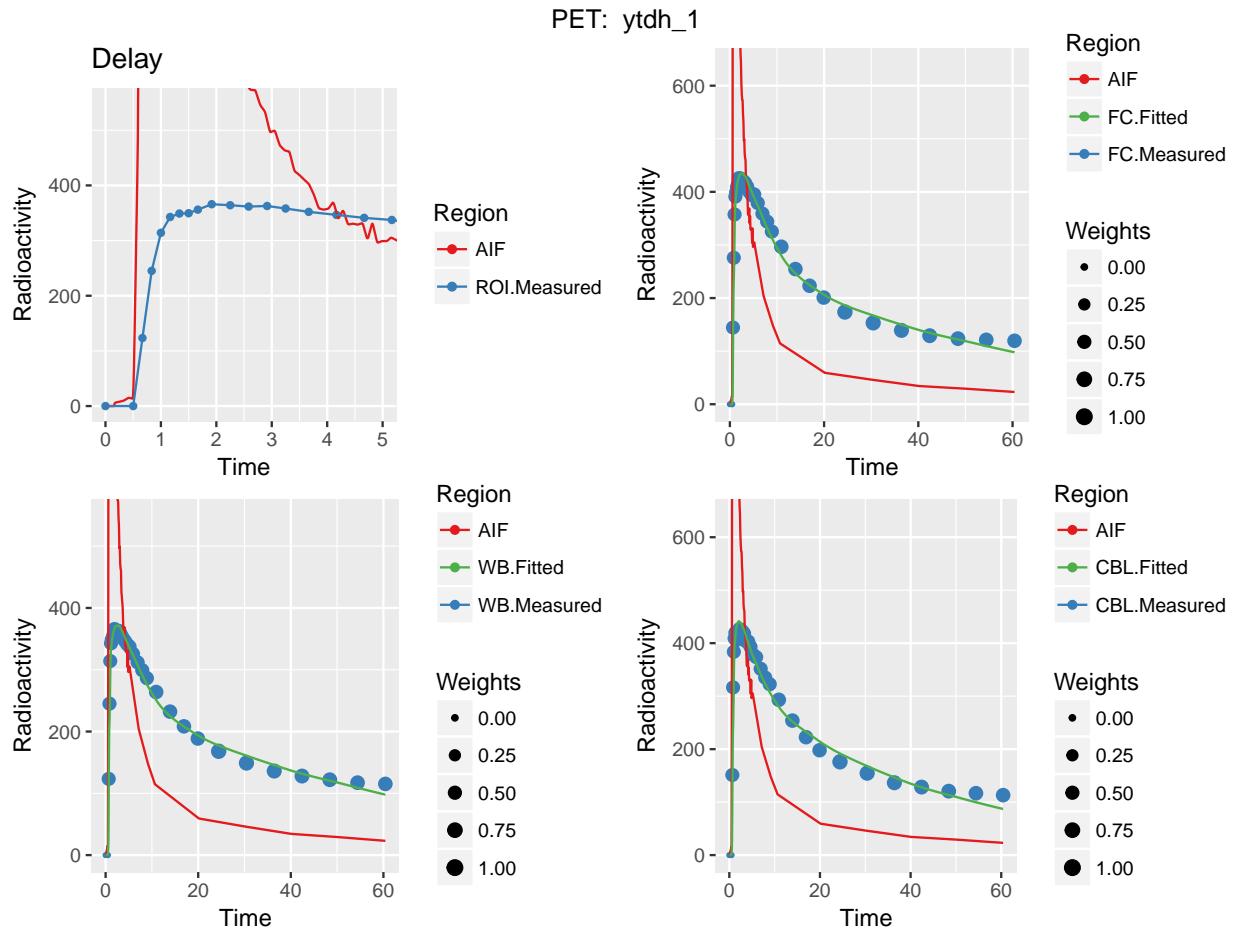






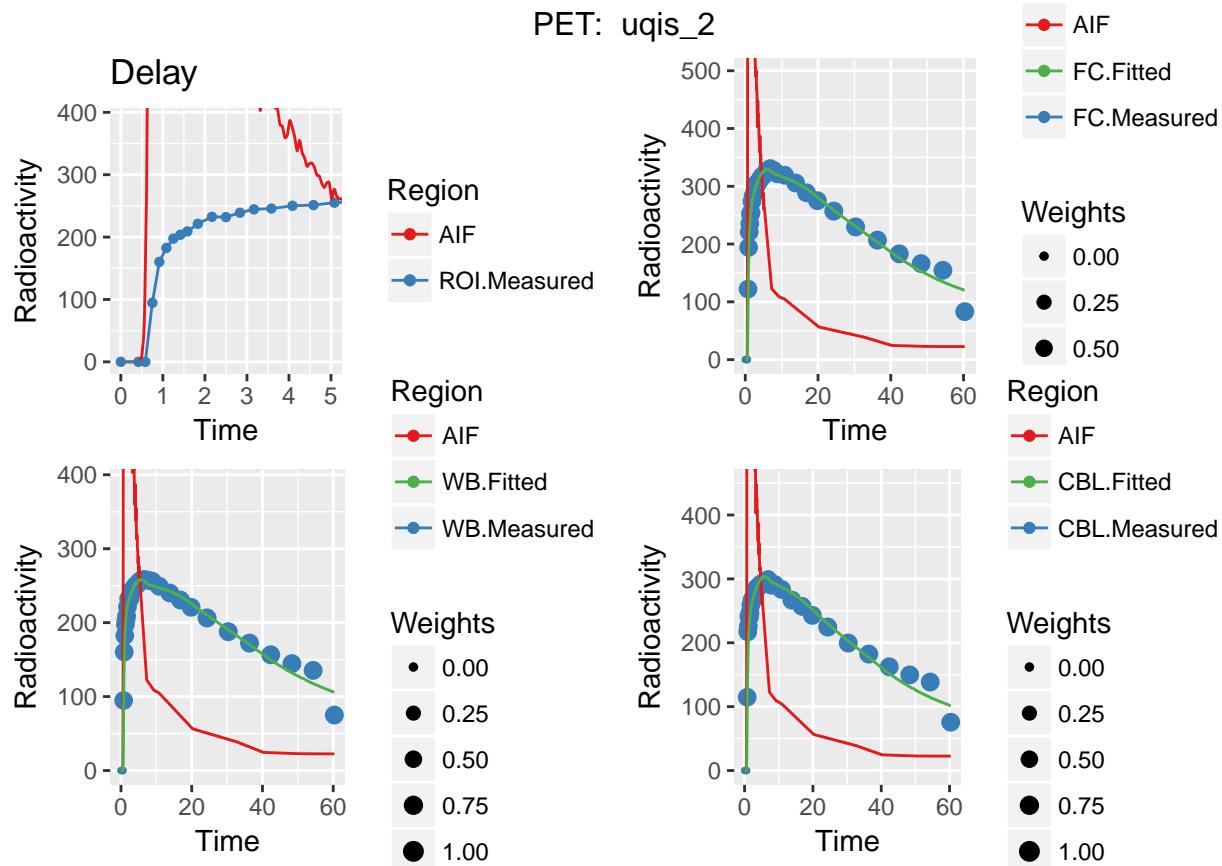






## Excluded Measurement

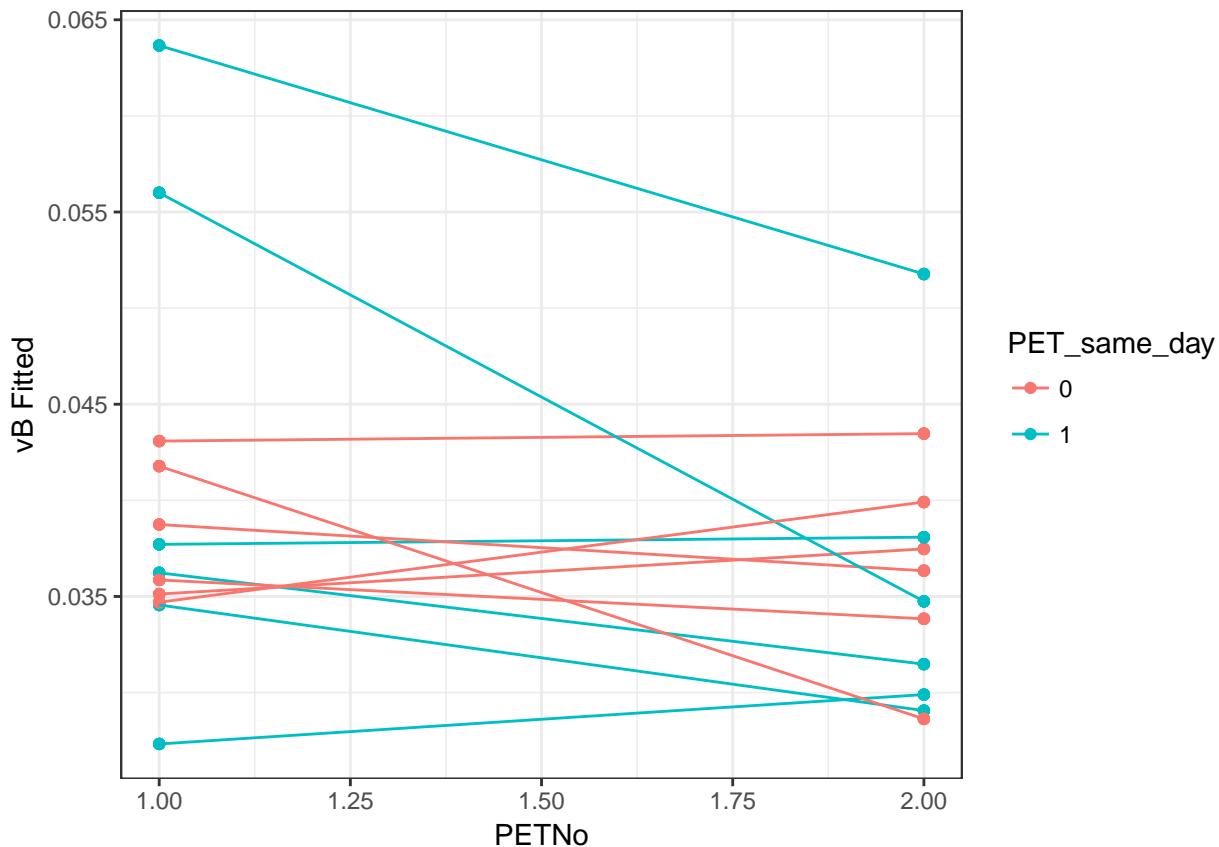
```
fitLabels_excl <- unique(allFits_excluded$PET)
marrangeGrob(allFits_excluded$Fit, nrow=2, ncol=2, top=quote(paste('PET: ', fitLabels_excl[g])))
```



## vB Checking

```
longdat %>%
  inner_join(demog) %>%
  mutate(PET_same_day = as.factor(PET_same_day)) %>%
  mutate(vB_fitted = map_dbl(fit_2tcm, c('par', 'vB'))) %>%
  ggplot(aes(y=vB_fitted, x=PETNo, group=Subjname, colour=PET_same_day)) +
  geom_point() + geom_line() +
  theme_bw() +
  labs(y='vB Fitted')

## Joining, by = c("PET", "Subjname", "PETNo", "bodyMass", "injRad", "Genotype")
```



```
vB_summary <- longdat %>%
  inner_join(demog) %>%
  filter(Region=='WB') %>%
  mutate(vB_fitted = map_dbl(fit_2tcm, c('par', 'vB'))) %>%
  group_by(Genotype) %>%
  summarise(vB_mean = mean(vB_fitted),
            vB_sd = sd(vB_fitted),
            vB_min = min(vB_fitted),
            vB_max = max(vB_fitted))

## Joining, by = c("PET", "Subjname", "PETNo", "bodyMass", "injRad", "Genotype")
pandoc.table(vB_summary, digits=2,
              caption='Fitted vB values for HABs and MABs')
```

Table 12: Fitted vB values for HABs and MABs

Genotype	vB_mean	vB_sd	vB_min	vB_max
HAB	0.039	0.0098	0.029	0.064
MAB	0.038	0.0076	0.027	0.056

## Session Info

```
sessionInfo()

## R version 3.4.1 (2017-06-30)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows >= 8 x64 (build 9200)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United Kingdom.1252
## [2] LC_CTYPE=English_United Kingdom.1252
## [3] LC_MONETARY=English_United Kingdom.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.1252
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] bindrcpp_0.2      pander_0.6.1      vmisc_0.3.5
## [4] kinfitr_0.2.0    pracma_2.0.7     readxl_1.0.0
## [7] psych_1.7.5      RColorBrewer_1.1-2 gridExtra_2.2.1
## [10] corrplot_0.77    stringr_1.2.0    dplyr_0.7.2
## [13] purrrr_0.2.3    readr_1.1.1     tidyverse_1.1.1
## [16] tibble_1.3.3     ggplot2_2.2.1    tidyverse_1.1.1
## [19] rmarkdown_1.6
##
## loaded via a namespace (and not attached):
## [1] minpack.lm_1.2-1      reshape2_1.4.2      haven_1.1.0
## [4] lattice_0.20-35       colorspace_1.3-2   htmltools_0.3.6
## [7] base64enc_0.1-3       yaml_2.1.14       rlang_0.1.2
## [10] foreign_0.8-69        glue_1.1.1       granviller_0.0.0.9000
## [13] modelr_0.1.1        bindr_0.1        plyr_1.8.4
## [16] munsell_0.4.3       gtable_0.2.0      cellranger_1.1.0
## [19] rvest_0.3.2         evaluate_0.10.1   labeling_0.3
## [22] knitr_1.17         forcats_0.2.0     parallel_3.4.1
## [25] broom_0.4.2         Rcpp_0.12.12     scales_0.4.1
## [28] backports_1.1.0     jsonlite_1.5     mnormt_1.5-5
## [31] hms_0.3             digest_0.6.12    stringi_1.1.5
## [34] rprojroot_1.2        quadprog_1.5-5   tools_3.4.1
## [37] magrittr_1.5         lazyeval_0.2.0   pkgconfig_2.0.1
## [40] xml2_1.1.1          lubridate_1.6.0  assertthat_0.2.0
## [43] httr_1.2.1          R6_2.2.2        nlme_3.1-131
## [46] compiler_3.4.1
```