R-code for the analysis I.) Consistency of the strain effect across replicate experiments

Please note that in the following code variables referring to the ‘conventional standardised design’ contain the labels ‘s’, ‘S’ or ‘Stand.’ and variables referring to the ‘mini-experiment design’ contain the labels ‘h’, ‘H’ or ‘Het.’.

setwd**(**"//nwz.wwu.de/dfs/Home/V/v\_vonk01/Vanessa\_I/PhD/Batchprojekt/Daten"**)**

data**<-**read.csv2**(**"Rawdata\_mini-experiment.csv"**)**

#library(nlme)

library**(**lme4**)**

library**(**lmerTest**)**

#library(car)

######### data table organisation #################

# two separate data tables are created, one for the conventional design and one for the mini-experiment design

#### (conventional = dS, mini-experiment = dH)

dS**=**data**[**data**$**Design**==**'Con ',**-**c**(**22**:**36**)]**

dH**=**data**[**data**$**Design**==**'Mini',**-**c**(**22**:**36**)]**

# make sure outcome measure columns are numbers

**for** **(**k **in** 6**:**21**){**

dS**[**,k**]<-**as.numeric**(**as.character**(**dS**[**,k**]))**

**}**

**for** **(**k **in** 6**:**21**){**

dH**[**,k**]<-**as.numeric**(**as.character**(**dH**[**,k**]))**

**}**

########## split data table in strain effects ######################

sBL6**<-**dS**[**dS**$**Strain**==**'C57BL/6',**]**

sDBA**<-**dS**[**dS**$**Strain**==**'DBA/2',**]**

sBALB**<-**dS**[**dS**$**Strain**==**'Balb/c',**]**

sB6D2F1**<-**dS**[**dS**$**Strain**==**'B6D2F1',**]**

hBL6**<-**dH**[**dH**$**Strain**==**'C57BL/6',**]**

hDBA**<-**dH**[**dH**$**Strain**==**'DBA/2',**]**

hBALB**<-**dH**[**dH**$**Strain**==**'Balb/c',**]**

hB6D2F1**<-**dH**[**dH**$**Strain**==**'B6D2F1',**]**

####### bind data table in strain effects #################

sBL6vsDBA**<-**rbind**(**sBL6,sDBA**)**

hBL6vsDBA**<-**rbind**(**hBL6,hDBA**)**

sDBAvsBDF**<-**rbind**(**sDBA,sB6D2F1**)**

hDBAvsBDF**<-**rbind**(**hDBA,hB6D2F1**)**

sBL6vsBALB**<-**rbind**(**sBL6,sBALB**)**

hBL6vsBALB**<-**rbind**(**hBL6,hBALB**)**

sBL6vsBDF**<-**rbind**(**sBL6,sB6D2F1**)**

hBL6vsBDF**<-**rbind**(**hBL6,hB6D2F1**)**

sDBAvsBALB**<-**rbind**(**sDBA,sBALB**)**

hDBAvsBALB**<-**rbind**(**hDBA,hBALB**)**

sBALBvsBDF**<-**rbind**(**sBALB,sB6D2F1**)**

hBALBvsBDF**<-**rbind**(**hBALB,hB6D2F1**)**

############################ Analysis: strain x replicate interaction for 1. Strain effect - BL6 vs DBA - #####################################

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

# define model variables

hBL6vsDBA**$**replicate**=**as.factor**(**hBL6vsDBA**$**Replicate.experiment**)**

hBL6vsDBA**$**block**=**as.factor**(**hBL6vsDBA**$**Block**)**

hBL6vsDBA**$**Strain**=**as.factor**(**hBL6vsDBA**$**Strain**)**

sBL6vsDBA**$**replicate**=**as.factor**(**sBL6vsDBA**$**Replicate.experiment**)**

sBL6vsDBA**$**block**=**as.factor**(**sBL6vsDBA**$**Block**)**

sBL6vsDBA**$**Strain**=**as.factor**(**sBL6vsDBA**$**Strain**)**

# define outcome measure

parS**<-**sBL6vsDBA**$**HCB.percentage.active

parH**<-**hBL6vsDBA**$**HCB.percentage.active

####### overview full linear model for both designs #######

########### linear model for conventional design ####################

lmS**=**lmer**(**parS **~** Strain **+** **(**1**|**block**)** **+** **(**1**|**block**:**Strain**)** **+** **(**1**|**replicate**)** **+** **(**1**|**Strain**:**replicate**)**, data**=**sBL6vsDBA**)**

############## linear model for mini-experiment design ##################

lmH**=**lmer**(**parH **~** Strain **+** **(**1**|**block**)** **+** **(**1**|**block**:**Strain**)** **+** **(**1**|**replicate**)** **+** **(**1**|**Strain**:**replicate**)**,data**=**hBL6vsDBA**)**

######### transformation of outcome measures #########

sBL6vsDBA**$**HCB.drink.feed**<-**sqrt**(**sBL6vsDBA**$**HCB.drink.feed**)**

sBL6vsDBA**$**HCB.climbing**<-**sqrt**(**sBL6vsDBA**$**HCB.climbing**)**

sBL6vsDBA**$**EPM.OA.rel..Time**<-**sqrt**(**sBL6vsDBA**$**EPM.OA.rel..Time**)**

sBL6vsDBA**$**OF.Center.time**<-**sqrt**(**sBL6vsDBA**$**OF.Center.time**)**

sBL6vsDBA**$**Barrier.Latency..s.**<-**log10**(**sBL6vsDBA**$**Barrier.Latency..s.**+**1**)**

sBL6vsDBA**$**FCM**<-**sqrt**(**sBL6vsDBA**$**FCM**)**

hBL6vsDBA**$**HCB.drink.feed**<-**sqrt**(**hBL6vsDBA**$**HCB.drink.feed**)**

hBL6vsDBA**$**HCB.climbing**<-**sqrt**(**hBL6vsDBA**$**HCB.climbing**)**

hBL6vsDBA**$**EPM.OA.rel..Time**<-**sqrt**(**hBL6vsDBA**$**EPM.OA.rel..Time**)**

hBL6vsDBA**$**OF.Center.time**<-**sqrt**(**hBL6vsDBA**$**OF.Center.time**)**

hBL6vsDBA**$**Barrier.Latency..s.**<-**log10**(**hBL6vsDBA**$**Barrier.Latency..s.**+**1**)**

hBL6vsDBA**$**FCM**<-**sqrt**(**hBL6vsDBA**$**FCM**)**

###### loop for linear model analysis of all outcome measures ###########

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

interaction**=**matrix**(NA**,nrow **=** 16,ncol **=** 4**)** #create empty matrix to fill in the p-values, NA to detect errors when something is missing

colnames**(**interaction**)=**c**(**'Stand.int','Het.int','Stand.main','Het.main'**)**

rownames**(**interaction**)=**names**(**dH**)[-**c**(**1**:**5,22**:**23**)]**

Chi\_sq**=**matrix**(NA**,nrow **=** 16,ncol **=** 4**)** #create empty matrix to fill in the Chi\_sq-values, NA to detect errors when something is missing

colnames**(**Chi\_sq**)=**c**(**'Stand.int','Het.int','Stand.main','Het.main'**)**

rownames**(**Chi\_sq**)=**names**(**dH**)[-**c**(**1**:**5,22**:**23**)]**

**for** **(** k **in** 6**:**21**){**

# conventional design

lm0**=**lmer**(**sBL6vsDBA**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)+(**1**|**Strain**:**replicate**)**,data**=**sBL6vsDBA**)** #first model with all factors

lm1**=**lmer**(**sBL6vsDBA**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**sBL6vsDBA**)** #second model with the 'strain x replicate' interaction is missing to test their significant influence against the first model

lm2**=**lmer**(**sBL6vsDBA**[**,k**]~(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**sBL6vsDBA**)** #third model without 'strain', to test strain effect

interaction**[**k**-**5,c**(**1,3**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,8**])** #test the three models against each other and create a matrix out of it

Chi\_sq**[**k**-**5,c**(**1,3**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,6**])**

#fill interaction-matrix (column 1 and 3), vector in the end: reorganising p-values

# mini-experiment design

lm0**=**lmer**(**hBL6vsDBA**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)+(**1**|**Strain**:**replicate**)**,data**=**hBL6vsDBA**)** #first model with all factors

lm1**=**lmer**(**hBL6vsDBA**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**hBL6vsDBA**)** #second model with the'strain x replicate' interaction is missing to test their significant influence against the first model

lm2**=**lmer**(**hBL6vsDBA**[**,k**]~(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**hBL6vsDBA**)** #third model without 'strain', to test strain effect

interaction**[**k**-**5,c**(**2,4**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,8**])**

Chi\_sq**[**k**-**5,c**(**2,4**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,6**])**

**}**

View**(**interaction**)** #show interaction matrix

View**(**Chi\_sq**)**

# visualisation of main strain effect and strain x replicate interaction for both designs

boxplot**(**interaction**)**

# visualisation of strain x replicate interaction for both designs

boxplot**(**interaction**[**,c**(**1,2**)])**

#pairwise comparison of p-values from the 'strain x replicate interaction term' between both experimental designs

wilcox.test**(**interaction**[**,1**]**,interaction**[**,2**]**,paired **=** **TRUE)**

## Note: gives out p-value for two sided wilcox.test, has to be dived by 2 for one-sided hypothesis! ###

############################ Analysis: strain x replicate interaction for 2. Strain effect - BL6 vs BALB - #####################################

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

# define model variables

hBL6vsBALB**$**replicate**=**as.factor**(**hBL6vsBALB**$**Replicate.experiment**)**

hBL6vsBALB**$**block**=**as.factor**(**hBL6vsBALB**$**Block**)**

hBL6vsBALB**$**Strain**=**as.factor**(**hBL6vsBALB**$**Strain**)**

sBL6vsBALB**$**replicate**=**as.factor**(**sBL6vsBALB**$**Replicate.experiment**)**

sBL6vsBALB**$**block**=**as.factor**(**sBL6vsBALB**$**Block**)**

sBL6vsBALB**$**Strain**=**as.factor**(**sBL6vsBALB**$**Strain**)**

# define outcome measure

parS**<-**sBL6vsBALB**$**Diff.weight.13.10

parH**<-**hBL6vsBALB**$**Diff.weight.13.10

####### overview full linear model for both designs #######

########### linear model for conventional design ####################

lmS**=**lmer**(**parS **~** Strain **+** **(**1**|**block**)** **+** **(**1**|**block**:**Strain**)** **+** **(**1**|**replicate**)** **+** **(**1**|**Strain**:**replicate**)**, data**=**sBL6vsBALB**)**

############## linear model for mini-experiment design ##################

lmH**=**lmer**(**parH **~** Strain **+** **(**1**|**block**)** **+** **(**1**|**block**:**Strain**)** **+** **(**1**|**replicate**)** **+** **(**1**|**Strain**:**replicate**)**,data**=**hBL6vsBALB**)**

######### transformation of outcome measures #########

sBL6vsBALB**$**HCB.drink.feed**<-**sqrt**(**sBL6vsBALB**$**HCB.drink.feed**)**

sBL6vsBALB**$**HCB.climbing**<-**sqrt**(**sBL6vsBALB**$**HCB.climbing**)**

sBL6vsBALB**$**EPM.OA.rel..Time**<-**sqrt**(**sBL6vsBALB**$**EPM.OA.rel..Time**)**

sBL6vsBALB**$**Barrier.Latency..s.**<-**1**/(**sBL6vsBALB**$**Barrier.Latency..s.**+**1**)**

sBL6vsBALB**$**FCM**<-**log10**(**sBL6vsBALB**$**FCM**+**1**)**

hBL6vsBALB**$**HCB.drink.feed**<-**sqrt**(**hBL6vsBALB**$**HCB.drink.feed**)**

hBL6vsBALB**$**HCB.climbing**<-**sqrt**(**hBL6vsBALB**$**HCB.climbing**)**

hBL6vsBALB**$**EPM.OA.rel..Time**<-**sqrt**(**hBL6vsBALB**$**EPM.OA.rel..Time**)**

hBL6vsBALB**$**Barrier.Latency..s.**<-**1**/(**hBL6vsBALB**$**Barrier.Latency..s.**+**1**)**

hBL6vsBALB**$**FCM**<-**log10**(**hBL6vsBALB**$**FCM**+**1**)**

###### loop for linear model analysis of all outcome measures ###########

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

interaction**=**matrix**(NA**,nrow **=** 16,ncol **=** 4**)** #create empty matrix to fill in the p-values, NA to detect errors when something is missing

colnames**(**interaction**)=**c**(**'Stand.int','Het.int','Stand.main','Het.main'**)**

rownames**(**interaction**)=**names**(**dH**)[-**c**(**1**:**5,22**:**23**)]**

Chi\_sq**=**matrix**(NA**,nrow **=** 16,ncol **=** 4**)** #create empty matrix to fill in the Chi\_sq-values, NA to detect errors when something is missing

colnames**(**Chi\_sq**)=**c**(**'Stand.int','Het.int','Stand.main','Het.main'**)**

rownames**(**Chi\_sq**)=**names**(**dH**)[-**c**(**1**:**5,22**:**23**)]**

**for** **(** k **in** 6**:**21**){**

# conventional design

lm0**=**lmer**(**sBL6vsBALB**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)+(**1**|**Strain**:**replicate**)**,data**=**sBL6vsBALB**)** #first model with all factors

lm1**=**lmer**(**sBL6vsBALB**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**sBL6vsBALB**)** #second model with the 'strain x replicate' interaction is missing to test their significant influence against the first model

lm2**=**lmer**(**sBL6vsBALB**[**,k**]~(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**sBL6vsBALB**)** #third model without 'strain', to test strain effect

interaction**[**k**-**5,c**(**1,3**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,8**])** #test the three models against each other and create a matrix out of it

Chi\_sq**[**k**-**5,c**(**1,3**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,6**])**

#fill interaction-matrix (column 1 and 3), vector in the end: reorganising p-values

# mini-experiment design

lm0**=**lmer**(**hBL6vsBALB**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)+(**1**|**Strain**:**replicate**)**,data**=**hBL6vsBALB**)** #first model with all factors

lm1**=**lmer**(**hBL6vsBALB**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**hBL6vsBALB**)** #second model with the'strain x replicate' interaction is missing to test their significant influence against the first model

lm2**=**lmer**(**hBL6vsBALB**[**,k**]~(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**hBL6vsBALB**)** #third model without 'strain', to test strain effect

interaction**[**k**-**5,c**(**2,4**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,8**])**

Chi\_sq**[**k**-**5,c**(**2,4**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,6**])**

**}**

View**(**interaction**)** #show interaction matrix

View**(**Chi\_sq**)**

# visualisation of main strain effect and strain x replicate interaction for both designs

boxplot**(**interaction**)**

# visualisation of strain x replicate interaction for both designs

boxplot**(**interaction**[**,c**(**1,2**)])**

#pairwise comparison of p-values from the 'strain x replicate interaction term' between both experimental designs

wilcox.test**(**interaction**[**,1**]**,interaction**[**,2**]**,paired **=** **TRUE)**

## Note: gives out p-value for two sided wilcox.test, has to be dived by 2 for one-sided hypothesis! ###

############################ Analysis: strain x replicate interaction for 3. Strain effect - BL6 vs B6D2F1 - #####################################

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

# define model variables

hBL6vsBDF**$**replicate**=**as.factor**(**hBL6vsBDF**$**Replicate.experiment**)**

hBL6vsBDF**$**block**=**as.factor**(**hBL6vsBDF**$**Block**)**

hBL6vsBDF**$**Strain**=**as.factor**(**hBL6vsBDF**$**Strain**)**

sBL6vsBDF**$**replicate**=**as.factor**(**sBL6vsBDF**$**Replicate.experiment**)**

sBL6vsBDF**$**block**=**as.factor**(**sBL6vsBDF**$**Block**)**

sBL6vsBDF**$**Strain**=**as.factor**(**sBL6vsBDF**$**Strain**)**

# define outcome measure

parS**<-**sBL6vsBDF**$**Diff.weight.13.10

parH**<-**hBL6vsBDF**$**Diff.weight.13.10

####### overview full linear model for both designs #######

########### linear model for conventional design ####################

lmS**=**lmer**(**parS **~** Strain **+** **(**1**|**block**)** **+** **(**1**|**block**:**Strain**)** **+** **(**1**|**replicate**)** **+** **(**1**|**Strain**:**replicate**)**, data**=**sBL6vsBDF**)**

############## linear model for mini-experiment design ##################

lmH**=**lmer**(**parH **~** Strain **+** **(**1**|**block**)** **+** **(**1**|**block**:**Strain**)** **+** **(**1**|**replicate**)** **+** **(**1**|**Strain**:**replicate**)**,data**=**hBL6vsBDF**)**

######### transformation of outcome measures #########

sBL6vsBDF**$**HCB.drink.feed**<-**sqrt**(**sBL6vsBDF**$**HCB.drink.feed**)**

sBL6vsBDF**$**HCB.climbing**<-**sqrt**(**sBL6vsBDF**$**HCB.climbing**)**

sBL6vsBDF**$**EPM.OA.rel..Time**<-**sqrt**(**sBL6vsBDF**$**EPM.OA.rel..Time**)**

sBL6vsBDF**$**Barrier.Latency..s.**<-**1**/(**sBL6vsBDF**$**Barrier.Latency..s.**+**1**)**

sBL6vsBDF**$**FCM**<-**log10**(**sBL6vsBDF**$**FCM**+**1**)**

hBL6vsBDF**$**HCB.drink.feed**<-**sqrt**(**hBL6vsBDF**$**HCB.drink.feed**)**

hBL6vsBDF**$**HCB.climbing**<-**sqrt**(**hBL6vsBDF**$**HCB.climbing**)**

hBL6vsBDF**$**EPM.OA.rel..Time**<-**sqrt**(**hBL6vsBDF**$**EPM.OA.rel..Time**)**

hBL6vsBDF**$**Barrier.Latency..s.**<-**1**/(**hBL6vsBDF**$**Barrier.Latency..s.**+**1**)**

hBL6vsBDF**$**FCM**<-**log10**(**hBL6vsBDF**$**FCM**+**1**)**

###### loop for linear model analysis of all outcome measures ###########

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

interaction**=**matrix**(NA**,nrow **=** 16,ncol **=** 4**)** #create empty matrix to fill in the p-values, NA to detect errors when something is missing

colnames**(**interaction**)=**c**(**'Stand.int','Het.int','Stand.main','Het.main'**)**

rownames**(**interaction**)=**names**(**dH**)[-**c**(**1**:**5,22**:**23**)]**

Chi\_sq**=**matrix**(NA**,nrow **=** 16,ncol **=** 4**)** #create empty matrix to fill in the Chi\_sq-values, NA to detect errors when something is missing

colnames**(**Chi\_sq**)=**c**(**'Stand.int','Het.int','Stand.main','Het.main'**)**

rownames**(**Chi\_sq**)=**names**(**dH**)[-**c**(**1**:**5,22**:**23**)]**

**for** **(** k **in** 6**:**21**){**

# conventional design

lm0**=**lmer**(**sBL6vsBDF**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)+(**1**|**Strain**:**replicate**)**,data**=**sBL6vsBDF**)** #first model with all factors

lm1**=**lmer**(**sBL6vsBDF**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**sBL6vsBDF**)** #second model with the 'strain x replicate' interaction is missing to test their significant influence against the first model

lm2**=**lmer**(**sBL6vsBDF**[**,k**]~(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**sBL6vsBDF**)** #third model without 'strain', to test strain effect

interaction**[**k**-**5,c**(**1,3**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,8**])** #test the three models against each other and create a matrix out of it

Chi\_sq**[**k**-**5,c**(**1,3**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,6**])**

#fill interaction-matrix (column 1 and 3), vector in the end: reorganising p-values

# mini-experiment design

lm0**=**lmer**(**hBL6vsBDF**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)+(**1**|**Strain**:**replicate**)**,data**=**hBL6vsBDF**)** #first model with all factors

lm1**=**lmer**(**hBL6vsBDF**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**hBL6vsBDF**)** #second model with the'strain x replicate' interaction is missing to test their significant influence against the first model

lm2**=**lmer**(**hBL6vsBDF**[**,k**]~(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**hBL6vsBDF**)** #third model without 'strain', to test strain effect

interaction**[**k**-**5,c**(**2,4**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,8**])**

Chi\_sq**[**k**-**5,c**(**2,4**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,6**])**

**}**

View**(**interaction**)** #show interaction matrix

View**(**Chi\_sq**)**

# visualisation of main strain effect and strain x replicate interaction for both designs

boxplot**(**interaction**)**

# visualisation of strain x replicate interaction for both designs

boxplot**(**interaction**[**,c**(**1,2**)])**

#pairwise comparison of p-values from the 'strain x replicate interaction term' between both experimental designs

wilcox.test**(**interaction**[**,1**]**,interaction**[**,2**]**,paired **=** **TRUE)**

## Note: gives out p-value for two sided wilcox.test, has to be dived by 2 for one-sided hypothesis! ###

############################ Analysis: strain x replicate interaction for 4. Strain effect - DBA vs BALB - #####################################

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

# define model variables

hDBAvsBALB**$**replicate**=**as.factor**(**hDBAvsBALB**$**Replicate.experiment**)**

hDBAvsBALB**$**block**=**as.factor**(**hDBAvsBALB**$**Block**)**

hDBAvsBALB**$**Strain**=**as.factor**(**hDBAvsBALB**$**Strain**)**

sDBAvsBALB**$**replicate**=**as.factor**(**sDBAvsBALB**$**Replicate.experiment**)**

sDBAvsBALB**$**block**=**as.factor**(**sDBAvsBALB**$**Block**)**

sDBAvsBALB**$**Strain**=**as.factor**(**sDBAvsBALB**$**Strain**)**

# define outcome measure

parS**<-**sDBAvsBALB**$**Diff.weight.13.10

parH**<-**hDBAvsBALB**$**Diff.weight.13.10

####### overview full linear model for both designs #######

########### linear model for conventional design ####################

lmS**=**lmer**(**parS **~** Strain **+** **(**1**|**block**)** **+** **(**1**|**block**:**Strain**)** **+** **(**1**|**replicate**)** **+** **(**1**|**Strain**:**replicate**)**, data**=**sDBAvsBALB**)**

############## linear model for mini-experiment design ##################

lmH**=**lmer**(**parH **~** Strain **+** **(**1**|**block**)** **+** **(**1**|**block**:**Strain**)** **+** **(**1**|**replicate**)** **+** **(**1**|**Strain**:**replicate**)**,data**=**hDBAvsBALB**)**

######### transformation of outcome measures #########

sDBAvsBALB**$**HCB.drink.feed**<-**1**/(**sDBAvsBALB**$**HCB.drink.feed**+**1**)**

sDBAvsBALB**$**EPM.OA.rel..Time**<-**1**/(**sDBAvsBALB**$**EPM.OA.rel..Time**+**1**)**

sDBAvsBALB**$**Barrier.Latency..s.**<-**log10**(**sDBAvsBALB**$**Barrier.Latency..s.**+**1**)**

sDBAvsBALB**$**FCM**<-**log10**(**sDBAvsBALB**$**FCM**+**1**)**

sDBAvsBALB**$**NT.24.h**<-**sqrt**(**sDBAvsBALB**$**NT.24.h**)**

hDBAvsBALB**$**HCB.drink.feed**<-**1**/(**hDBAvsBALB**$**HCB.drink.feed**+**1**)**

hDBAvsBALB**$**EPM.OA.rel..Time**<-**1**/(**hDBAvsBALB**$**EPM.OA.rel..Time**+**1**)**

hDBAvsBALB**$**Barrier.Latency..s.**<-**log10**(**hDBAvsBALB**$**Barrier.Latency..s.**+**1**)**

hDBAvsBALB**$**FCM**<-**log10**(**hDBAvsBALB**$**FCM**+**1**)**

hDBAvsBALB**$**NT.24.h**<-**sqrt**(**hDBAvsBALB**$**NT.24.h**)**

###### loop for linear model analysis of all outcome measures ###########

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

interaction**=**matrix**(NA**,nrow **=** 16,ncol **=** 4**)** #create empty matrix to fill in the p-values, NA to detect errors when something is missing

colnames**(**interaction**)=**c**(**'Stand.int','Het.int','Stand.main','Het.main'**)**

rownames**(**interaction**)=**names**(**dH**)[-**c**(**1**:**5,22**:**23**)]**

Chi\_sq**=**matrix**(NA**,nrow **=** 16,ncol **=** 4**)** #create empty matrix to fill in the Chi\_sq-values, NA to detect errors when something is missing

colnames**(**Chi\_sq**)=**c**(**'Stand.int','Het.int','Stand.main','Het.main'**)**

rownames**(**Chi\_sq**)=**names**(**dH**)[-**c**(**1**:**5,22**:**23**)]**

**for** **(** k **in** 6**:**21**){**

# conventional design

lm0**=**lmer**(**sDBAvsBALB**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)+(**1**|**Strain**:**replicate**)**,data**=**sDBAvsBALB**)** #first model with all factors

lm1**=**lmer**(**sDBAvsBALB**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**sDBAvsBALB**)** #second model with the 'strain x replicate' interaction is missing to test their significant influence against the first model

lm2**=**lmer**(**sDBAvsBALB**[**,k**]~(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**sDBAvsBALB**)** #third model without 'strain', to test strain effect

interaction**[**k**-**5,c**(**1,3**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,8**])** #test the three models against each other and create a matrix out of it

Chi\_sq**[**k**-**5,c**(**1,3**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,6**])**

#fill interaction-matrix (column 1 and 3), vector in the end: reorganising p-values

# mini-experiment design

lm0**=**lmer**(**hDBAvsBALB**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)+(**1**|**Strain**:**replicate**)**,data**=**hDBAvsBALB**)** #first model with all factors

lm1**=**lmer**(**hDBAvsBALB**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**hDBAvsBALB**)** #second model with the'strain x replicate' interaction is missing to test their significant influence against the first model

lm2**=**lmer**(**hDBAvsBALB**[**,k**]~(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**hDBAvsBALB**)** #third model without 'strain', to test strain effect

interaction**[**k**-**5,c**(**2,4**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,8**])**

Chi\_sq**[**k**-**5,c**(**2,4**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,6**])**

**}**

View**(**interaction**)** #show interaction matrix

View**(**Chi\_sq**)**

# visualisation of main strain effect and strain x replicate interaction for both designs

boxplot**(**interaction**)**

# visualisation of strain x replicate interaction for both designs

boxplot**(**interaction**[**,c**(**1,2**)])**

#pairwise comparison of p-values from the 'strain x replicate interaction term' between both experimental designs

wilcox.test**(**interaction**[**,1**]**,interaction**[**,2**]**,paired **=** **TRUE)**

## Note: gives p-value for two-sided test, has to be divided by 2 for one-sided hypothesis!##

############################ Analysis: strain x replicate interaction for 5. Strain effect - DBA vs B6D2F1 - #####################################

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

# define model variables

hDBAvsBDF**$**replicate**=**as.factor**(**hDBAvsBDF**$**Replicate.experiment**)**

hDBAvsBDF**$**block**=**as.factor**(**hDBAvsBDF**$**Block**)**

hDBAvsBDF**$**Strain**=**as.factor**(**hDBAvsBDF**$**Strain**)**

sDBAvsBDF**$**replicate**=**as.factor**(**sDBAvsBDF**$**Replicate.experiment**)**

sDBAvsBDF**$**block**=**as.factor**(**sDBAvsBDF**$**Block**)**

sDBAvsBDF**$**Strain**=**as.factor**(**sDBAvsBDF**$**Strain**)**

# define outcome measure

parS**<-**sDBAvsBDF**$**Diff.weight.13.10

parH**<-**hDBAvsBDF**$**Diff.weight.13.10

####### overview full linear model for both designs #######

########### linear model for conventional design ####################

lmS**=**lmer**(**parS **~** Strain **+** **(**1**|**block**)** **+** **(**1**|**block**:**Strain**)** **+** **(**1**|**replicate**)** **+** **(**1**|**Strain**:**replicate**)**, data**=**sDBAvsBDF**)**

############## linear model for mini-experiment design ##################

lmH**=**lmer**(**parH **~** Strain **+** **(**1**|**block**)** **+** **(**1**|**block**:**Strain**)** **+** **(**1**|**replicate**)** **+** **(**1**|**Strain**:**replicate**)**,data**=**hDBAvsBDF**)**

######### transformation of outcome measures #########

sDBAvsBDF**$**HCB.drink.feed**<-**log10**((**sDBAvsBDF**$**HCB.drink.feed**)+**1**)**

sDBAvsBDF**$**EPM.OA.rel..Time**<-**sqrt**(**sDBAvsBDF**$**EPM.OA.rel..Time**)**

sDBAvsBDF**$**OF.Center.time**<-**sqrt**(**sDBAvsBDF**$**OF.Center.time**)**

sDBAvsBDF**$**Barrier.Latency..s.**<-**log10**(**sDBAvsBDF**$**Barrier.Latency..s.**+**1**)**

sDBAvsBDF**$**PB.Latency.t3**<-**sqrt**(**sDBAvsBDF**$**PB.Latency.t3**)**

sDBAvsBDF**$**FCM**<-**log10**(**sDBAvsBDF**$**FCM**+**1**)**

hDBAvsBDF**$**HCB.drink.feed**<-**log10**((**hDBAvsBDF**$**HCB.drink.feed**)+**1**)**

hDBAvsBDF**$**EPM.OA.rel..Time**<-**sqrt**(**hDBAvsBDF**$**EPM.OA.rel..Time**)**

hDBAvsBDF**$**OF.Center.time**<-**sqrt**(**hDBAvsBDF**$**OF.Center.time**)**

hDBAvsBDF**$**Barrier.Latency..s.**<-**log10**(**hDBAvsBDF**$**Barrier.Latency..s.**+**1**)**

hDBAvsBDF**$**PB.Latency.t3**<-**sqrt**(**hDBAvsBDF**$**PB.Latency.t3**)**

hDBAvsBDF**$**FCM**<-**log10**(**hDBAvsBDF**$**FCM**+**1**)**

###### loop for linear model analysis of all outcome measures ###########

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

interaction**=**matrix**(NA**,nrow **=** 16,ncol **=** 4**)** #create empty matrix to fill in the p-values, NA to detect errors when something is missing

colnames**(**interaction**)=**c**(**'Stand.int','Het.int','Stand.main','Het.main'**)**

rownames**(**interaction**)=**names**(**dH**)[-**c**(**1**:**5,22**:**23**)]**

Chi\_sq**=**matrix**(NA**,nrow **=** 16,ncol **=** 4**)** #create empty matrix to fill in the Chi\_sq-values, NA to detect errors when something is missing

colnames**(**Chi\_sq**)=**c**(**'Stand.int','Het.int','Stand.main','Het.main'**)**

rownames**(**Chi\_sq**)=**names**(**dH**)[-**c**(**1**:**5,22**:**23**)]**

**for** **(** k **in** 6**:**21**){**

# conventional design

lm0**=**lmer**(**sDBAvsBDF**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)+(**1**|**Strain**:**replicate**)**,data**=**sDBAvsBDF**)** #first model with all factors

lm1**=**lmer**(**sDBAvsBDF**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**sDBAvsBDF**)** #second model with the 'strain x replicate' interaction is missing to test their significant influence against the first model

lm2**=**lmer**(**sDBAvsBDF**[**,k**]~(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**sDBAvsBDF**)** #third model without 'strain', to test strain effect

interaction**[**k**-**5,c**(**1,3**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,8**])** #test the three models against each other and create a matrix out of it

Chi\_sq**[**k**-**5,c**(**1,3**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,6**])**

#fill interaction-matrix (column 1 and 3), vector in the end: reorganising p-values

# mini-experiment design

lm0**=**lmer**(**hDBAvsBDF**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)+(**1**|**Strain**:**replicate**)**,data**=**hDBAvsBDF**)** #first model with all factors

lm1**=**lmer**(**hDBAvsBDF**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**hDBAvsBDF**)** #second model with the'strain x replicate' interaction is missing to test their significant influence against the first model

lm2**=**lmer**(**hDBAvsBDF**[**,k**]~(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**hDBAvsBDF**)** #third model without 'strain', to test strain effect

interaction**[**k**-**5,c**(**2,4**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,8**])**

Chi\_sq**[**k**-**5,c**(**2,4**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,6**])**

**}**

View**(**interaction**)** #show interaction matrix

View**(**Chi\_sq**)**

# visualisation of main strain effect and strain x replicate interaction for both designs

boxplot**(**interaction**)**

# visualisation of strain x replicate interaction for both designs

boxplot**(**interaction**[**,c**(**1,2**)])**

#pairwise comparison of p-values from the 'strain x replicate interaction term' between both experimental designs

wilcox.test**(**interaction**[**,1**]**,interaction**[**,2**]**,paired **=** **TRUE)**

## Note: gives p-value for two-sided test, has to be divided by 2 for one-sided hypothesis! ##

############################ Analysis: strain x replicate interaction for 6. Strain effect - BALB vs B6D2F1 - #####################################

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

# define model variables

hBALBvsBDF**$**replicate**=**as.factor**(**hBALBvsBDF**$**Replicate.experiment**)**

hBALBvsBDF**$**block**=**as.factor**(**hBALBvsBDF**$**Block**)**

hBALBvsBDF**$**Strain**=**as.factor**(**hBALBvsBDF**$**Strain**)**

sBALBvsBDF**$**replicate**=**as.factor**(**sBALBvsBDF**$**Replicate.experiment**)**

sBALBvsBDF**$**block**=**as.factor**(**sBALBvsBDF**$**Block**)**

sBALBvsBDF**$**Strain**=**as.factor**(**sBALBvsBDF**$**Strain**)**

# define outcome measure

parS**<-**sBALBvsBDF**$**Diff.weight.13.10

parH**<-**hBALBvsBDF**$**Diff.weight.13.10

####### overview full linear model for both designs #######

########### linear model for conventional design ####################

lmS**=**lmer**(**parS **~** Strain **+** **(**1**|**block**)** **+** **(**1**|**block**:**Strain**)** **+** **(**1**|**replicate**)** **+** **(**1**|**Strain**:**replicate**)**, data**=**sBALBvsBDF**)**

############## linear model for mini-experiment design ##################

lmH**=**lmer**(**parH **~** Strain **+** **(**1**|**block**)** **+** **(**1**|**block**:**Strain**)** **+** **(**1**|**replicate**)** **+** **(**1**|**Strain**:**replicate**)**,data**=**hBALBvsBDF**)**

######### transformation of outcome measures #########

sBALBvsBDF**$**HCB.climbing**<-**1**/(**sBALBvsBDF**$**HCB.climbing**+**1**)**

sBALBvsBDF**$**EPM.OA.rel..Time**<-**sqrt**(**sBALBvsBDF**$**EPM.OA.rel..Time**)**

sBALBvsBDF**$**Barrier.Latency..s.**<-**1**/(**sBALBvsBDF**$**Barrier.Latency..s.**+**1**)**

sBALBvsBDF**$**FCM**<-**log10**(**sBALBvsBDF**$**FCM**+**1**)**

sBALBvsBDF**$**FTM**<-**sqrt**(**sBALBvsBDF**$**FTM**)**

hBALBvsBDF**$**HCB.climbing**<-**1**/(**hBALBvsBDF**$**HCB.climbing**+**1**)**

hBALBvsBDF**$**EPM.OA.rel..Time**<-**sqrt**(**hBALBvsBDF**$**EPM.OA.rel..Time**)**

hBALBvsBDF**$**Barrier.Latency..s.**<-**1**/(**hBALBvsBDF**$**Barrier.Latency..s.**+**1**)**

hBALBvsBDF**$**FCM**<-**log10**(**hBALBvsBDF**$**FCM**+**1**)**

hBALBvsBDF**$**FCM**<-**sqrt**(**hBALBvsBDF**$**FTM**)**

###### loop for linear model analysis of all outcome measures ###########

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

interaction**=**matrix**(NA**,nrow **=** 16,ncol **=** 4**)** #create empty matrix to fill in the p-values, NA to detect errors when something is missing

colnames**(**interaction**)=**c**(**'Stand.int','Het.int','Stand.main','Het.main'**)**

rownames**(**interaction**)=**names**(**dH**)[-**c**(**1**:**5,22**:**23**)]**

Chi\_sq**=**matrix**(NA**,nrow **=** 16,ncol **=** 4**)** #create empty matrix to fill in the Chi\_sq-values, NA to detect errors when something is missing

colnames**(**Chi\_sq**)=**c**(**'Stand.int','Het.int','Stand.main','Het.main'**)**

rownames**(**Chi\_sq**)=**names**(**dH**)[-**c**(**1**:**5,22**:**23**)]**

**for** **(** k **in** 6**:**21**){**

# conventional design

lm0**=**lmer**(**sBALBvsBDF**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)+(**1**|**Strain**:**replicate**)**,data**=**sBALBvsBDF**)** #first model with all factors

lm1**=**lmer**(**sBALBvsBDF**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**sBALBvsBDF**)** #second model with the 'strain x replicate' interaction is missing to test their significant influence against the first model

lm2**=**lmer**(**sBALBvsBDF**[**,k**]~(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**sBALBvsBDF**)** #third model without 'strain', to test strain effect

interaction**[**k**-**5,c**(**1,3**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,8**])** #test the three models against each other and create a matrix out of it

Chi\_sq**[**k**-**5,c**(**1,3**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,6**])**

#fill interaction-matrix (column 1 and 3), vector in the end: reorganising p-values

# mini-experiment design

lm0**=**lmer**(**hBALBvsBDF**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)+(**1**|**Strain**:**replicate**)**,data**=**hBALBvsBDF**)** #first model with all factors

lm1**=**lmer**(**hBALBvsBDF**[**,k**]~**Strain**+(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**hBALBvsBDF**)** #second model with the'strain x replicate' interaction is missing to test their significant influence against the first model

lm2**=**lmer**(**hBALBvsBDF**[**,k**]~(**1**|**block**)+(**1**|**block**:**Strain**)+(**1**|**replicate**)**,data**=**hBALBvsBDF**)** #third model without 'strain', to test strain effect

interaction**[**k**-**5,c**(**2,4**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,8**])**

Chi\_sq**[**k**-**5,c**(**2,4**)]=(**as.matrix.data.frame**(**anova**(**lm2,lm1,lm0**))[**c**(**3,2**)**,6**])**

**}**

View**(**interaction**)** #show interaction matrix

View**(**Chi\_sq**)**

# visualisation of main strain effect and strain x replicate interaction for both designs

boxplot**(**interaction**)**

# visualisation of strain x replicate interaction for both designs

boxplot**(**interaction**[**,c**(**1,2**)])**

#pairwise comparison of p-values from the 'strain x replicate interaction term' between both experimental designs

wilcox.test**(**interaction**[**,1**]**,interaction**[**,2**]**,paired **=** **TRUE)**

## Note: gives p-value for two-sided test, has to be divided by 2 for one-sided hypothesis! ##