Distributional data analysis via quantile functions and its application to modelling digital biomarkers of gait in Alzheimer's Disease

Rahul Ghosal

04/21/2021

Introduction

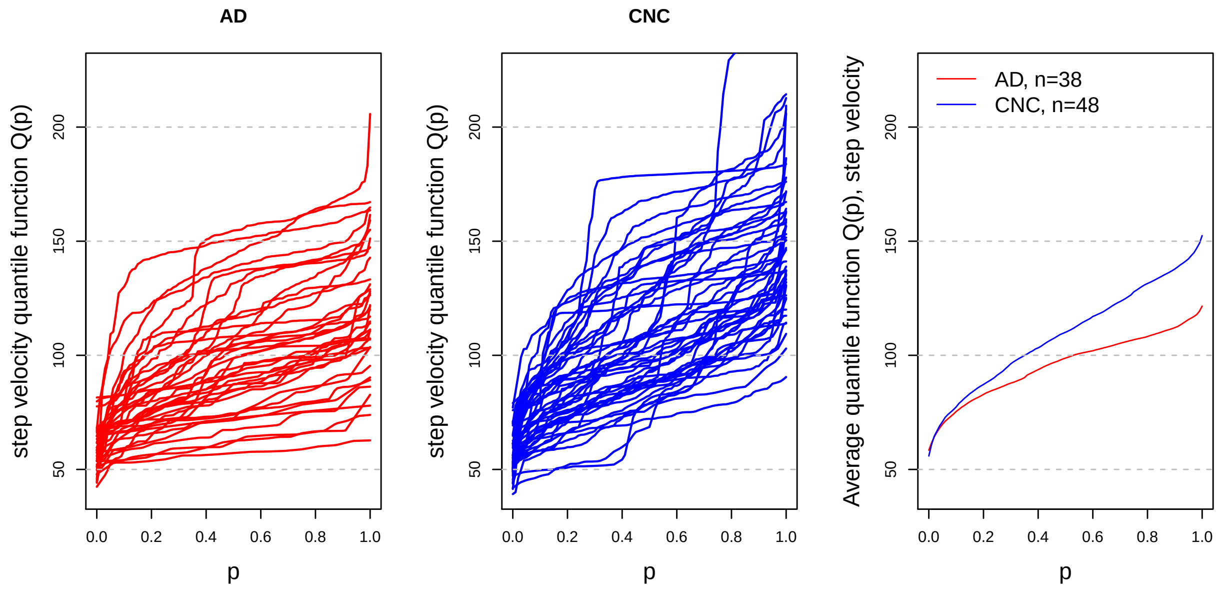
This document presents an illustration of the SOQFR and SOQFR-L method proposed in Ghosal et al. (2021) using the gait features collected on AD and CNC participants from KU-ADC. In particular, we illustrate the methods using the gait feature step velocity and the response cognitive status (mild AD or CNC). First, we plot the individual (left two panel) and average (right panel) quantile functions of step velocity for AD and CNC (Figure 1 in the paper).

```
#####Load the data############
load("spearman correlations-20180913.RData")
#preprocessing
gait<-gait[,-c(14,15,71,72,73,74)]
gait$sex<-as.numeric(gait$sex)-1
gait$adstatus<-as.numeric(gait$adstatus)-1
#names(gait)
#select id,age,sex,adstatus and gait feature step velocity
gait<-gait[,c(1,2,3,9,61)]
head(gait)
```

```
##   id age sex adstatus Step_Velocity__cm_sec_
## 1  2  67   1        1               119.3454
## 2  2  67   1        1               137.5917
## 3  2  67   1        1               138.2585
## 4  2  67   1        1               139.4403
## 5  2  67   1        1               139.4684
## 6  2  67   1        1               140.0375
```

```
agg1 = aggregate(gait,
                 by = list(gait$id),
                 FUN = mean,na.rm=TRUE,na.action=na.omit)
###Calculating subject-specific quantile function by id#####
agg2 = aggregate(gait,
                 by = list(gait$id),
                 FUN = quantile,probs=c(seq(0,1,l=101)),na.rm=TRUE)
p<-seq(0,1,l=101)
#######getting AD and CNC groups###############################
```

```
ady<-which(agg1$adstatus==1)
svely<-agg2$Step_Velocity__cm_sec_[ady,]
sveln<-agg2$Step_Velocity__cm_sec_[-ady,]
#######IMPUTING NA values with mean###########################
for(i in 1:ncol(svely)){
  svely[is.na(svely[,i]), i] <- mean(svely[,i], na.rm = TRUE)
}
for(i in 1:ncol(sveln)){
  sveln[is.na(sveln[,i]), i] <- mean(sveln[,i], na.rm = TRUE)
}
#############Plotting subject-specific and average QF##########
par(mfrow=c(1,3))
par(mar =  c(5.1, 4.5, 4.1, 2.1))
matplot(p, t(svely), type="l", lty=rep(1,38),
        ylab="step velocity quantile function Q(p)", xlab="p",
        cex.lab=1.5, lwd=1.5,
        col="red",main="AD",ylim=c(40,225))
abline(h=c(50,100,150,200),col="grey",lty=2)
matplot(p, t(sveln), type="l", lty=rep(1,48),
        ylab="step velocity quantile function Q(p)", xlab="p",
        cex.lab=1.5, lwd=1.5,
        col="blue",main="CNC",ylim=c(40,225))
abline(h=c(50,100,150,200),col="grey",lty=2)
plot(p,colMeans(svely),ylab="Average quantile function Q(p), step velocity",xlab="p",col="red",ylim=c(40
lines(p,colMeans(sveln),ylab=" Average quantile function Q(p), step velocity",xlab="p",col="blue")
abline(h=c(50,100,150,200),col="grey",lty=2)
legend('topleft',c("AD, n=38","CNC, n=48") ,
       lty=c(1,1), col=c("red", "blue"), bty='n', cex=1.4)
```



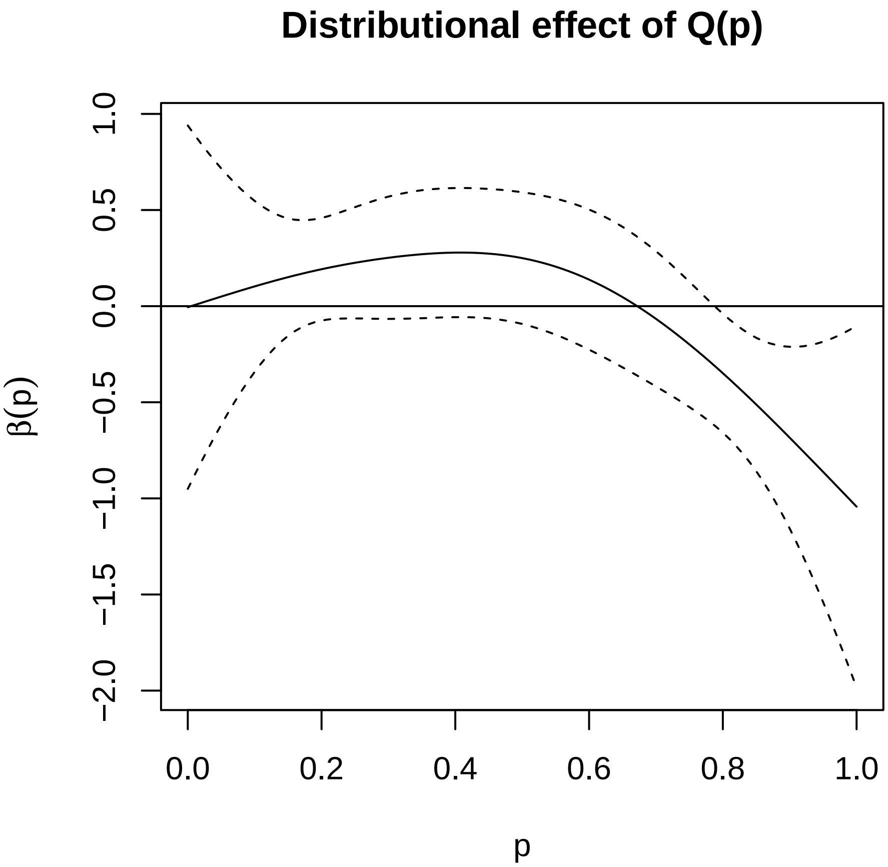
SOQFR for discrimination of AD

We illustrate the SOQFR method with logit-link to model mild-AD vs CNC using subject-specific quantile functions of step velocity and and adjusting for age, sex. We display the estimated coefficient function $\beta(p)$.

```
####fitting SOQFR#############
agg<-cbind(agg1[,c(3,4,5)])
svel<-agg2[,6]
#####Replace NA##############
for(i in 1:ncol(svel)){
svel[is.na(svel[,i]), i] <- mean(svel[,i], na.rm = TRUE)
}
agg$step_vel<-svel
library(refund)
fit.lf <- pfr(adstatus ~ age+sex+lf(step_vel,argvals = p, k=20, bs="ps",m=2), data=agg,family="binomial
deviance<-1-(fit.lf$deviance/fit.lf$null.deviance)
summary(fit.lf)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## adstatus ~ age + sex + s(x = step_vel.tmat, by = L.step_vel,
##     k = 20, bs = "ps", m = 2)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 16.87647    5.65643   2.984  0.00285 **
## age         -0.14831    0.05599  -2.649  0.00808 **
## sex          3.70658    0.90441   4.098 4.16e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Approximate significance of smooth terms:
##                                edf Ref.df Chi.sq p-value
## s(step_vel.tmat):L.step_vel 3.041  3.387  16.37 0.00184 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.508   Deviance explained = 49.8%
## -REML = 34.587  Scale est. = 1         n = 86
```

```
bhat.lf <- coef(fit.lf, n=101)
bhat.lf$upper <- bhat.lf$value + 1.96*bhat.lf$se
bhat.lf$lower <- bhat.lf$value - 1.96*bhat.lf$se
matplot(p, bhat.lf[,c("value", "upper", "lower")],
          type="l", lty=c(1,2,2), col=1,
          ylab=expression(paste(beta(p))), xlab="p",main="Distributional effect of Q(p)")
abline(h=0)
```



SOQFR-L for discrimination of AD

Next, We illustrate the SOQFR-L method with logit-link to model mild-AD vs CNC using subject-specific L-moments of step velocity and and adjusting for age, sex. Finally We display the estimated coefficient function $\beta(p)$ from SOQFR-L.

```
####Calculate first 4 L-moments of step velocity############
library(lmom)
agg3 = aggregate(gait,by = list(gait$id), FUN =samlmu,nmom=4,ratios=FALSE)
svelLmom<-agg3[,6]
##Replace NA
for(i in 1:ncol(svelLmom)){
svelLmom[is.na(svelLmom[,i]), i] <- mean(svelLmom[,i], na.rm = TRUE)
}
####Perform GLM with first 4 L-moments adjusting for age,sex####
aggsvel<-cbind(agg1[,c(3,4,5)])
aggsvel<-cbind(aggsvel,svelLmom)
names(aggsvel)[4:7]<-paste("step_vel L_",1:4)
head(aggsvel)
```

```
##   age sex adstatus step_vel L_ 1 step_vel L_ 2 step_vel L_ 3 step_vel L_ 4
## 1  67   1        1     118.66671     14.283733   -4.20857914    -0.2473097
## 2  77   1        1     108.29013      6.390106   -2.69455205     2.1709742
## 3  79   0        1      62.70655      4.315913    0.85176849     0.9894494
## 4  83   1        1      71.76222      4.380337   -0.72913200     0.7133580
## 5  68   1        1      96.29518     10.659547   -0.83558694     0.5474216
## 6  76   0        1      76.20300      5.993737    0.04845354     0.8160179
```

```
SOQFR_L<-glm(adstatus~.,data=aggsvel,family = "binomial")
summary(SOQFR_L)
```

```
##
## Call:
## glm(formula = adstatus ~ ., family = "binomial", data = aggsvel)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2867  -0.4442  -0.0585   0.4475   1.8164
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     18.20321    6.06171   3.003  0.00267 **
## age             -0.15152    0.05746  -2.637  0.00836 **
## sex              3.70622    0.91500   4.051 5.11e-05 ***
## ‘step_vel L_ 1‘ -0.04440    0.03193  -1.390  0.16441
## ‘step_vel L_ 2‘ -0.48015    0.14901  -3.222  0.00127 **
## ‘step_vel L_ 3‘ -0.60122    0.28703  -2.095  0.03620 *
## ‘step_vel L_ 4‘ -0.21182    0.41532  -0.510  0.61004
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 118.056  on 85  degrees of freedom
## Residual deviance:  60.495  on 79  degrees of freedom
## AIC: 74.495
##
## Number of Fisher Scoring iterations: 6
```

```
###calculating deviance explained###################
devianceL<-1-(SOQFR_L$deviance/SOQFR_L$null.deviance)
devianceL
```

```
## [1] 0.4875766
```

We display the coefficient function $\beta(p)$ from SOQFR-L below, along with the estimated $\beta(p)$ from SOQFR.

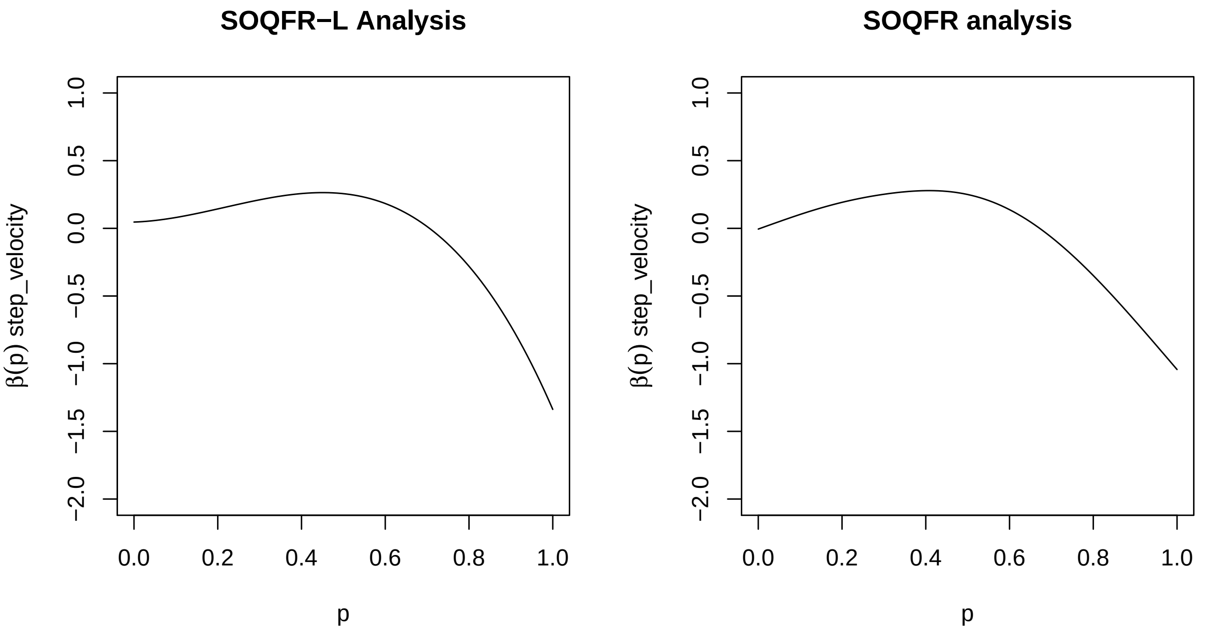
```
####Calculate the beta(p) from SOQFR-L,number of L-momets=4#####
beta<-as.numeric(coef(SOQFR_L)[-c(1:3)])
p0<-function(x){1}
p1<-function(x){-1 + 2*x }
p2<-function(x){
  1 - 6*x + 6*x^2}
```

```
p3<-function(x){
  -1 + 12*x - 30*x^2 + 20*x^3}

betax<-function(x)
{beta[1]*p0(x)+beta[2]*p1(x)+beta[3]*p2(x)+beta[4]*p3(x) }

x<-seq(0,1,l=100)
par(mfrow=c(1,2))
plot(x,betax(x),type="l",ylim=c(-2,1),xlab="p",ylab=expression(paste(beta(p)," ","step_velocity")),main=

matplot(p, bhat.lf[,c("value")],
        type="l", lty=c(1), col=1,
        ylab=expression(paste(beta(p)," ","step_velocity")), xlab="p",main="SOQFR analysis",ylim=c(-2,1
```



Higher maximal performance for step velocity is found to be associated with lower odds of AD.

SOQFR-L for modelling cognitive score of VM

Next, We illustrate an eample of the SOQFR-L method with identity-link to model VM score (one of the cognitive scores) using subject-specific L-moments of step velocity and and adjusting for age, sex and education. The second order L-moment is found to be associated with VM.

```
###loading cognitive scores data##############################
load("cogscores.RData")

####Perform SOQFR with first 4 L-moments of step velocity#####
dfsvel<-cbind(cogdata,svelLmom)  #61-14+1
names(dfsvel)[8:11]<-paste("step_vel L_",1:4)
out<-lm(VM~.,data=dfsvel[,-c(1,5,7)])
summary(out)
```

```
##
## Call:
## lm(formula = VM ~ ., data = dfsvel[, -c(1, 5, 7)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3497 -0.8845  0.0171  1.0027  3.0869
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.283e+00  2.311e+00  -1.420   0.1595
## sex            -1.591e+00  3.269e-01  -4.865 5.83e-06 ***
## age             9.013e-03  2.393e-02   0.377   0.7074
## Edu             1.191e-01  5.199e-02   2.291   0.0246 *
## `step_vel L_ 1`  7.410e-06  1.500e-02   0.000   0.9996
## `step_vel L_ 2`  1.313e-01  5.210e-02   2.520   0.0138 *
## `step_vel L_ 3`  9.408e-02  1.001e-01   0.940   0.3502
## `step_vel L_ 4`  1.188e-01  1.889e-01   0.629   0.5313
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.452 on 78 degrees of freedom
## Multiple R-squared:  0.4091, Adjusted R-squared:  0.356
## F-statistic: 7.713 on 7 and 78 DF,  p-value: 4.602e-07
```

JIVE with L-moments.

We illustrate the JIVE approach with L-moments developed in Section 3.2 of the paper. We focus on the domainsof Pace (3 features), Rhythm (13 features) and Variability (19 features).

```
rm(list = ls())
#################Load the gait data#####################
load("spearman correlations-20180913.RData")
#preprocessing
gait<-gait[,-c(14,15,71,72,73,74)]
gait$sex<-as.numeric(gait$sex)-1
gait$adstatus<-as.numeric(gait$adstatus)-1
#names(gait)
gait<-gait[,14:68]  ##all the gait features
namegait<-names(gait)

#REMOVE frqap frqml frqv #####these are repeated features###
gait<-gait[,-c(9,10,11)]
namegaitnew<-names(gait)

###load previously extracted pre-normalized and standardized L-moments data for all features ##########
load("prenormlmomallfeat.RData")
domain<-c("Amplitude","Pace","Rhythm","Symmetry","Variability")
facdom<-as.numeric(as.factor(domain))
group<-c(1,3,1,1,1,1,1,1,3,3,3,5,5,5,5,5,5,5,5,5,4,5,4,3,3,4,5,4,3,3,4,5,4,3,3,4,4,4,3,3,3,
         3,5,3,5,2,5,2,2,5,3,5,1,5,5)
lmomlist<-lmom1subj[-c(9,10,11)]
group<-group[-c(9,10,11)]
```

```r
datablock<-list()
for(j in 1:5)
{indj<-which(group==j)
tempdata<-lmomlist[indj]
data<-Reduce(cbind,tempdata)
datablock[[j]]<-t(data)
}
#########We focus on 3 domains of Pace, Rhythm and Variability" ########
datablock<-datablock[c(2,3,5)]
names(datablock)<-domain[c(2,3,5)]
######################################
library(r.jive)
Results = jive(datablock)
```

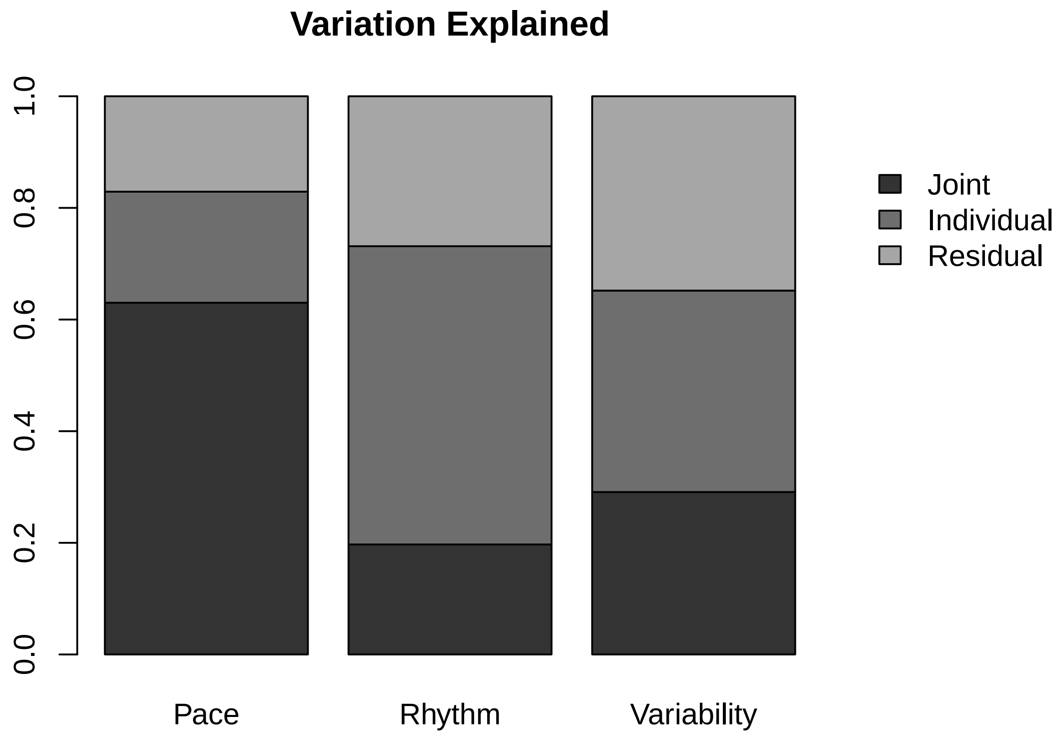
```
## Estimating  joint and individual ranks via permutation...
## Running JIVE algorithm for ranks:
## joint rank: 2 , individual ranks: 3 6 6
## JIVE algorithm converged after  42  iterations.
## Re-estimating  joint and individual ranks via permutation...
## Running JIVE algorithm for ranks:
## joint rank: 2 , individual ranks: 2 7 9
## JIVE algorithm converged after  39  iterations.
## Re-estimating  joint and individual ranks via permutation...
## Final joint rank: 2 , final individual ranks: 2 7 9
```

```r
summary(Results)
```

```
## $Method
## [1] "perm"
##
## $Ranks
##      Source        Rank
## [1,] "Joint"       "2"
## [2,] "Pace"        "2"
## [3,] "Rhythm"      "7"
## [4,] "Variability" "9"
##
## $Variance
##             Pace Rhythm Variability
## Joint      0.630  0.197       0.291
## Individual 0.199  0.534       0.361
## Residual   0.171  0.269       0.348
```

The amount of variation explained by joint and individual components in each of the three domains are displayed below.

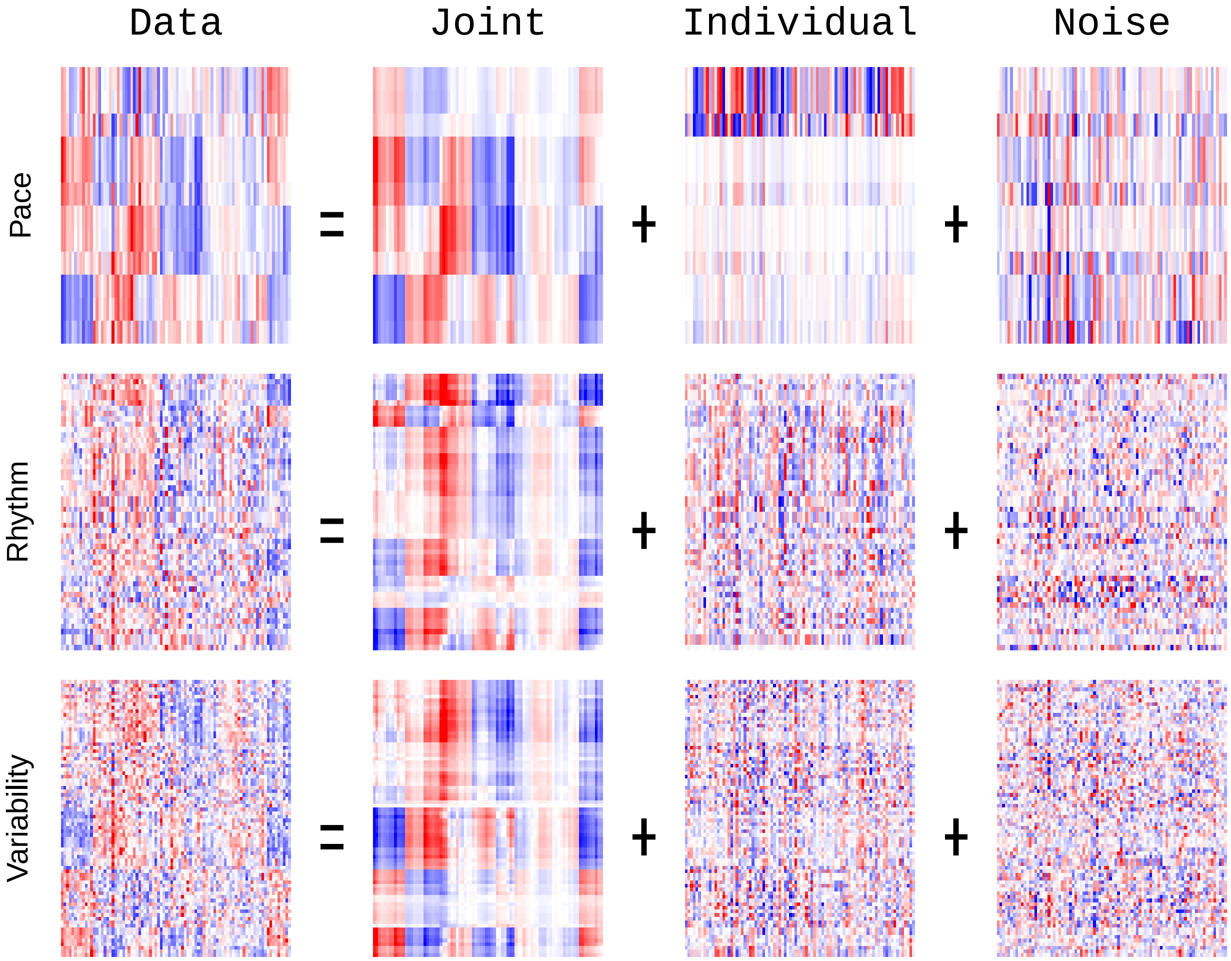
```r
showVarExplained(Results)
```



JIVE estimates of the joint and individual structures are displayed below.

```
showHeatmaps(Results)
```

29	2	17	6	21	1	25	13
30	3	18	7	22	10	26	14
31	4	19	8	23	11	27	15
32	5	20	9	24	12	28	16



The joint structure explains quite a large variation in the pace domain, where as the individual structures explain the majority of the variation in the Rhythm and Variability domain.