

data explore

Wanying Ma

12/18/2018

Access to data files

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

Add group index, merge with player information, keep only football players, delete duplicates between 3 groups, and construct group index.

```
#####
##### players information table #####
##### create group index #####
##### delete duplicates #####
#####

player.info <- read.csv("/Users/mawanying/Dropbox/Luo_Jarek_YC/MRI_Injury_Description_03OCT18_All_Concu
basic.info <- player.info %>% select(SUBJECTSTUDYNUM, INJURYSPORTVARSITY, CONTACTSPORTQGID, NONCONTACTS
football.info <- basic.info %>% filter(INJURYSPORTVARSITY=='Football') #football.info is 88x4
#first in second
d1 <- football.info$SUBJECTSTUDYNUM %in% football.info$CONTACTSPORTQGID
#first in third
d2 <- football.info$SUBJECTSTUDYNUM %in% football.info$NONCONTACTSPORTQGID
#second in first
d3 <- football.info$CONTACTSPORTQGID %in% football.info$SUBJECTSTUDYNUM
#second in third
d4 <- football.info$CONTACTSPORTQGID %in% football.info$NONCONTACTSPORTQGID
#third in first
d5 <- football.info$NONCONTACTSPORTQGID %in% football.info$SUBJECTSTUDYNUM
#third in second
d6 <- football.info$NONCONTACTSPORTQGID %in% football.info$CONTACTSPORTQGID
dupli <- cbind(d1, d2, d3, d4, d5, d6) #88x6
football.noDup <- football.info[rowSums(dupli)<1, ] # football.noDup is 53x4

#####
##### construct group index for football players #####
#####
##### group1: SUBJECTSTUDYNUM #####
##### group2: CONTACTSPORTQGID #####
##### group3: NONCONTACTSPORTQGID #####
#####
```

```

group1 <- unique(as.character(football.noDup$SUBJECTSTUDYNUM))      #SUBJECTSTUDYNUM 43x1
group2 <- unique(as.character(football.noDup$CONTACTSPORTQGID))    #CONTACTSPORTQGID 43x1
group3 <- unique(as.character(football.noDup$NONCONTACTSPORTQGID)) #NONCONTACTSPORTQGID 43x1

# after merge, alldata_fb will have 28 players in group 1, 30players in group 2, 28 players in group 3
#####
alldata.player <- as.character(alldata$player) #21052x1
alldata.fb.player <- alldata.player %in% c(group1, group2, group3) #indicator vector: 9280x1
groupindex <- c()
for(i in alldata.player[alldata.fb.player]){
  if(i %in% group1){
    groupindex <- c(groupindex, 1)
  }
  if(i %in% group2){
    groupindex <- c(groupindex, 2)
  }
  if(i %in% group3){
    groupindex <- c(groupindex, 3)
  }
}

addgroupindex <- function(data, football_index, groupindex){
  fbdata <- data[football_index, ]
  data.fb <- fbdata %>% mutate(group=groupindex)
  return(data.fb)
}

alldata.fb <- addgroupindex(alldata, alldata.fb.player, groupindex) #9280x204
alldata.fb_qdt <- addgroupindex(alldata_qdt, alldata.fb.player, groupindex) #9280x204

colnames(alldata.fb)[4:203] <- colnames(alldata.fb_qdt)[4:203] <- paste0("den", 1:200)
#table(alldata.fb$group)
#1      2      3
#3016 3240 3024
# there are 86 different players in total for alldata.fb
#alldata_fb will have 28 players in group 1, 30players in group 2, 28 players in group 3

```

cluster

Using equally-spaced density(no transformation), as well as quantile-density transformation of density $-\log(f(Q(t)))$

```

m.tract.cluster <- function(data, iter){
  # data: input dataset
  # m: measurement, in {"FA", "MD", "Da", "Dr"}
  # t: tract
  t <- as.character(para[iter, 1])
  m <- as.character(para[iter, 2])
  data.tm <- data %>% filter(tract==t, measure==m) #86x204
  den.tm <- select(data.tm, den1:den200) #86x200
  den.tm <- den.tm - as.vector(rep(1, dim(den.tm)[1])) %*% t(colMeans(den.tm))
  den.fpca <- refund::fpca.face(data.matrix(den.tm), center = TRUE, argvals = (1:200)/200, knots = 7, p
  den.score <- den.fpca$scores/sqrt(dim(den.tm)[2]) #86x5

```

```

den.cluster <- kmeans(den.score, centers = 3)

permute <- function(vec, p1, p2, p3){
  for(i in 1:length(vec)){
    if(vec[i]==1){
      vec[i] <- p1
    }else{
      if(vec[i]==2){vec[i] <- p2}else{
        vec[i] <- p3
      }
    }
  }
}

return(vec)
}

perm1 <- permute(den.cluster$cluster, 1,2,3)
perm2 <- permute(den.cluster$cluster, 1,3,2)
perm3 <- permute(den.cluster$cluster, 2,1,3)
perm4 <- permute(den.cluster$cluster, 2,3,1)
perm5 <- permute(den.cluster$cluster, 3,1,2)
perm6 <- permute(den.cluster$cluster, 3,2,1)
misrate <- min(sum(perm1!=data.tm$group)/length(data.tm$group),
  sum(perm2!=data.tm$group)/length(data.tm$group),
  sum(perm3!=data.tm$group)/length(data.tm$group),
  sum(perm4!=data.tm$group)/length(data.tm$group),
  sum(perm5!=data.tm$group)/length(data.tm$group),
  sum(perm6!=data.tm$group)/length(data.tm$group))
data.tm.comb = cbind(data.tm, den.score)
return(list(misrate=misrate,
  data.tm.comb=data.tm.comb,
  den.fPCA=den.fPCA))
}

m.tract.cluster2 <- function(data, iter){
  # data: input dataset
  # m: measurement, in {"FA", "MD", "Da", "Dr"}
  # t: tract
  t <- as.character(para[iter, 1])
  m <- as.character(para[iter, 2])

  data.tm <- data %>% filter(tract==t, measure==m, group != 3) #86x204
  den.tm <- select(data.tm, den1:den200) #86x200
  #den.tm <- den.tm - as.vector(rep(1, dim(den.tm)[1])) %*% t(colMeans(den.tm))
  den.fPCA <- refund::fPCA.face(data.matrix(den.tm), center = TRUE, argvals = (1:200)/200, knots = 7, p
  den.score <- den.fPCA$scores/sqrt(dim(den.tm)[2]) #86x5
  den.cluster <- kmeans(den.score, centers = 2)
  permute <- function(vec, p1, p2){
    for(i in 1:length(vec)){
      if(vec[i]==1){vec[i] <- p1}else{vec[i] <- p2}
    }
  }
  return(vec)
}

```

```

perm1 <- permute(den.cluster$cluster, 1,2)
perm2 <- permute(den.cluster$cluster, 2,1)
misrate <- min(sum(perm1!=data.tm$group)/length(data.tm$group),
               sum(perm2!=data.tm$group)/length(data.tm$group))
data.tm.comb = cbind(data.tm, den.score)
return(list(misrate=misrate,
            data.tm.comb=data.tm.comb,
            den.fpca=den.fpca))
}

```

cluster results

```

## Misrate for 3 clusters:
## No transformation: FA
## [1] 0.5348837 0.5581395 0.5581395 0.5697674 0.5697674
## No transformation: MD
## [1] 0.5232558 0.5465116 0.5465116 0.5764706 0.5813953
## No transformation: Da
## [1] 0.5465116 0.5581395 0.5697674 0.5813953 0.5813953
## No transformation: Dr
## [1] 0.5697674 0.5813953 0.5813953 0.5813953 0.5813953
## Corresponding tracts:
## FA:
## [1] "cgc_r" "ifo_r" "slf_l" "ar_r" "cst_r"
## MD:
## [1] "unc_r" "mcp" "slf_l" "ml_r" "ar_l"
## Da:
## [1] "str_r" "unc_r" "ar_r" "cgc_l" "cst_l"
## Dr:
## [1] "str_r" "ar_l" "atr_l" "cgh_l" "slf_l"
## Misrate for 3 clusters:
## Quantile-density transformation: FA
## [1] 0.5116279 0.5232558 0.5465116 0.5581395 0.5697674
## Quantile-density transformation: MD
## [1] 0.5882353 0.5882353 0.5930233 0.5930233 0.5930233
## Quantile-density transformation: Da
## [1] 0.5465116 0.5813953 0.5813953 0.5813953 0.5813953
## Quantile-density transformation: Dr
## [1] 0.5348837 0.5581395 0.5581395 0.5697674 0.5697674

```

```

## Corresponding tracts:
## FA:
## [1] "ptr_l" "cgc_r" "ar_r" "slf_l" "atr_l"
## MD:
## [1] "ml_l" "ml_r" "cgh_r" "fmi" "ilf_l"
## Da:
## [1] "ar_r" "cgh_r" "fma" "ifo_l" "unc_l"
## Dr:
## [1] "ar_l" "fma" "ptr_l" "ar_r" "slf_l"

```

only cluster first two groups

```

## Misrate for 2 clusters:
## No transformation: FA
## [1] 0.3448276 0.4137931 0.4137931 0.4137931 0.4137931
## No transformation: MD
## [1] 0.3965517 0.4137931 0.4137931 0.4137931 0.4310345
## No transformation: Da
## [1] 0.3793103 0.3793103 0.3965517 0.3965517 0.4137931
## No transformation: Dr
## [1] 0.3793103 0.3793103 0.3793103 0.3965517 0.4137931
## Corresponding tracts:
## FA:
## [1] "cgh_l" "ar_r" "cgc_l" "cgc_r" "ptr_l"
## MD:
## [1] "cst_l" "cst_r" "slf_l" "unc_r" "ar_r"
## Da:
## [1] "fmi" "unc_r" "atr_l" "str_r" "cgh_r"
## Dr:
## [1] "atr_l" "ptr_r" "unc_r" "ifo_r" "ar_l"
## Misrate for 2 clusters:
## FA
## [1] 0.3620690 0.3620690 0.3793103 0.3793103 0.3965517
## qd transformation: MD
## [1] 0.3793103 0.4310345 0.4310345 0.4310345 0.4310345
## qd transformation: Da
## [1] 0.3620690 0.3620690 0.3965517 0.4310345 0.4310345

```

```
## qd transformation: Dr
## [1] 0.3793103 0.3965517 0.3965517 0.3965517 0.4137931
## Corresponding tracts:
## Quantile-density transformation: FA
## [1] "cst_l" "mcp" "cgh_l" "cst_r" "cgc_r"
## Quantile-density transformation: MD
## [1] "unc_r" "atr_r" "cgc_l" "cgc_r" "ifo_l"
## Quantile-density transformation: Da
## [1] "ptr_r" "unc_r" "str_l" "ifo_l" "ilf_r"
## Quantile-density transformation: Dr
## [1] "fma" "ar_l" "cgh_r" "cst_l" "slf_l"
```

Visualized variation modes

```
#fb_score <- lapply(1:dim(para)[1], function(x) m.tract.cluster2(alldata.fb, x)$data.tm.comb)
#fb_score <- do.call(rbind, fb_score) #9280x209
#fb_scorefpca <- lapply(1:dim(para)[1], function(x) m.tract.cluster2(alldata.fb, x)$den.fpca)

var_mode <- function(data, m, t, title, qd){
  loc_m <- which(para[,2]==m)
  loc <- loc_m[which(para[loc_m,1]==t)] #loc in the para that is tract=='unc_r', measure=='Da'
  fb_scorefpca <- lapply(loc, function(x) m.tract.cluster2(data, x)$den.fpca)

  eigenf <- fb_scorefpca[[1]]$efunctions*sqrt(200)
  eigens <- fb_scorefpca[[1]]$scores/sqrt(200)
  mean <- fb_scorefpca[[1]]$mu
  pc1_score_q <- quantile(eigens[,1], c(0.1, 0.25, 0.75, 0.95))
  modes1 <- cbind(t(as.vector(pc1_score_q)%*% t(eigenf[,1])) + mean, mean) #200x5
  pc2_score_q <- quantile(eigens[,2], c(0.1, 0.25, 0.75, 0.95))
  modes2 <- cbind(t(as.vector(pc2_score_q)%*% t(eigenf[,2])) + mean, mean) #200x5
  tran_back <- function(den.tran.y, den.quantiles){
    thetax <- sum((head(exp(den.tran.y), -1)+tail(exp(den.tran.y), -1))/2*(diff(den.quantiles/100)[1]))
    tranback_x <- sapply(1:200, function(x){
      sum((head(exp(den.tran.y)[1:x], -1)+tail(exp(den.tran.y)[1:x], -1))/2*(diff(den.quantiles/100)[1]))/
    })
    tranback_y <- exp(-den.tran.y)*thetax
    return(list(tranback_x= tranback_x,
               tranback_y=tranback_y,
               thetax=thetax))
  }

  if(qd==0){
    matplot(den.quantiles/100, modes1, main=paste0("1st PC modes, ",title, "\n tract:", t, ", measure:", m),
            xlab="t", ylab=expression(psi(t)),
            col=c("red","pink","green","blue", "black"), type="l",
            lty=c(1,1,1,1,1), lwd=c(1,1,1,1,1.5))
```

```

abline(h=0, col="grey", lwd=1, lty=2)
legend("topleft", c(expression(alpha[1]), expression(alpha[2]), expression(alpha[3]), expression(alpha[4])),
      col=c("red", "pink", "green", "blue", "black"), type="l",
      lty=c(1,1,1,1,1), lwd=c(1,1,1,1,1.5))

matplot(den.quantiles/100, modes2, main=paste0("1st PC modes, ", title, "\n tract:", t, ", measure:", m),
      xlab="t", ylab=expression(psi(t)),
      col=c("red", "pink", "green", "blue", "black"), type="l",
      lty=c(1,1,1,1,1), lwd=c(1,1,1,1,1.5))
abline(h=0, col="grey", lwd=1, lty=2)
legend("topleft", c(expression(alpha[1]), expression(alpha[2]), expression(alpha[3]), expression(alpha[4])),
      col=c("red", "pink", "green", "blue", "black"), type="l",
      lty=c(1,1,1,1,1), lwd=c(1,1,1,1,1.5))

}
if(qd==1){
  tran_den <- lapply(data.frame(modes1), function(x){tran_back(x, den.quantiles)})
  plot(tran_den[[1]]$tranback_x, tran_den[[1]]$tranback_y, type="l", main=paste0("1st PC modes, ", title, "\n tract:", t, ", measure:", m),
      xlab="x", ylab="f(x)",
      col=c("red", "pink", "green", "blue", "black"), type="l",
      lty=c(1,1,1,1,1), lwd=c(1,1,1,1,1.5))
  abline(h=0, col="grey", lwd=1, lty=2)
  lines(tran_den[[2]]$tranback_x, tran_den[[2]]$tranback_y, type="l", col="pink")
  lines(tran_den[[3]]$tranback_x, tran_den[[3]]$tranback_y, type="l", col="green")
  lines(tran_den[[4]]$tranback_x, tran_den[[4]]$tranback_y, type="l", col="blue")
  lines(tran_den[[5]]$tranback_x, tran_den[[5]]$tranback_y, col="black", lwd=1.5)
  legend("topleft", c(expression(alpha[1]), expression(alpha[2]), expression(alpha[3]), expression(alpha[4])),
      col=c("red", "pink", "green", "blue", "black"), type="l",
      lty=c(1,1,1,1,1), lwd=c(1,1,1,1,1.5))

  tran_den2 <- lapply(data.frame(modes2), function(x){tran_back(x, den.quantiles)})
  plot(tran_den2[[1]]$tranback_x, tran_den2[[1]]$tranback_y, type="l", main=paste0("2nd PC modes, ", title, "\n tract:", t, ", measure:", m),
      xlab="x", ylab=expression(psi^{-1}(x)),
      col=c("red", "pink", "green", "blue", "black"), type="l",
      lty=c(1,1,1,1,1), lwd=c(1,1,1,1,1.5))
  abline(h=0, col="grey", lwd=1, lty=2)
  lines(tran_den2[[2]]$tranback_x, tran_den2[[2]]$tranback_y, type="l", col="pink")
  lines(tran_den2[[3]]$tranback_x, tran_den2[[3]]$tranback_y, type="l", col="green")
  lines(tran_den2[[4]]$tranback_x, tran_den2[[4]]$tranback_y, type="l", col="blue")
  lines(tran_den2[[5]]$tranback_x, tran_den2[[5]]$tranback_y, col="black", lwd=1.5)
  legend("topleft", c(expression(alpha[1]), expression(alpha[2]), expression(alpha[3]), expression(alpha[4])),
      col=c("red", "pink", "green", "blue", "black"), type="l",
      lty=c(1,1,1,1,1), lwd=c(1,1,1,1,1.5))

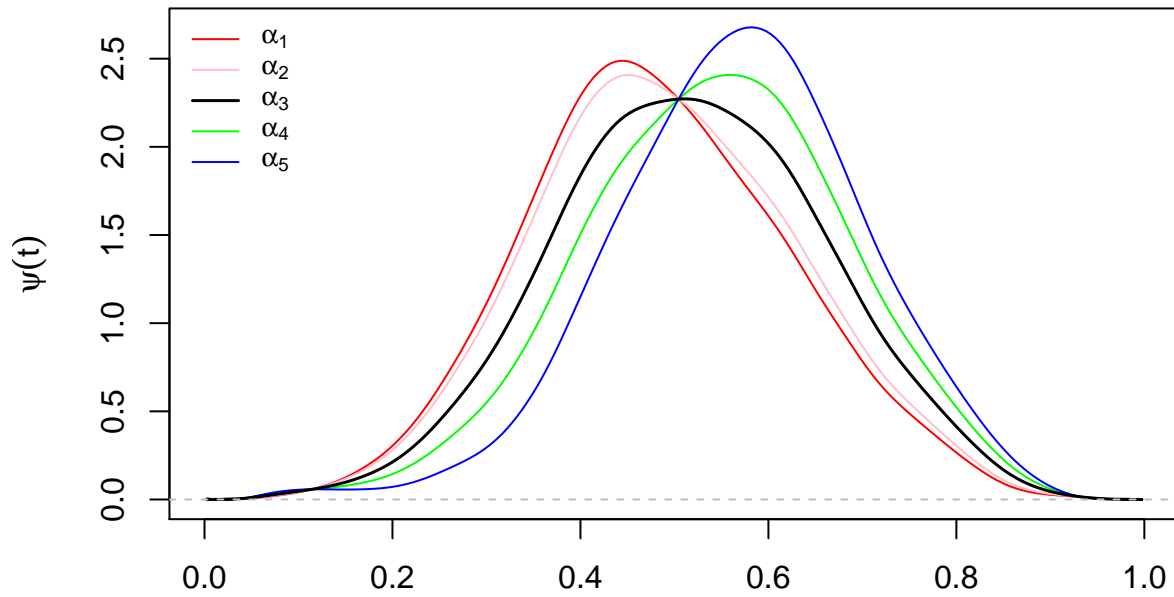
}
}

locM <- which.min(c(min(misrate_FA_qdt2), min(misrate_MD_qdt2), min(misrate_Da_qdt2), min(misrate_Dr_qdt2)))
minM <- c('FA', 'MD', 'Da', 'Dr')[locM]
minT <- c(as.character(tractnames)[order(misrate_FA_qdt2)[1]],
  as.character(tractnames)[order(misrate_MD_qdt2)[1]],
  as.character(tractnames)[order(misrate_Da_qdt2)[1]],
  as.character(tractnames)[order(misrate_Dr_qdt2)[1]])[locM]

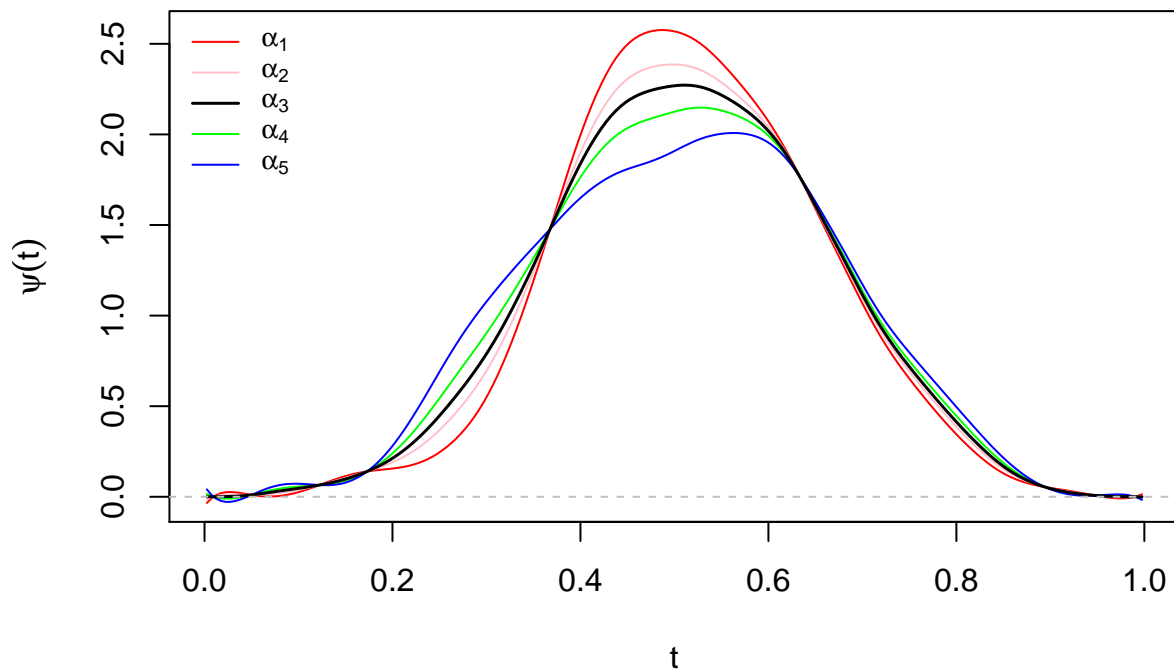
var_mode(alldata.fb, m=minM, t=minT, title='Ordinary FPCA', qd=0)

```

1st PC modes, Ordinary FPCA
tract:cst_I, measure:FA

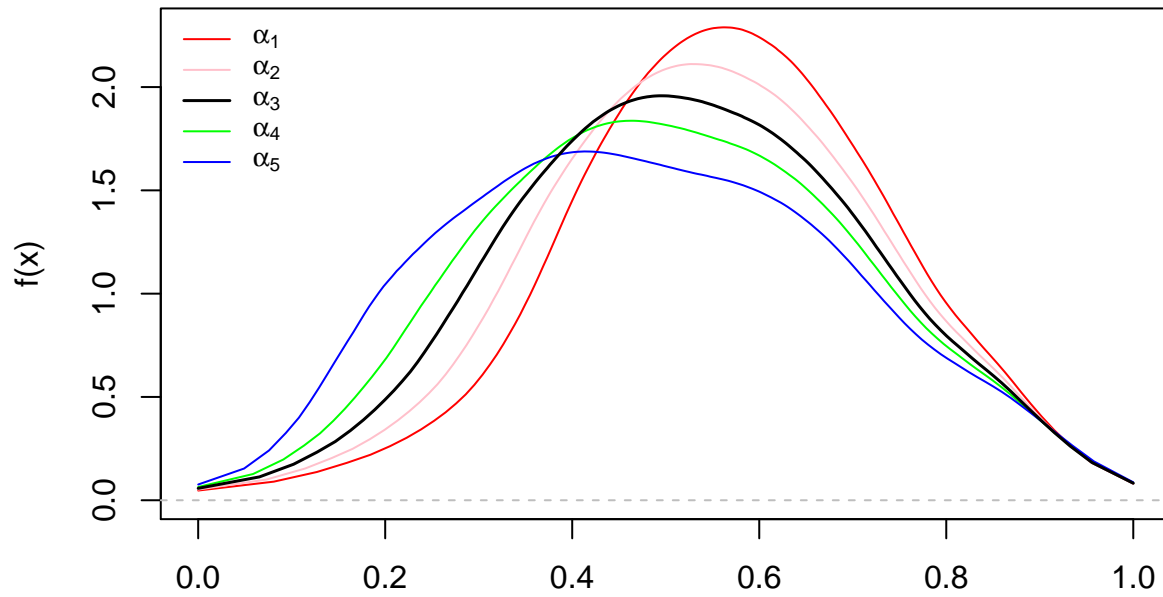


1st PC modes, Ordinary FPCA
tract:cst_I, measure:FA



```
var_mode(alldata.fb_qdt, m=minM, t=minT, title='log quantile-density transformation', qd=1)
```


1st PC modes, log quantile–density transformation
tract:cst_I, measure:FA



2nd PC modes, log quantile–density transformation
tract:cst_I, measure:FA

