

# ACOTSP 1000–4500 Estimation of parameter importance with fANOVA

Marcus Ritt

(We focus for now on the acotsp1000-4500-1 data set.)

## 0. Helpers

```
measures=c("perf","norm","quan","rank","irank","qrank")
# wrapping function
wrap.it <- function(x, len) {
  sapply(x, function(y) paste(strwrap(y, len), collapse = "\n"), USE.NAMES = FALSE)
}
resultFile <- function(mname,dir="test1",rep="") {
  paste(dir,"/irace-acotsp1000-4500-1-",mname,"-importance",rep,".dat",sep="")
}
readImportance <- function(fname,tag="") {
  data=read.table(fname,h=T) %>% select(parameter,ind_imp,ind_std) %>% arrange(-ind_imp)
  if (tag!="") {
    data$measure=tag
  }
  colnames(data)[1:3]=c("variable","importance","std_dev")
  return(data)
}
importanceTable <- function(mname) {
  data=readImportance(resultFile(mname))
  kable(data,caption=paste("Importance for measure `",mname,"' (single run)",sep=""), digits=c(0,3,3))
}
readAll <- function() {
  data=data.frame()
  for (mname in measures) {
    data = data %>% rbind(readImportance(resultFile(mname),tag=mname))
  }
  return(data)
}
readReplications <- function(mname) {
  data=data.frame()
  for (r in c(1,2,3,4,5)) {
    data = data %>% rbind(readImportance(resultFile(mname,dir="test2",rep=paste("_r",r,sep="")),tag=r))
  }
  return(data)
}
bumpSingle <- function() {
  data=readAll() %>% group_by(measure) %>% mutate(rank=rank(importance),measure=factor(measure,levels=measures))
  first=data %>% filter(measure=="perf")
  ggplot(data=aes(x=measure,y=rank,group=variable,color=variable))+geom_point()+geom_line()+geom_text(aes(label=rank,dy=-10))
}
```

```

}
bumpPlot <- function(mname) {
  data=readReplications(mname) %>% group_by(measure) %>% mutate(rank=rank(importance))
  first=data %>% filter(measure=="1")
  ggplot(data=data,aes(x=measure,y=rank,group=variable,color=variable))+geom_point()+geom_line()+geom_t
}

```

## 1. Analysis of a single run

### 1.1. Dependent variable: raw performance

```
importanceTable("perf")
```

Table 1: Importance for measure “perf” (single run)

variable	importance	std_dev
instance	0.588	0.149
nnls	0.039	0.062
dlb	0.036	0.052
localsearch	0.016	0.035
q0	0.007	0.010
ants	0.003	0.008
beta	0.002	0.003
alpha	0.002	0.002
rho	0.001	0.001
elitistants	0.000	0.000
dummy	0.000	0.000
rasrank	0.000	0.000
algorithm	0.000	0.000

### 1.2. Dependent variable: normalized performance

```
importanceTable("norm")
```

Table 2: Importance for measure “norm” (single run)

variable	importance	std_dev
localsearch	0.350	0.363
dlb	0.183	0.311
instance	0.127	0.146
nnls	0.109	0.257
alpha	0.010	0.018
rho	0.003	0.004
elitistants	0.002	0.002
beta	0.002	0.001
q0	0.002	0.002
algorithm	0.001	0.001
ants	0.001	0.001
dummy	0.001	0.001
rasrank	0.000	0.000

### 1.3. Dependent variable: performance quantile

```
importanceTable("quan")
```

Table 3: Importance for measure “quan” (single run)

variable	importance	std_dev
localsearch	0.231	0.130
q0	0.052	0.076
elitists	0.024	0.020
nnls	0.021	0.024
algorithm	0.018	0.037
alpha	0.015	0.016
instance	0.011	0.006
dlb	0.006	0.011
beta	0.004	0.008
ants	0.004	0.004
rho	0.002	0.004
dummy	0.001	0.002
rasrank	0.000	0.000

### 1.4. Dependent variable: normalized ranking

```
importanceTable("rank")
```

Table 4: Importance for measure “rank” (single run)

variable	importance	std_dev
instance	0.466	0.099
localsearch	0.113	0.054
q0	0.019	0.014
alpha	0.017	0.026
nnls	0.008	0.012
elitists	0.008	0.011
dlb	0.004	0.006
algorithm	0.003	0.005
ants	0.002	0.002
rho	0.002	0.005
beta	0.000	0.001
dummy	0.000	0.001
rasrank	0.000	0.000

### 1.5. Dependent variable: normalized ranking with imputation

```
importanceTable("irank")
```

### 1.6. Dependent variable: ranking quartile with imputation

```
importanceTable("qrank")
```

Table 5: Importance for measure “irank” (single run)

variable	importance	std_dev
instance	0.080	0.024
localsearch	0.038	0.021
q0	0.011	0.009
beta	0.006	0.008
rho	0.005	0.004
nls	0.004	0.004
alpha	0.004	0.004
algorithm	0.002	0.004
dlb	0.002	0.003
ants	0.001	0.000
elitists	0.001	0.001
rank	0.000	0.001
dummy	0.000	0.000

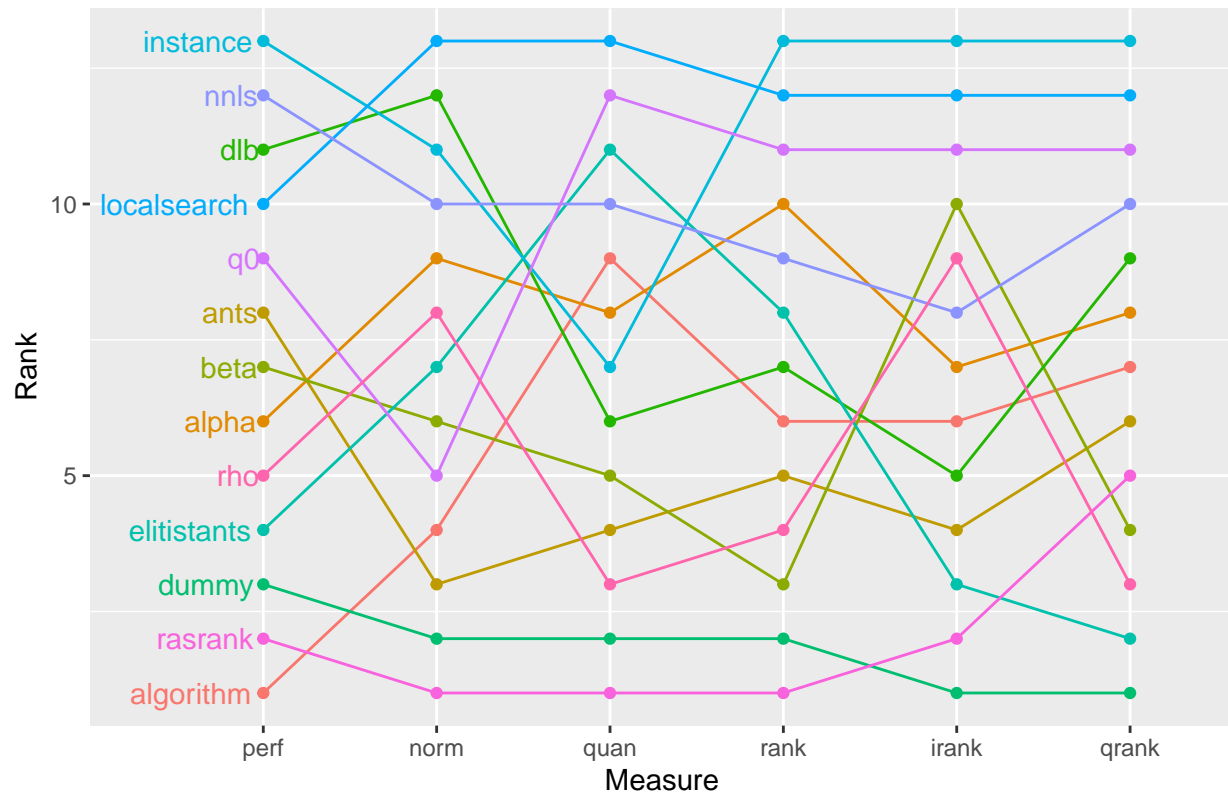
Table 6: Importance for measure “qrank” (single run)

variable	importance	std_dev
instance	0.770	0.022
localsearch	0.004	0.002
q0	0.002	0.001
nls	0.001	0.001
dlb	0.000	0.001
alpha	0.000	0.000
algorithm	0.000	0.001
ants	0.000	0.000
rank	0.000	0.000
beta	0.000	0.000
rho	0.000	0.000
elitists	0.000	0.000
dummy	0.000	0.000

## 2. Comparison of measures among a single run

bumpSingle()

### Ranking under different measures

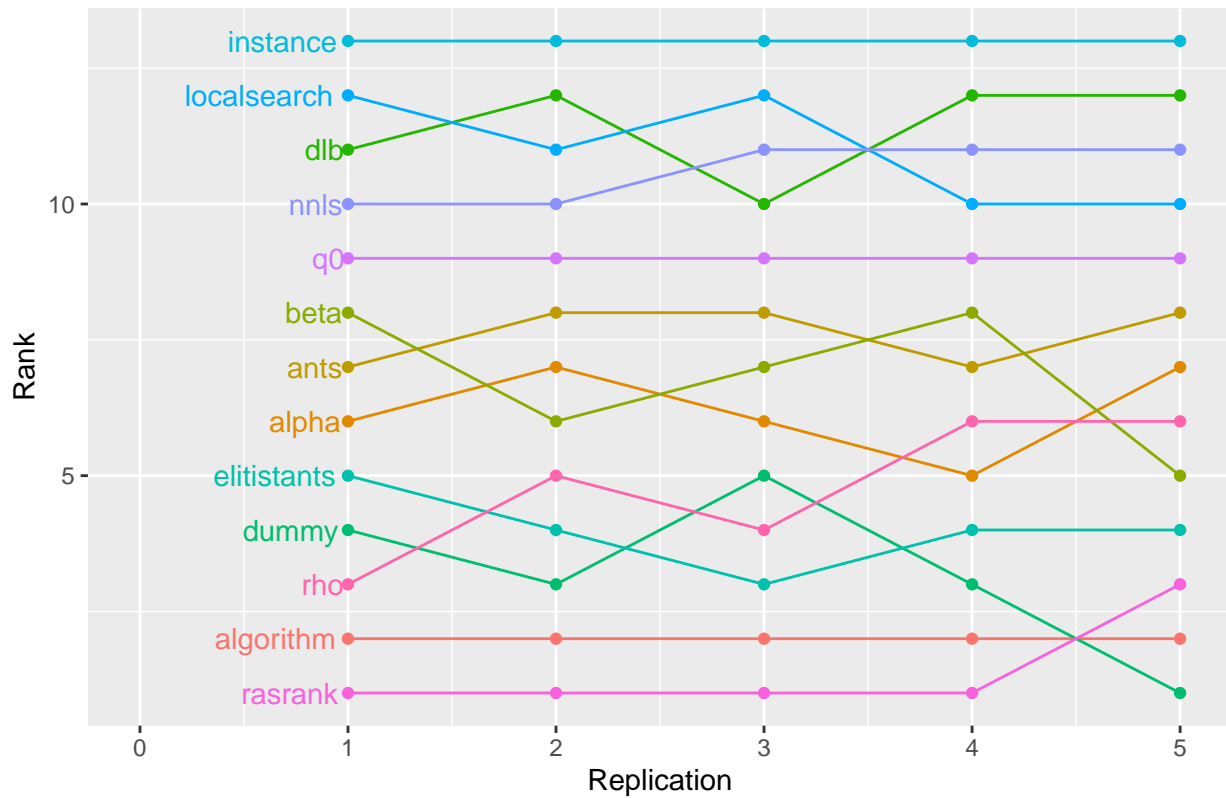


## 3. Comparison of five replications

### 3.1. Dependent variable: raw performance

```
bumpPlot("perf")
```

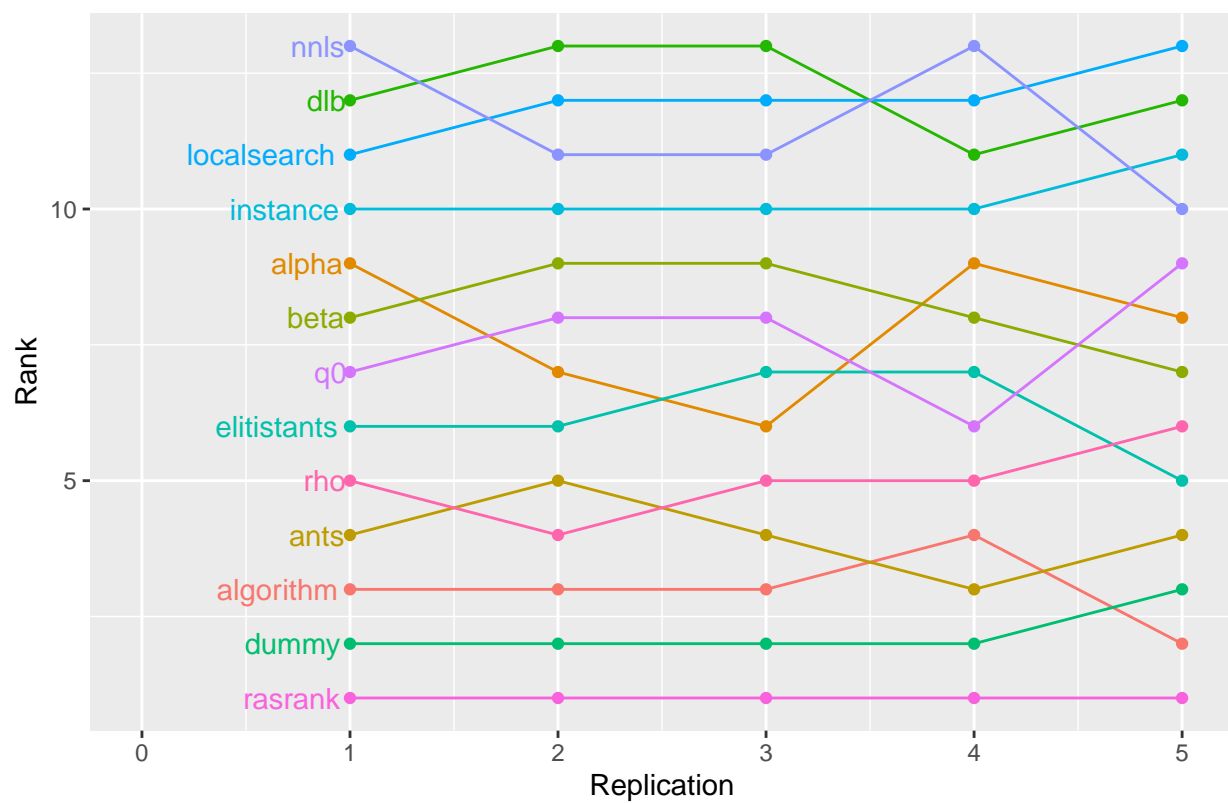
Ranking over different replications for measure “perf”



### 3.2. Dependent variable: normalized performance

```
bumpPlot("norm")
```

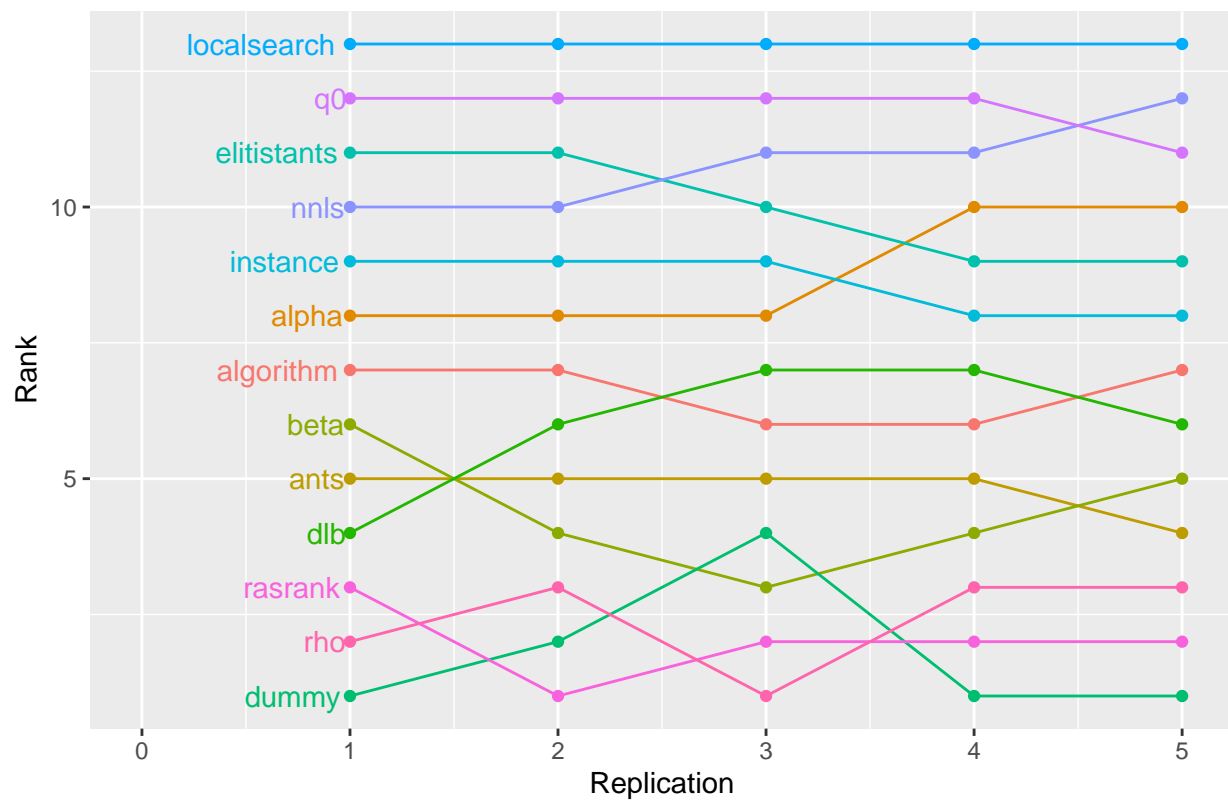
Ranking over different replications for measure “norm”



### 3.3. Dependent variable: performance quantile

```
bumpPlot("quan")
```

Ranking over different replications for measure “quan”

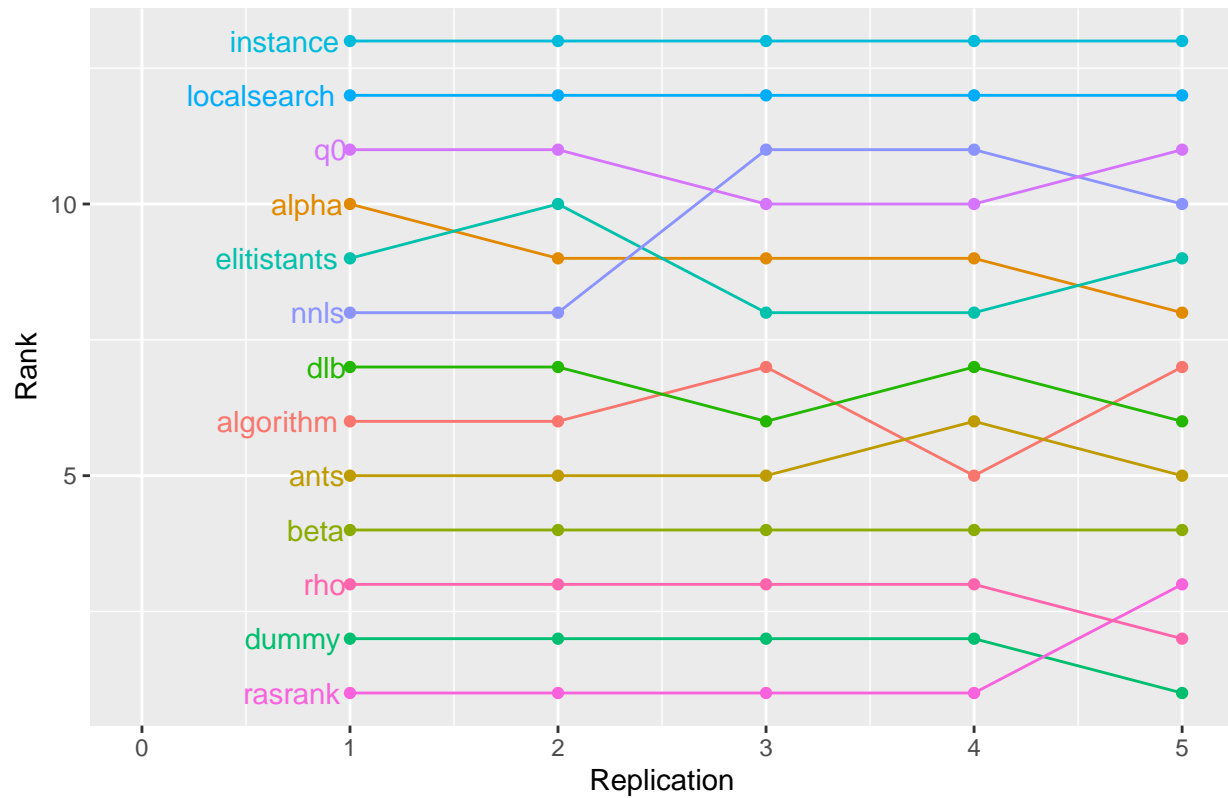


### 3.4. Dependent variable: normalized ranking

```
bumpPlot("rank")
```



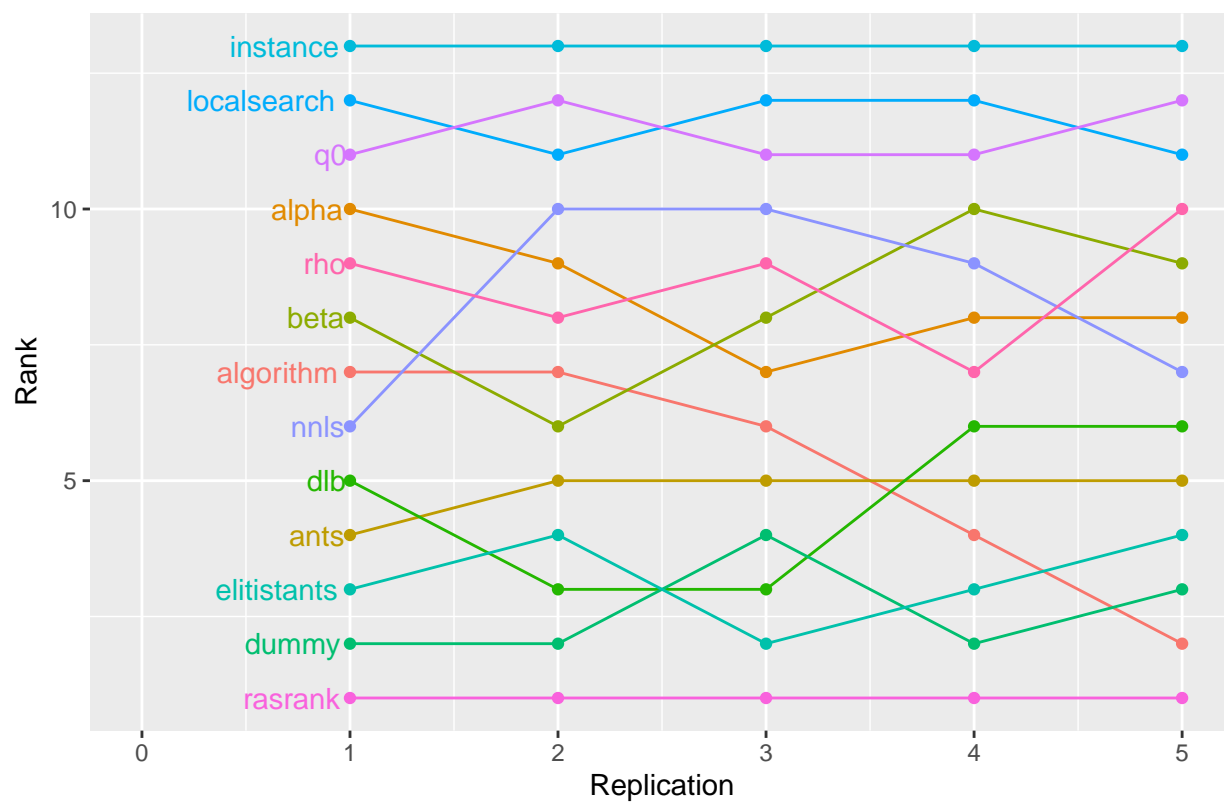
Ranking over different replications for measure “rank”



### 3.5. Dependent variable: normalized ranking with imputation

```
bumpPlot("irank")
```

Ranking over different replications for measure “irank”



### 3.6. Dependent variable: ranking quartile with imputation

```
bumpPlot("qrank")
```

Ranking over different replications for measure “qrank”

