Boosting datasets

Some of our existing datasets contain sub-cellular compartments (classes) with very few markers which makes it difficult to use these data in testing our algorithms. We wish to include as many classes as possible in testing our algorithms as we wish to cover as much subcellular diversity as possible.

Currently, all markers for the datasets * andy2011 * dunkley2006 * E14TG2a * tan2009 have been defined by biologists, who conducted the experiments. These have been definied through manual searching of the literature and consulting databases.

In attempt to gather more markers and boost the above existing marker sets we have pulled GO terms with the cellular compartment namespace which have been experimentally defined and which have a unique term. These are extra terms have been added as markers to the original marker lists and then plotted using `plot2D` to check validitiy with the structure of the data.

The below code demostrated the above procedure.

Dataset `andy2011` - Human HEK 293

```
library("pRoloc2")
library("pRolocdata")
data(andy2011)
fData(andy2011)$UniprotID <- featureNames(andy2011)
featureNames(andy2011) <- fData(andy2011)$Accession.No.
fData(andy2011)$markers <- fData(andy2011)$markers.original
fData(andy2011)$markers <- as.character(fData(andy2011)$pd.markers)

## Set parameters
andy2011params <- setAnnotationParams()

## Get GO annotations assigned to single organelle with EXP only
getGoLoc <- pRoloc2::getGO(andy2011, evidence="EXP", params=andy2011params)
idN <- names(which(table(getGoLoc[,1])==1))
id <- sapply(idN, function(z) which(getGoLoc[,1]==z))
goLoc <- getGoLoc[unlist(id),]
cc <- goLoc[,4]
names(cc) <- goLoc[,1]
allCl <- names(table(cc))
allN <- featureNames(andy2011)
#fData(andy2011)$go <- rep("unknown", nrow(andy2011))
fcol = "markers"

## We now look for terms in `allCl` that match each sub-cellular compartment
## and add these as potential markers (curation will be further downstream)

## 1.ER
nam <- grep("endop", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("ER", length(.id))

## 2.Golgi
```

```
nam <- grep("Golgi", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("Golgi", length(.id))

## 3.Lysosome
nam <- grep("lysoso", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("Lysosome", length(.id))

## 4.Mitochondrion
nam <- grep("mitochond", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("Mitochondrion", length(.id))

## 5.Plasma membrane
nam <- grep("plasma mem", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("PM", length(.id))

## 6.Nucleus
nam <- grep("nucleus", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("Nucleus", length(.id))

## 7.Proteasome
nam <- grep("protea", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("Proteasome", length(.id))

## 8.Cytoskeleton
nam <- grep("cytoske", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("Cytoskeleton", length(.id))

## 9.Ribosome
nam <- grep("ribo", allCl, value=TRUE)
nam <- nam[1]
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
```

```
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("Ribosome (60S)", length(.id))

## 11.Peroxisome
nam <- grep("perox", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("Peroxisome", length(.id))

## 12.Nucleolus
nam <- grep("nucleo", allCl, value=TRUE)
nam <- nam[2]
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("Nucleolus", length(.id))

## 13. Cytosol
nam <- grep("cytosol", allCl, value=TRUE)
nam <- nam[1]
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("Cytosol", length(.id))

## 14. Cytoplasm
nam <- grep("cytoplasm", allCl, value=TRUE)
nam <- nam[1]
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("Cytoplasm", length(.id))

## 15. Chromosome
nam <- grep("chro", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("Chromosome", length(.id))

## 16. Endosome
nam <- grep("endosom", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(andy2011)[.id, fcol] <- rep("Endosome", length(.id))

## Compare annotations
par(mfrow = c(1, 2))
plot2D(andy2011, "pd.markers")
plot2D(andy2011, "markers")
```

Annotations in the `fData` column `go` are then examined via `plot2D` and and then added to `pd.markers`. With this new list of markers we run a round of classification using and SVM and examine any new members to the existing clusters through querying the GO once more, other databases and the literature.

```
opt <- svmOptimisation(andy2011, fcol = "markers")
andy2011 <- svmClassification(andy2011, assessRes = opt)
## Now examine new members of sub-compartments
## curate
save(andy2011, file = "curatedAndy2011.rda")
```

Dataset dunkley2006 - Arabidopsis thaliana

As above, we follow the same procedure e.g. * Use `getGO` to pull unique GO CC terms with experimental evidence * Add to current `pd.markers` * Examine via `plot2D` to check compliance with data structure and remove any that are not. * Perform a round of classification with this new marker list and examine new entries as potential markers by comparing with databases and literature.

```
data(dunkley2006)
#data(dunkley2006params)
dunkley2006params <- setAnnotationParams()

## Get GO.CC information for proteins in dunkley only with EXP evidence
ann <- pRoloc2::getGO(dunkley2006, evidence = "EXP", params = dunkley2006params)

## Now only pull unique
unique <- names(which(table(ann[,1])==1))
cmn <- match(unique, ann[, 1])
ann <- ann[cmn, ]

## Add labels to dunkley
fData(dunkley2006)$markers <- fData(dunkley2006)$markers.original
fData(dunkley2006)$markers <- fData(dunkley2006)$pd.markers
go.cc <- ann[,4]
names(go.cc) <- ann[, 1]
dunkley2006 <- addMarkers(dunkley2006, markers = go.cc, mcol = "go.cc.unique")

mt <- which(fData(dunkley2006)$go.cc.unique=="mitochondrion")
fData(dunkley2006)$markers[mt]
fData(dunkley2006)$markers[mt] <- rep("Mitochondrion", length(mt))

va <- grep("vacuol", fData(dunkley2006)$go.cc.unique)
fData(dunkley2006)$markers[va]
fData(dunkley2006)$markers[va] <- rep("vacuole", length(va))

pt <- grep("plastid", fData(dunkley2006)$go.cc.unique)
fData(dunkley2006)$markers[pt]
fData(dunkley2006)$markers[pt[7]] <- rep("Plastid", length(pt[7]))

cl <- grep("chloroplast", fData(dunkley2006)$go.cc.unique)
fData(dunkley2006)$markers[cl[c(7,9:10)]]
featureNames(dunkley2006)[cl[c(7,9:10)]]
fData(dunkley2006)$markers[cl[c(7,9:10)]] <- rep("Plastid", length(cl[c(7,9:10)]))

pm <- grep("plasma membrane", fData(dunkley2006)$go.cc.unique)
fData(dunkley2006)$markers[pm]
.x <- which(fData(dunkley2006)$markers[pm]=="unknown")
```

```
pm <- pm[.x]
pm <- pm[-c(1,2,9,15)]
fData(dunkley2006)$markers[pm]
featureNames(dunkley2006)[pm]
fData(dunkley2006)$markers[pm] <- rep("PM", length(pm))

## Check for compliance with structure
plot2D(dunkley2006, "markers")

## Further curation

## Now run svm to boost further
opt <- svmOptimisation(dunkley2006, fcol = "markers")
dunkley2006 <- svmClassification(dunkley2006, assessRes = opt)

## Now examine new assignments and add any new markers found within these clusters.
```

Dataset E14TG2A - Mouse

```
library("organelledb")
data("E14TG2aS1")
featureNames(E14TG2aS1) <- fData(E14TG2aS1)$Uniprot.ID

fData(E14TG2aS1)$markers0 <- fData(E14TG2aS1)$markers
fData(E14TG2aS1)$markers <- NULL
fData(E14TG2aS1)$UniprotName <- featureNames(E14TG2aS1)
featureNames(E14TG2aS1) <- fData(E14TG2aS1)$Uniprot.ID

## Load marker list as curated by Claire M Mulvey (CMM) and Andy Christoforou (AC)
load("e14mrk.rda")
E14TG2aS1 <- addMarkers(E14TG2aS1, mrk)

## Add markers from the original 'markers0' and move to 'markers'
## These were the original markers added by AC, check
cy <- which(fData(E14TG2aS1)$markers0 == "CYT")
fData(E14TG2aS1)$markers[cy]
fData(E14TG2aS1)$markers[cy] <- rep("Cytosol", length(cy))
er <- which(fData(E14TG2aS1)$markers0 == "ERT")
fData(E14TG2aS1)$markers[er]
fData(E14TG2aS1)$markers[er] <- rep("Endoplasmic reticulum", length(er))
ly <- which(fData(E14TG2aS1)$markers0 == "LYS")
fData(E14TG2aS1)$markers[ly]
ly <- ly[-2]
fData(E14TG2aS1)$markers[ly]
fData(E14TG2aS1)$markers[ly] <- rep("Lysosome", length(ly))
mt <- which(fData(E14TG2aS1)$markers0 == "MIT")
fData(E14TG2aS1)$markers[mt]
fData(E14TG2aS1)$markers[mt] <- rep("Mitochondrion", length(mt))
# peroxisome already a marker, nuc markers already labelled 'chromatin'
pm <- which(fData(E14TG2aS1)$markers0 == "PLM")
fData(E14TG2aS1)$markers[pm]
```

```
fData(E14TG2aS1)$markers[pm] <- rep("Plasma membrane", length(pm))
# no new ribo

## Get GO loc for EXP evidence only
E14TG2aS1params <- setAnnotationParams()
go.E14 <- getGO(E14TG2aS1, evidence="EXP", params=E14TG2aS1params)
idN <- names(which(table(go.E14[,1])==1))
id <- sapply(idN, function(z) which(go.E14[,1]==z))
goLoc <- go.E14[unlist(id),]
cc <- goLoc[,4]
names(cc) <- goLoc[,1]
allCl <- names(table(cc))
allN <- featureNames(E14TG2aS1)
fData(E14TG2aS1)$go <- rep("unknown", nrow(E14TG2aS1))
fcol = "go"

## 1.ER
nam <- grep("endop", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(E14TG2aS1)[.id, fcol] <- rep("Endoplasmic reticulum", length(.id))

## 2.Golgi
nam <- grep("Golgi", allCl, value=TRUE)
nam <- nam[1:2]
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(E14TG2aS1)[.id, fcol] <- rep("Golgi", length(.id))

## 3.Lysosome
nam <- grep("lysoso", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(E14TG2aS1)[.id, fcol] <- rep("Lysosome", length(.id))

## 4.Mitochondrion
nam <- grep("mitochond", allCl, value=TRUE)
nam <- nam[3]
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(E14TG2aS1)[.id, fcol] <- rep("Mitochondrion", length(.id))

## 5.Plasma membrane
nam <- grep("plasma mem", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(E14TG2aS1)[.id, fcol] <- rep("Plasma membrane", length(.id))

## 6.Nucleus
```

```
nam <- grep("nucleus", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(E14TG2aS1)[.id, fcol] <- rep("Nucleus", length(.id))

## 7.Proteasome
nam <- grep("protea", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(E14TG2aS1)[.id, fcol] <- rep("Proteasome", length(.id))

## 8.Cytoskeleton
# no markers

## 9.Ribosome
nam <- grep("ribo", allCl, value=TRUE)
nam <- nam[1]
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(E14TG2aS1)[.id, fcol] <- rep("60S Ribosome", length(.id))

## 11.Peroxisome
# none

## 12.Nucleolus
nam <- grep("nucleo", allCl, value=TRUE)
nam <- nam[1]
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(E14TG2aS1)[.id, fcol] <- rep("Nucleus - Nucleolus", length(.id))

## 13. Cytosol
nam <- grep("cytosol", allCl, value=TRUE)
nam <- nam[1]
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(E14TG2aS1)[.id, fcol] <- rep("Cytosol", length(.id))

## 14. Cytoplasm
nam <- grep("cytoplasm", allCl, value=TRUE)
nam <- nam[1]
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(E14TG2aS1)[.id, fcol] <- rep("Cytoplasm", length(.id))

## 15. Chromatin
# none

## 16. Actin
nam <- grep("actin", allCl, value=TRUE)
```

```
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(E14TG2aS1)[.id, fcol] <- rep("Actin cytoskeleton", length(.id))

## 16. ECM
nam <- grep("matrix", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(E14TG2aS1)[.id, fcol] <- rep("Extracellular matrix", length(.id))

## Now look at fcol = "go" in plot2D and see if correlate with LOPIT

## Remove any outliers and check markers fit the structure of the data.
```

Dataset `tan2009r1` - Drosophila melanogaster

Again, as above we pull unique GO CC terms to see if we can boost the number of markers for each sub-cellular compartment in `pd.markers`. We then examine these new additions using `plot2D` to check they correlate well with the existing data structure reflective of the experiment. We only find 21 new proteins we can use as markers.

```
data(tan2009r1)
fData(tan2009r1)$FBgn <- featureNames(tan2009r1)
featureNames(tan2009r1) <- fData(tan2009r1)$AccessionNo

## Get GO.CC information for proteins in dunkley only with EXP evidence
tan2009params <- setAnnotationParams()
go.tan <- getGO(tan2009r1, evidence = "EXP", params = tan2009params)

idN <- names(which(table(go.tan[,1])==1))
id <- sapply(idN, function(z) which(go.tan[,1]==z))
goLoc <- go.tan[unlist(id),]
cc <- goLoc[,4]
names(cc) <- goLoc[,1]
allCl <- names(table(cc))
allN <- featureNames(tan2009r1)
fData(tan2009r1)$markers.new <- fData(tan2009r1)$pd.markers
fcol = "markers.new"

## 1.ER
nam <- grep("endop", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(tan2009r1)[.id, fcol] <- rep("ER", length(.id))

## 2. Cytoskeleton
nam <- grep("cytoskel", allCl, value=TRUE)
nam

## 3. Golgi
```

```
nam <- grep("Golgi", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(tan2009r1)[.id, fcol] <- rep("Golgi", length(.id))

## 4. Lysosome
nam <- grep("lysos", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(tan2009r1)[.id, fcol]

## 5. mitochondrion
nam <- grep("mitoc", allCl, value=TRUE)
nam <- nam[2]
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(tan2009r1)[.id, fcol] <- "mitochondrion"

## 6. nucleus
nam <- grep("nucleus", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(tan2009r1)[.id, fcol]
fData(tan2009r1)[.id, fcol] <- rep("Nucleus", length(.id))

## 7. Peroxisome
nam <- grep("perox", allCl, value=TRUE)
nam

## 8. PM
nam <- grep("plasma mem", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(tan2009r1)[.id, fcol]
fData(tan2009r1)[.id, fcol] <- rep("PM", length(.id))

## 9. Proteasome
nam <- grep("proteasome", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
.id <- sapply(nam, function(z) which(allN==z))
fData(tan2009r1)[.id, fcol]
fData(tan2009r1)[.id, fcol] <- rep("Proteasome", length(.id))

## 10/11. ribosomes
nam <- grep("ribo", allCl, value=TRUE)
nam
nam <- goLoc[unlist(sapply(nam, function(z) which(cc==z))), 1]
```

```
.id <- sapply(nam, function(z) which(allN==z))
fData(tan2009r1)[.id, fcol]
featureNames(tan2009r1)[.id]
.id.40s <- .id[6]
.id.60s <- .id[-6]

fData(tan2009r1)[.id.40s, fcol] <- rep("Ribosome 40S", length(.id.40s))
fData(tan2009r1)[.id.60s, fcol] <- rep("Ribosome 60S", length(.id.60s))

## Now plot and remove any outliers
.torm <- plot2D(tan2009r1, fcol = "markers.new", identify = TRUE)
fData(curatedTan)$markers.new[.torm] <- rep("unknown", length(.torm))

## rename columns in fData
fData(tan2009r1)$markers.orig <- fData(tan2009r1)$markers
fData(tan2009r1)$markers <- fData(tan2009r1)$markers.new
fData(tan2009r1)$markers.new <- NULL
fvarLabels(tan2009r1)
#save(tan2009r1, file = "../sandbox/curatedDatasets/curatedTan2009r1.rda")
```