

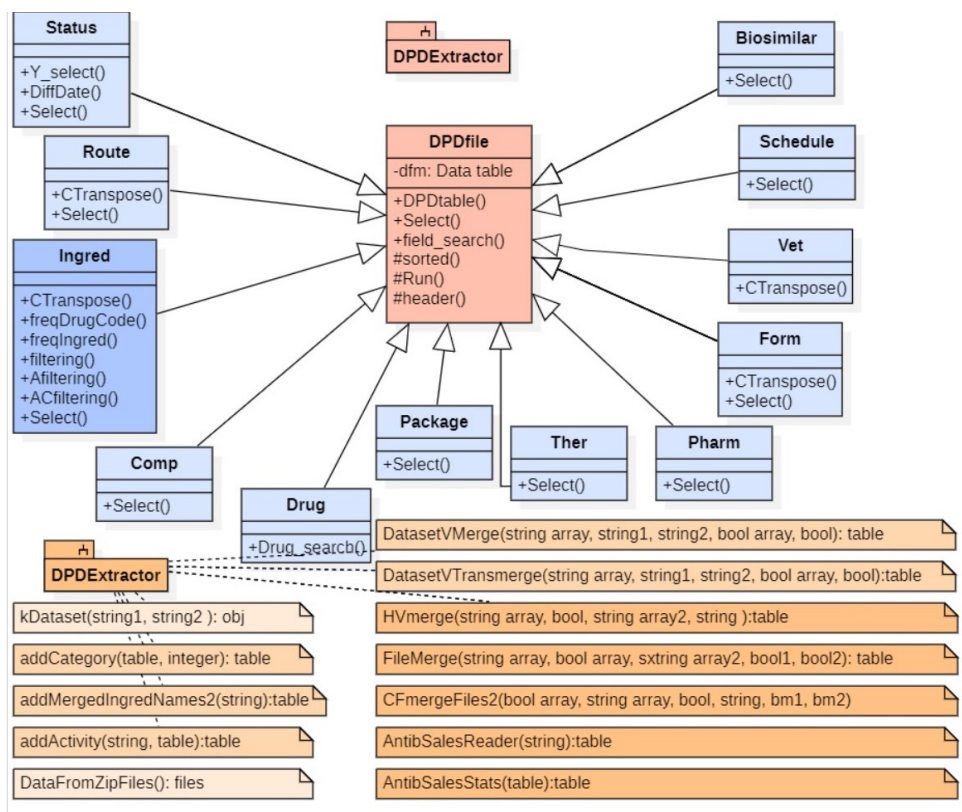
# Introduction to DPDdataextractor3

The DPDataExtractor3 package was developed for the processing of the Government of Canada's Drug Product Database (DPD) files. A modularized and object-oriented design was adopted not only to ensure the greatest possible flexibility in terms of the formats of the tables that can be generated in relation to the number, identity and order of the columns, but also to facilitate completely different future developments of this tool within the field of drug utilization statistics. Although the emphasis was put for this version of the package on the processing of information related to antibiotics, it could be used for any drug.

```
library(DPDextractor3)
```

The package consists of two scripts, DPDFunctions.R and DPDEXtractor.R, the first one contains 14 global functions that directly or indirectly use the methods of the DPDEXtractor library. DPDEXtractor contains 13 portable R6 classes that group 42 methods. These classes are interconnected by inheritance relationships in a star structure to a central class, DPDfile, from which they inherit some basic methods. Each class corresponds to a DPD database table, many of these classes contain only the most basic methods such as constructors and table column selectors. These classes could serve as a basis for possible future extensions of the library.

These 14 global functions use directly or undirectly DPDEXtractor methods: AntibSalesReader(), AntibSalesStats(), DataFromZipfiles(). CFmergeFiles2(), FileMerge() and HVmerge(), datasetVTransmerge(), datasetVmerge(), addActivity(), AddMergedIngredNames(), AddMergedIngredNames2(), addExtraVetcols(), addCategory(), and kdataset().



Package diagram

## Examples of the use of high-level functions: DataFromZipfiles(), CFmergeFiles2() y FileMerge()

```
---  
DataFromZipfiles()  
CFmergeFiles2()  
---
```

The `DataFromZipfiles()` function downloads and unzips the DPD files. Therefore, all global functions of the package require the prior execution of this function. `CFmergeFiles2()` is a pipeline that concatenates the execution of several functions, each of which performs the merging of the rows corresponding to the approved, marketed, dormant and cancelled drugs for all the DPD tables. Two functions, `datasetVTransmerge()` and `datasetVmerge()`, are mainly used in this pipeline for merging and selective transposition of tables. The first function can perform two types of transposition, For the first type, all different values are transformed into columns or variables, and for the second type, the maximum number of values assigned to the same drug code is transformed into columns or variables.

```
---  
DataFromZipfiles()  
  
CFmergeFiles2(fout="superTabIngDrugrouteformCompStaVetPack_CF1.csv", bm1=0, bm2=0, extrac=TRUE)  
CFmergeFiles2(fout="superTabIngDrugrouteformCompStaVetPack_CF3.csv", bm1=1, bm2=1, extrac=FALSE)  
CFmergeFiles2(fout="superTabIngDrugrouteformCompStaVetPack_CF4.csv", bm1=2, bm2=2, extrac=TRUE)  
---
```

Another function such as `FileMerge()` generates separate csv files from each of the DPD tables. Other functions such as `ddActivity()`, `AddMergedIngredNames2()`, `addExtraVetcols()` and `addCategory()` incorporate in the generated tables, fields not found in the DPD database.

```
---  
txt_files <- list.files(path = getwd(), pattern = "txt")
```

```

FileMerge(txt_files)
FileMerge(txt_files, b=c(0,0,0,0))
FileMerge(txt_files, b=c(1,1,0,1))
FileMerge(txt_files, b=c(1,1,1,1))
FileMerge(txt_files, b=c(0,0,1,1), dpdlabel=c("ingred", "form", "route","vet"))
FileMerge(txt_files, b=c(0,0,0,0), dpdlabel=c("drug","status", "route", "form", "comp","ingred",
"package", "vet"), filt=FALSE, outf=TRUE)
---
```

## Examples of the use of functions: datasetVTransmerge(), datasetVmerge(), addMergedIngredNames2, addCategory()y addActivity()

```

---
txt_files <- list.files(path = getwd(), pattern = "txt")

ing<-datasetVTransmerge(txt_files, dpdlabel="ingred")
write.csv(ing, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/transTabIA.csv", row.names=FALSE)

ingtf<-datasetVTransmerge(txt_files, dpdlabel="ingred",f="listAntibiodine.txt", stat=c(1, 1, 1, 1),
filt=TRUE)
write.csv(ingtf, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/transTabIAAtf.csv", row.names=FALSE)

ingf<-datasetVmerge(txt_files, dpdlabel="ingred")
write.csv(ingf, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/TabIA.csv", row.names=FALSE)

sta<-datasetVmerge(txt_files, dpdlabel="status")
write.csv(sta, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/TabStatus.csv", row.names=FALSE)

drug<-datasetVmerge(txt_files, dpdlabel="drug")
```

```

transroute<-datasetVmerge(txt_files, dpdlabel="route")
write.csv(transroute, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/transTabRoute.csv",
row.names=FALSE)

form<-datasetVmerge(txt_files, dpdlabel="form")
write.csv(form, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/TabForm.csv", row.names=FALSE)

com<-datasetVmerge(txt_files, dpdlabel="comp")
write.csv(com, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/TabCom.csv", row.names=FALSE)

pack<-datasetVmerge(txt_files, dpdlabel="package")
write.csv(pack, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/TabPack.csv", row.names=FALSE)

the<-datasetVmerge(txt_files, dpdlabel="ther")
write.csv(the, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/TabTher.csv", row.names=FALSE)

transve<-datasetVTransmerge(txt_files, dpdlabel="vet")
write.csv(transve, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/transTabVet.csv", row.names=FALSE)

#drug ingred form route AddMergedIngred AddCategory AddActivity Status Com Pack Vet

tab<-read.csv("D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/TabDrug.csv", header=TRUE,
stringsAsFactors=FALSE)
tab2<-read.csv("D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/transTabIA.csv", header=TRUE,
stringsAsFactors=FALSE)
tab <- merge(x=tab,y=tab2,by="DRUG_CODE", all.y=T)

write.csv(tab, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/superTab0.csv", row.names=FALSE)

tab2<-read.csv("D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/TabForm.csv", header=TRUE,
stringsAsFactors=FALSE)

```

```

tab <- merge(tab,tab2,by="DRUG_CODE", all.tab=T)
tab[, 'DRUG_CODE']<- as.character(tab[, 'DRUG_CODE'])
write.csv(tab, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/superTab2.csv", row.names=FALSE)
updatedTab<-AddMergedIngredNames2("superTab2.csv")
updatedTab<-AddMergedIngredNames2(tab)
write.csv(updatedTab, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/superTab21.csv",
row.names=FALSE)

updatedTable<-addCategory("superTab21.csv")
updatedTable<-addCategory(updatedTable)
write.csv(updatedTable, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/superTab3.csv",
row.names=FALSE)

actTable <-addActivity ("activity2.txt", "superTab3.csv")
actTable<-addActivity ("activity2.txt",updatedTable)
write.csv(actTable, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/actTable.csv", row.names=FALSE)

tab2<-read.csv("D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/TabStatus.csv", header=TRUE,
stringsAsFactors=FALSE)
tab <- merge(tab=actTable,tab2=tab2,by="DRUG_CODE",all.tab=T)

tab[, 'DRUG_CODE']<- as.character(tab[, 'DRUG_CODE'])
write.csv(tab, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/superTab3.csv", row.names=FALSE)

tab2<-read.csv("D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/TabCom.csv", header=TRUE,
stringsAsFactors=FALSE)
tab <- merge(tab,tab2,by="DRUG_CODE", all.tab=T)
tab[, 'DRUG_CODE']<- as.character(tab[, 'DRUG_CODE'])
write.csv(tab, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/superTab4.csv", row.names=FALSE)

```

```

tab <- merge(tab,tab2,by="DRUG_CODE", all.tab=T)
tab[, 'DRUG_CODE'] <- as.character(tab[, 'DRUG_CODE'])
write.csv(tab, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/superTab2.csv", row.names=FALSE)
updatedTab<-AddMergedIngredNames2("superTab2.csv")
updatedTab<-AddMergedIngredNames2(tab)
write.csv(updatedTab, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/superTab21.csv",
row.names=FALSE)

updatedTable<-addCategory("superTab21.csv")
updatedTable<-addCategory(updatedTable)
write.csv(updatedTable, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/superTab3.csv",
row.names=FALSE)

actTable <-addActivity ("activity2.txt", "superTab3.csv")
actTable<-addActivity ("activity2.txt",updatedTable)
write.csv(actTable, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/actTable.csv", row.names=FALSE)

tab2<-read.csv("D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/TabStatus.csv", header=TRUE,
stringsAsFactors=FALSE)
tab <- merge(tab=actTable,tab2=tab2,by="DRUG_CODE",all.tab=T)

tab[, 'DRUG_CODE'] <- as.character(tab[, 'DRUG_CODE'])
write.csv(tab, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/superTab3.csv", row.names=FALSE)

tab2<-read.csv("D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/TabCom.csv", header=TRUE,
stringsAsFactors=FALSE)
tab <- merge(tab,tab2,by="DRUG_CODE", all.tab=T)
tab[, 'DRUG_CODE'] <- as.character(tab[, 'DRUG_CODE'])
write.csv(tab, "D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/superTab4.csv", row.names=FALSE)

```

```

tab2<-read.csv("D:/WIAAgnosante/Mapaq-Antibioticos/DPDataExtractor/TabPack.csv", header=TRUE,
stringsAsFactors=FALSE)
tab <- merge(tab,tab2,by="DRUG_CODE", all=T)
tab[, 'DRUG_CODE']<- as.character(tab[, 'DRUG_CODE'])
write.csv(tab, "D:/WIAAgnosante/Mapaq-Antibioticos/DPDataExtractor/superTab5.csv", row.names=FALSE)

tab2<-read.csv("D:/WIAAgnosante/Mapaq-Antibioticos/DPDataExtractor/transTabVet.csv", header=TRUE,
stringsAsFactors=FALSE)
tab <- merge(tab,tab2,by="DRUG_CODE", all=T)
tab[, 'DRUG_CODE']<- as.character(tab[, 'DRUG_CODE'])
tab<-tab[!is.na(tab$CLASS),]
write.csv(tab, "D:/WIAAgnosante/Mapaq-Antibioticos/DPDataExtractor/superTabtrouteformPack0.csv",
row.names=FALSE)
---
```

## Examples of the use of various methods of DPDextractor classes:

```

---
path<-getwd()
dir<-paste0(path,"/", "route.txt")

obj1<-DPDfile$new(dir)
obj1$DPDtable

obj1$field_search("DRUG", TRUE)
obj1$field_search("PRODUCT")
```

```

---
path<-getwd()
dir<-paste0(path,"/", "vet.txt")

obj2<-vet$new(dir)
obj2$select()

Tab2<-obj2$DPDtable
transTab2<-obj2$CTranspose()

dir2<-paste0(path,"/", "Tab_vet_Marketed.csv")

dir3<-paste0(path,"/", "transTab_vet_Marketed.csv")

write.csv(Tab2, dir2, row.names=FALSE)

write.csv(transTab2, dir3, row.names=FALSE)

---

path<-getwd()
dir<-paste0(path,"/", "route.txt")

obj3<-route$new(dir)

Tab<-obj3$DPDtable
transTab<-obj3$CTranspose()

dir2<-paste0(path,"/", "Tab_route_Marketed.csv")

dir3<-paste0(path,"/", "transTab_route_Marketed.csv")

```



```

write.csv(Tab, dir2, row.names=FALSE)

write.csv(transTab, dir3, row.names=FALSE)

---

path<-getwd()
dir<-paste0(path,"/", "listAntibiodine.txt")

dfmanti<- read.table("D:/WIAAgrosante/Mapaq-Antibioticos/DPDataExtractor/listAntibiodine.txt",sep=';',
header=TRUE)

Antib_list<-dfmanti[, "AIwhocc"]
AntiFam_list<-dfmanti[, "GroupeChimSC1"]

---

path<-getwd()
dir<-paste0(path,"/", "ingred.txt")

obj1<-ingred$new(dir)
obj1$select()
obj1$Afiltering(Antib_list, AntiFam_list, b=TRUE, verbose=TRUE)
Tab1<-obj1$DPDtable
dir2<-paste0(path,"/", "df_ingred_Marketed.csv")

write.csv(Tab1, dir2, row.names=FALSE)

---

```

```

---

obj1<-ingred$new(dir)
obj1$select()

obj1$Afiltering(Antib_list, AntiFam_list)
Tab1<-obj1$DPDtable
transtab0<-obj1$CTranspose()

dir3<-paste0(path,"/", "df_Transtab_ingred_Marketed.csv")
write.csv(transtab0, dir3, row.names=FALSE)

---

path<-getwd()
dir<-paste0(path,"/", "status.txt")

obj3<-status$new(dir)
obj3$Y_select()
obj3$select()
Tab3<-obj3$DPDtable
dir3<-paste0(path,"/", "df_status_Marketed.csv")

write.csv(Tab3, dir3, row.names=FALSE)

---

```