MetaMass: tools for mass spectrometry data meta-analysis

Jan Stuchlý, Fridtjof Lund-Johansen

April 27, 2016

jan.stuchly@lfmotol.cuni.cz

1 Introduction

The presented package provides tools for meta-analysis of human proteome mass spectrometry (MS) data as described in (ref Lund-Johansen et al 2016). The purpose is to provide stand-alone tool for analyzing text files or `data.frame` with an integration with R for further analysis.
Area of usage: MetaMass is a tool for meta-analysis of sub-cellular proteomics data (ref Lund-Johansen et al 2016). Users can analyze mass spectrometry (MS) data within the context of published datasets and sets of markers identified by mining of MS datasets or curated annotations from Uniprot, GO and the Human Protein Atlas. The input is text files with official human gene symbols as protein identifiers and normalized MS signal values measured in sub-cellular fractions. The output is a cdt file for visualization of the classified datasets as heatmaps in JavaTreeView, a table with with gene names, annotations, assigned locations and purity scores and precision-recall curves that provide information about the fit between the dataset and the marker set.
Required software:

R (https://cran.r-project.org)

R Studio (https://www.rstudio.com) - for more user-friendly R-front-end

Rtools (https://cran.r-project.org/bin/windows/Rtools/) necessary on MS Windows operating system only

JavaTreeView (http://jtreeview.sourceforge.net/) - for visualisation of heatmaps (the .cdt files - see below)

```
> install.packages("devtools") ##  Install devtools from R
> library(devtools) ## load devtools
> install_github("stuchly/MetaMass") ## Install MetaMass
> library(MetaMass) ## load MetaMass
> vignette("MetaMass") ## see this vignette
```

2 Input

On input user provides a `data.frame` containing the MS data or path to text file(s) with the data. Data are then processed with respect to following conventions:

- All columns containing numerical values only are used as MS data
- MS data could be divided into different groups" if separated by blank/non-numeric column i.e. numeric columns flanked by non-numeric columns are understood as a single "group". With more file on input each file is considered as separate set of groups.
- `data.frame` must contain a column containing protein ID which match the annotation file (by default the genename in the first column;see below)

- by default tab-delimited text file is assumed

An annotation file is provided in the package

```
> library(MetaMass)
> data(AnnotationAM)
> head(AnnotationAM)

  Uniprot.Accession   Gene Christoforou Uniprot   G.O HPA.supportive
1            P04217   A1BG
2            Q9NQ94   A1CF                                  Nucleus
3            P01023    A2M
4            A8K2U0  A2ML1
5            Q9NPC4 A4GALT               Golgi Golgi
6            Q9UNA3  A4GNT               Golgi Golgi
  HPA.uncertain
1
2
3
4
5
6
```

2.1 Troubleshooting

The common challenge for new users of R is data import. The function `analyze.MSfile` expects tab-delimited file which could be read via function `read.table()`

```
> data_table<-read.table(filename,header=TRUE,sep="\t")
```

If there is an error concerning reading the input file, the user can try this function to check if the file is in correct format. The other possible issue is the grouping of the data columns. As mentioned above each contiguous sequence of numerical columns (flanked by non-numeric column) is considered as one group - the user can check if all (and only) the data he wants to analyze would be considered as MS data as follows

```
> colnames(data_table)[sapply(data.table,is.numeric)]
```

3 Output

By default (with parameter `output=NULL`) the user level function `analyze.MSfile` does not create any files in the working directory and returns named list to be analyzed within R (see below). However most of the users are expected to specify the parameter `output="name"` and inspect the results outside R. In this case three files will be created in the working directory.

- name_table.txt - spreadsheet containing the analyzed data and all annotations used (see below) for the analysis together the cluster assignment and purity with respect to the most abundant component
- name_pr.pdf - precision-recall curves of the cluster assignment with respect to all used annotations
- name_javatree.cdt - heatmap to be visualized in the Java TreeView application. Each line is annotated by the protein ID, annotation (Annot=;if present for this protein) and assignment (assign=;assignment of the cluster containing this protein)

4 Metadata

The package is distributed with MS data which could be used to reproduce the results in the paper or as a reference for user supplied data - these data are called Metadata - see `?Metadata`. The Metadata can be used as `data.frame` or if the user want to open then in a spreadsheet editor their location on the computer can be found via `system.file` function.

```
> filename<-system.file("extdata","Bileck.txt",package="MetaMass")
> filename

[1] "/Users/grigorij/Library/R/3.2/library/MetaMass/extdata/Bileck.txt"
```

5 Walkthough

In this section we give detailed description how create and analyze the MS data files. This section is intended for users with minimal experience with R. If the reader wants to avoid downloading a creating this files he can find the same examples in the section Examples where the same internally stored datasets are used. See `?analyze.MSfile` for detailed information.
Users should first retrieve supplementary table 1 from Lund-Johansen et al. ref. and follow instructions in the table to retrieve the following datasets: Fig. 1b (Table 1.5), Datasets 4, 9 and 10 (Table 1.7, use the filtering option explained in the text box in columns in columns DP:EC). The files should be saved in a new folder as tab-delimited text and named "Data_Fig1b.txt" "study4.txt", "study9.txt" and "study10.txt", respectively. Set this folder as the working directory in R-studio. (type ctrl_shift_H and navigate to the folder).

5.1 Analysis option 1

Generate the heatmap and the classification output files corresponding to Fig. 1b in the article: (typed commands are in script font, those below can be copied and pasted into R-studio, complete commands by pressing the enter key.)

```
> analyze.MSfile(MSfile = "Data_Fig1b.txt", Metadata = "Christoforou", output = "Fig1b")
```

Tip. *Typing in R-Studio is greatly simplified using the tab key to auto-complete the command.. Thus, the command above can be typed as follows: ana [tab], msf [tab], "dat [tab], meta [tab], "Christoforou"), out [tab], "Fig1b")*

Explanation: analyze.MSfile: name of function used to perform all the analyses. MSfile: dataset to be analyzed, Metadata: The "Christoforou" dataset is one of several stored in MetaMass as reference data. For a complete list, call the user manual for the program by typing `?Metadata`. The meta-data is not clustered or used for the classification, but simply aligned to the processed test dataset as reference. Also note that MetaMass-R generates a dataset corresponding to the overlap of the test data and the metadata. The Excel version uses all data from the test data, and ignores missing data in the metadataset. output: file-prefix for the output files.

Result: The output files are found in the working directory. All output files have the prefix "Fig1b". The Fig1b.cdt file is a heatmap file (JavaTreeView), the Fig.1b_table.txt is the classification result table (e.g. Excel), and the Fig1b_pr-pdf contains precision-recall curves (e.g. Acrobat Reader).

5.2 Analysis option 2

Generate recall precision curves for multiple different marker sets (i.e similar to those in Fig. 2a in the article).

```
> analyze.MSfile(MSfile = "Data_Fig1b.txt", Metadata = "Christoforou", output = "Fig2acurves", markers = c(3:7))
```

Tip. *This command is very similar to the one used in Example 1. Any command typed previously can be brought back by pressing the arrow up key (and arrow down to move to the next) To modify the command used in example 1, simply type the arrow up key and change the output to "Fig2acurves" and add markers = c(3:7).*

Explanation: The parameter `markers =` is used to analyze data in context of markers other than the Christoforou marker set, which is default. The marker sets have numbers from 4-7 (3= Christoforou, 4= Uniprot, 5= GO, 6= HPA supportive, 7= HPA uncertain). To use all markers in parallel: `markers= c(3:7)`. The heatmap will always correspond to the first marker in the sequence.
Result: The Recall-Precision curves show results obtained with all the marker sets in Fig 2a. The table contains columns for all the marker sets. The heatmap is the same as in Example 1.

5.3 Analysis option 3

Generate heatmaps to compare the mapping result obtained using different marker sets.

```
> analyze.MSfile(MSfile = "Data_Fig1b.txt", Metadata = "Christoforou", output = "Fig1bUniprot", markers =4)
```

Tip. *type arrow up to bring back the command from example 2, replace the text for output with "Fig1b_Uniprot" and the text for markers with* `markers = 4`*). Next time:* `output="Fig1b_HPAs"`, `markers =5` *etc. With this approach it takes very little time to make heatmaps for all marker sets.*

Result: The heatmap generated using the Uniprot marker set is rather similar to that obtained using the Christoforou marker set (Fig1b article, example 1). With markers from the Human Protein Atlas, the area in the map assigned to the nucleus is much larger, and many of the proteins assigned to the nucleus are found in the cytoplasmic fractions. The heatmap obtained with the "uncertain" annotations from the HPA has essentially no structure. Thus, there is very little correspondence with the MS data. In the output table, the mapped locations are listed alongside annotations from Uniprot, GO, and the HPA.

5.4 Analysis option 4

Generate recall-precision curves for single datasets (i.e. similar to Fig. 2b in article)

```
> analyze.MSfile(MSfile = "study4.txt", Metadata = "Christoforou", output = "study4")
```

Result: The recall response curves for cytosol and nucleus are similar to those for study 4 in Fig2a.

5.5 Analysis option 5

see Example 3 Perform a meta-analysis of datasets 4, 9 and 10.

```
> analyze.MSfile(MSfile = c("study4.txt","study9.txt", "study10.txt"), Metadata = "Christoforou", output = "study4910")
```

Explanation: c("study4.txt","study9.txt", "study10.txt") is used to merge data from multiple files into one analysis. The file contains the overlap of these three and the Christoforou dataset used as metadata.
Result: The fractionation methods used in studies 4, 9 and 10 have limited resolution. However, combined they have high resolution. Thus, study 4 has high resolution of mitochondria, but poor separation of ER and nuclei. Study 9 has good separation of cytoplasmic organelles and nuclei, but no resolution of ER, mitochondria or cytosol. Study 9 has good resolution of cytosol, membranes and nuclei, but does not discriminate ER from mitochondria. When the datasets are combined, they complement each other. See figure 1

5.6 Additional user-selectable variables

Number of groups in K-means clustering : The default option is 500 groups. The optimal number of proteins per group is 10-20. If the dataset has more than 5000 proteins or less than 3000 proteins users may specify the number of groups accordingly. The command clusters= 250 specifies 250 clusters. Comparison metrics: The default option is Eucledian distance. The command... metric ="correlation" specifies Pearson correlation (see `?analyze.MSfile`).

6 Analysis within R

Although the main purpose of this package is to create annotated lokup tables and heatmaps which can be conveniently analyzed outside R the results can be naturally treated as any R object. The function `analyze.MSfile` returns named list containing the original data, annotation(s) and cluster assignments. Two function can be used to extract it's contents - see `?get.data` and `?get.clusters`.

```
> file2<-system.file("extdata","Data_Fig_1b.txt",package="MetaMass")
> ##cluster with respect MSfile only (cluster.metadata=FALSE by default)
> res2<-analyze.MSfile(MSfile=file2,Metadata=c("Christoforou"),output="res2",markers=c(3:5))
> data2<-get.data(res2,data.only=TRUE)
> cls2_1<-get.clusters(res2,rID=1) #rID=1 annotation with respect to markers[1]; default
> head(cls2_1)
```

```
  cluster CYTOSOL MITOCHONDRION NUCLEUS PM LYSOSOME ER CS RIBOSOME ENDOSOME
1       1       0             1       0  0        0  0  0        0        0
2       2       0             0       4  1        0  0  0        0        0
3       3       0             0       0  0        0  0  0        0        0
4       4       0             0       1  0        0  1  0        0        0
5       5       1             0       0  0        0  0  0        0        0
6       6       0             0       1  0        0  0  0        1        0
  Nb_of_annotations purity_main_component main_component CYTOSOL_ratio
1                 1                   1.0  MITOCHONDRION             0
2                 5                   0.8        NUCLEUS             0
3                 0                    NA           <NA>           NaN
4                 2                   0.5        NUCLEUS             0
5                 1                   1.0        CYTOSOL             1
6                 2                   0.5        NUCLEUS             0
  MITOCHONDRION_ratio NUCLEUS_ratio PM_ratio LYSOSOME_ratio ER_ratio CS_ratio
1                   1           0.0      0.0              0      0.0        0
2                   0           0.8      0.2              0      0.0        0
3                 NaN           NaN      NaN            NaN      NaN      NaN
4                   0           0.5      0.0              0      0.5        0
5                   0           0.0      0.0              0      0.0        0
6                   0           0.5      0.0              0      0.0        0
  RIBOSOME_ratio ENDOSOME_ratio assigned_location Nb_main_component
1            0.0              0     MITOCHONDRION                 1
2            0.0              0           NUCLEUS                 4
3            NaN            NaN              <NA>                NA
4            0.0              0           NUCLEUS                 1
5            0.0              0           CYTOSOL                 1
6            0.5              0           NUCLEUS                 1
  Nb_assigned_location purity_assigned_location updated_order
1                    1                      1.0           344
2                    4                      0.8           413
3                   NA                       NA           455
4                    1                      0.5           449
5                    1                      1.0            70
6                    1                      0.5           450
```

Here we have extracted the data accompanied only by the protein ID and the cluster ID together with the analysis results with respect to the first marker set. As the the clusters in the `cls2_1 data.frame` are ordered by the cluster ID we can add any information to the data e.g.

```
> data2<-data.frame(data2,main_component1=cls2_1$main_component[data2$cluster])
```

7 Examples

The simplest way to use this package is the wrapper function `analyze.MSfile` (see `?analyze.MSfile`). This function reads and process your the tab-delimited text file with MS data and stores the results in the working directory. Set working directory: In R-studio go to menu: Session, select set working directory, identify the folder where your data are stored. The output files will also appear in this folder. After setting the working directory, it is necessary to provide the path to the tab-delimited file with MSdata. As the first example we use the data from figure 1 in the paper without the first (reference;metadata) columns.

7.1 Example 1

```
> file1<-system.file("extdata","Data_Fig_1a.txt",package="MetaMass")
```

The function MetaMass recognizes two types of data - `MSfile` and `Metadata`. The `MSfile` are the actual data to be analyzed whereas the Metadata stand for internally stored MSdata which can be used as reference (see `?Metadata`).

```
> ##proteins identified by gene-name -> annotation.ID=2 (see ?AnnotationAM)
> ##cluster with respect metadata only (group=0)
> res1<-analyze.MSfile(MSfile=file1,Metadata=c("Christoforou"),output="res1",group=0,cluster.metadata=TRUE)
```

In this case (compare the command to the `usage` section in `?analyze.MSfile`) we use the default annotation, as Metadata we use the Christoforou, Mulvey, Breckels, et al. (2016) dataset and the output will be stored in files starting with the string 'res'. As a toy example (to get the similar result as in figure 1a) we want to cluster only the metadata and align the `MSfile` - do this we set `group=0` which says that none of the data in MSfile should be clustered and we have to specify that want cluster the metadata `cluster.metadata=TRUE`).

7.2 Example 2

As the second example let us reconstruct the figure 1b in the paper. Here we want to cluster the MSfile (which is default and we need not specify the `group` parameter) and align the metadata to it.

```
> file2<-system.file("extdata","Data_Fig_1b.txt",package="MetaMass")
> ##cluster with respect MSfile only (cluster.metadata=FALSE by default)
> res2<-analyze.MSfile(MSfile=file2,Metadata=c("Christoforou"),output="res2")
```

The results are stored in the working directory in files starting with 'res2'.

7.3 Example 3

See Analysis option 5 As the third example we show how to use multiple files as input.

```
> ##compare multiple files component fractionation with Metadata
> files1<-system.file("extdata",c("Bileck.txt","Thakar.txt","Carvalho.txt"),package="MetaMass")
> res3<-analyze.MSfile(MSfile=files1,Metadata=c("Christoforou"),output="res3")
```

See figure 1

7.4 Example 4

As the last example let us show how to use multiple annotations (one can use custom annotation file if it corresponds the appropriate format). Let us use all but the 3rd annotation i.e. columns 3,4,6,7. And as another example of option we use 480 cluster.

```
> res4<-analyze.MSfile(MSfile=file2,Metadata=c("Christoforou"),output="res2_4annot",clusters=480,markers=c(3,4,6,7))
```

User can plot the precision-recall curves directly by (figure 2)

```
> par(mfrow=c(3,3),mar=c(1, 4, 2.2, 1) + 0.1,cex=0.45)
> plot.prAM(res4) #plot in  3 rows and 3 columns
```

And the heatmaps as seen in the Java TreeView application are shown in the figure 3

7.5 Example 5

By default only proteins which were detected in all studies (groups) and Metadata are used for clustering. This could be bypassed by setting the parameter `overlap=`k - in this case only the proteins which were detected in at least k studies are used for clustering. The other protein would reamain in the table and heatmap as unclassified.

```
> files2<-system.file("extdata",c("Bileck.txt","Thakar.txt","Carvalho.txt","Andreyev.txt","Rodriguez.txt"),package="MetaMass")
> res3<-analyze.MSfile(MSfile=files2,Metadata=c("Christoforou"),output="res3intersect")
```

Here only 1280 proteins present in all studies are exported.

```
> files2<-system.file("extdata",c("Bileck.txt","Thakar.txt","Carvalho.txt","Andreyev.txt","Rodriguez.txt"),package="MetaMass")
> res3<-analyze.MSfile(MSfile=files2,Metadata=c("Christoforou"),output="res3_5",overlap=3)
```

Here 8548 proteins present in at least 3 studies are analyzed.

```
> files2<-system.file("extdata",c("Bileck.txt","Thakar.txt","Carvalho.txt","Andreyev.txt","Rodriguez.txt"),package="MetaMass")
> res3<-analyze.MSfile(MSfile=files2,Metadata=c("Christoforou"),output="res3_6",overlap=5)
```

An finally 3315 proteins detected in at least 5 studies.

8 Number of clusters

It is obvious that the classification recall depends strongly on the number of clusters - the recall grows as the average cluster size decreases (with 100% agreement with the annotation when each cluster contains just 1 protein) however the number of unclassified proteins grows as well. To asses this fact we analyzed the data in examples 1-3 with different number of clusters (starting with average cluster size around 2000 - the exact value depends on the size of dataset - and decreasing the size to 2) - figure 4 The solid red line (example 1) shows almost perfect reconstruction of classification with as much as 100 proteins per cluster (left panel) and since in this example only annotated proteins were used no unclassified proteins. The results for the data in examples 2 and 3 show consistently no loss of resolution for cluster size above 25.

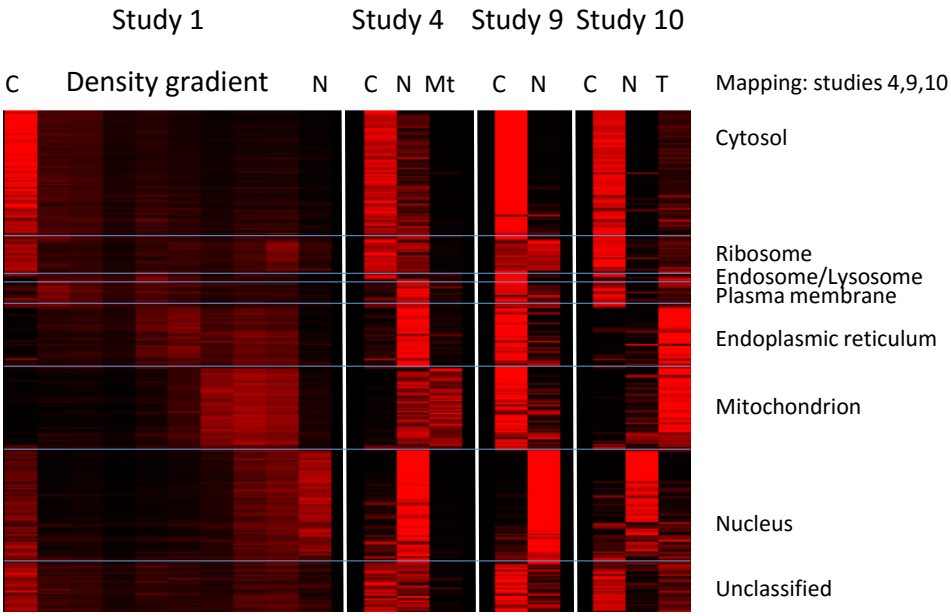


Figure 1: Heatmnap as seen in JavaTreeView

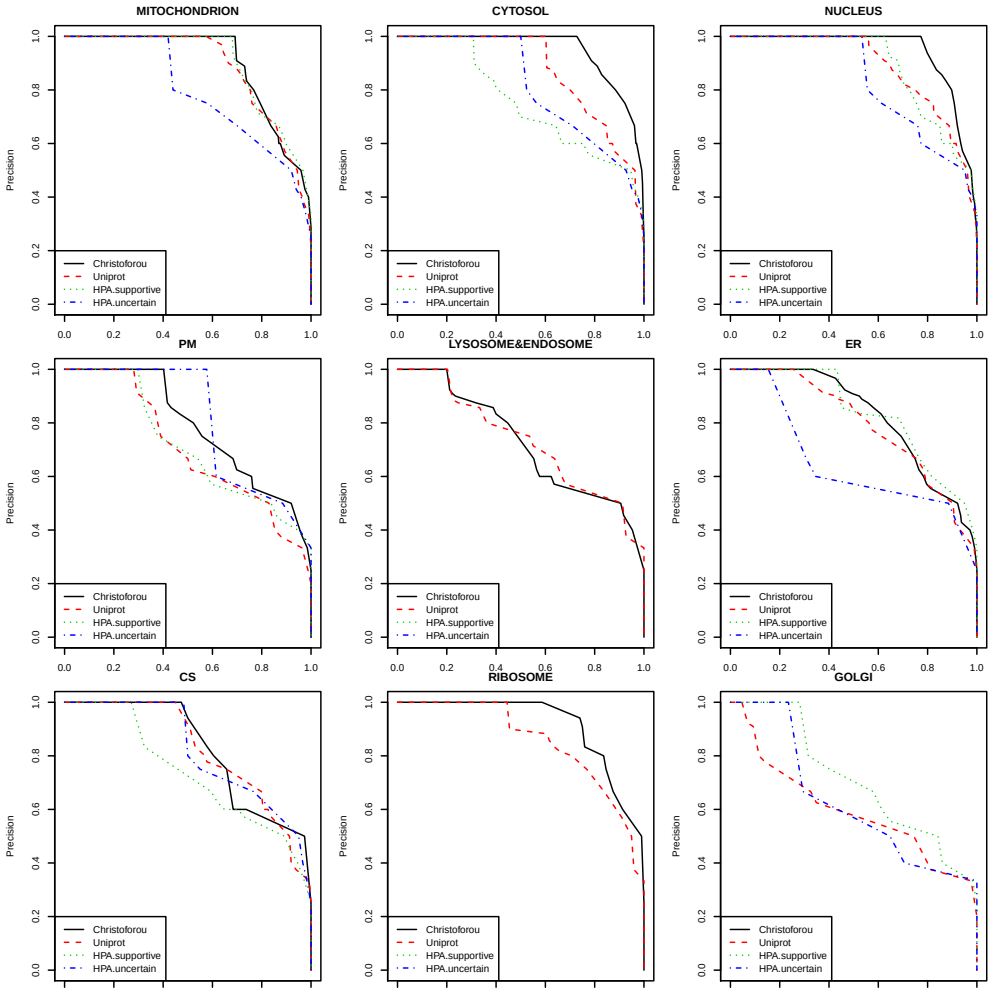
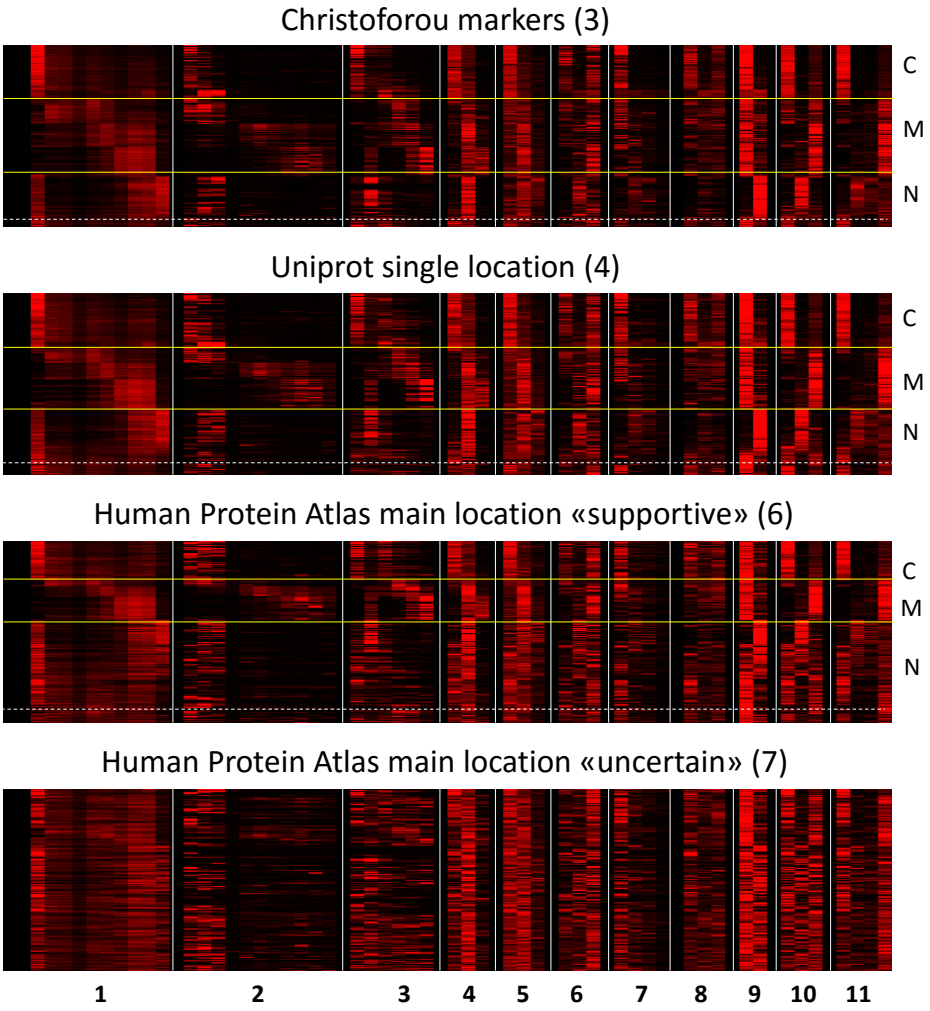


Figure 2: Precision-recall curves



The heatmaps show results obtained when datasets from studies 2-11 were classified on basis of indicated marker sets. Main subcellular components: C: cytoplasm, M: plasma membrane and cytoplasmic organelles, N: Nucleus. Proteins below the white dashed lines were not classifed. Study 1 (*Christoforou et al.*) represents the metadata. This dataset was not used for the analysis, but aligned afterwards (default option in MetaMS). The numbers in parenthesis represent the corresponding options for the markers = argument in MetaMS. Results obtained with the Uniprot annotations are similar to those obtained with the Christoforou markers, except that some nuclear proteins are found in the cytoplasmic fractions. The «supportive» annotations from the Human Protein Atlas (HPA) classify a larger number of proteins as nuclear, and a large number of these are recovered in cytoplasmic fractions. The HPA «uncertain» annotations yield a heatmap with no obvious pattern. This demonstrates that these annotations have a very poor fit with the MS data and the other marker sets.

Figure 3: heatmaps as seen in JavaTree view

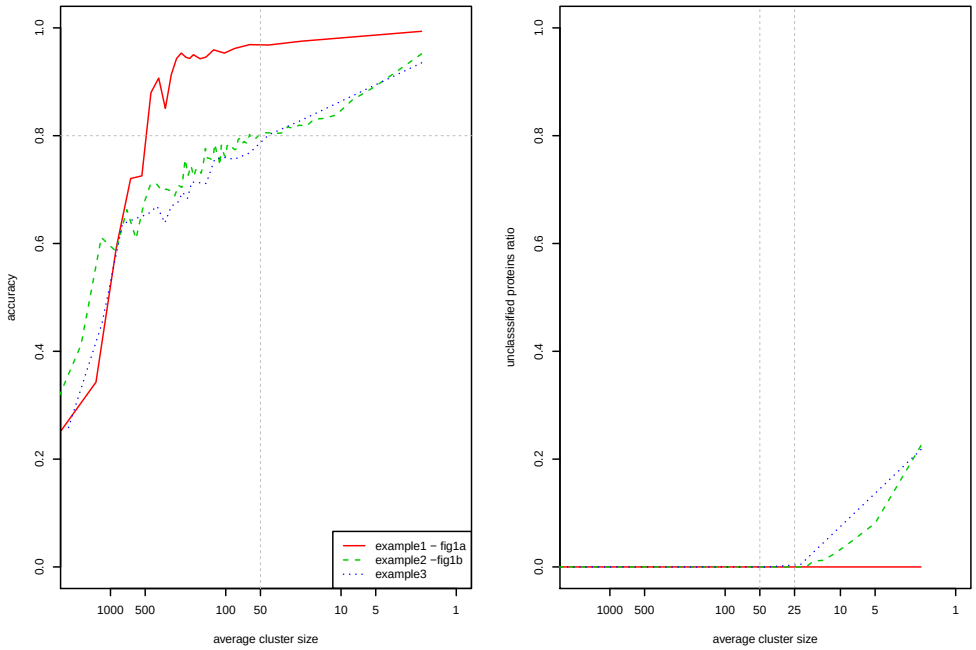


Figure 4: Performance against average cluster size