Overlap_markers_analysis

KimPankyung

2022-08-16

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#create markers file
#Bing the packages
library(data.table)
library(magrittr)
library(dplyr)
```

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

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

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

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

```
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
#Load the marker file
cortex_sc_marker<-readRDS('D:/pankyung/intern_back_up/sub/follow_SPOTlight_vignette/markers_sc.
RDS')
GSE118020_marker<-readRDS('D:/pankyung/intern_back_up/sub/GSE118020_RAW/markers_sc_a.rds')

#View columns in marker file
cn<-colnames(cortex_sc_marker)
cn
```

```
## [1] "p_val"      "avg_log2FC" "pct.1"      "pct.2"      "p_val_adj"
## [6] "cluster"    "gene"
```

```
cn1<-colnames(GSE118020_marker)
cn1
```

```
## [1] "p_val"      "avg_log2FC" "pct.1"      "pct.2"      "p_val_adj"
## [6] "cluster"    "gene"
```

```
#Extract avg_log2FC value and cluster of each gene from reference file
gca<-c("gene","cluster","avg_log2FC")
cortex_sc_marker <- cortex_sc_marker[,gca]

#Extract avg_log2FC value and cluster of each gene from GSE118020 file
GSE118020_marker <- GSE118020_marker[,gca]

#Top markers of reference file
tmc<-c(10) #In this case, top10
tmn<-paste0("cortex_sc_marker_top",tmc) #Set the number to save data easily
#Extract the top markers only and create data frame
tme <- as.data.frame(cortex_sc_marker %>% group_by(cluster) %>% top_n(n = tmc, wt = avg_log2F
C))

#Extract the top markers only and create data frame
tmg<-c(20)
tmeg <- as.data.frame(GSE118020_marker %>% group_by(cluster) %>% top_n(n = tmg, wt = avg_log2F
C))

#View the overlap markers
lc<-levels(cortex_sc_marker[,2]) #Extract the type of sample
lg<-levels(GSE118020_marker[,2]) #Extract the type of sample

#Create the dictionary contains all markers of each cell type
lstc<-list() #Empty list
for (i in lc){ #For loop in cell type names
  scc<-cortex_sc_marker[cortex_sc_marker$cluster==i,] #Select cell type based on cluster column
  sccg<-c(scc[,1]) #Extract the gene column
  dfc<-data.frame(sccg) #Create dataframe to add in list
  names(dfc)[1]<-i #Change the name of dataframe
  lstc<-append(dfc,lstc) #Keep adding the dataframe during the for loop
}

#Create the dictionary contains all markers of each cluster
lstg<-list() #Empty list
for (i1 in lg){ #For loop in cluster types
  scg<-GSE118020_marker[GSE118020_marker$cluster==i1,] #Select the cluster based on cluster col
umn
  scgg<-c(scg[,1]) #Extract the gene column
  dfg<-data.frame(scgg) #Create dataframe to add in list
  names(dfg)[1]<-i1 #Change the name of dataframe
  lstg<-append(dfg,lstg) #Keep adding the dataframe during the for loop
}

#Create the dictionary contains top markers of each cluster
lstgt<-list() #Empty list
for (i2 in lg){ #For loop in cluster types
  scgt<-tmeg[tmeg$cluster==i2,] #Select the cluster based on cluster column
  scgtg<-c(scgt[,1]) #Extract top rank genes
  dfgt<-data.frame(scgtg) #Create dataframe to add in list
  names(dfgt)[1]<-i2 #Change the name of dataframe
  lstgt<-append(dfgt,lstgt) #Keep adding the dataframe during the for loop
}

#Generate the overlaped marker number dataframe
```

```
lstw<-lstgt #Select the list you want to compare with reference file
for (i3 in lg){ #For loop in cluster types
  #print(paste0("###########",i3)) #Use it to check for loop is working well
  ovns<-c() #Empty vertor for overlap number list
  for (i4 in lc){ #For loop in cell type names
    #print(paste0("=======",i4)) #Use it to check for loop is working well
    ovn<-c(0) #Empty vector for count overlap number
    for (i5 in lstw[[i3]]){ #For loop in list you want to compare
      #print(zz) #Use it to check for loop is working well
      if (is.na(match(i5,lstc[[i4]]))){#print("") #Use is.na() to set condition of not matched
        ovn<-ovn #If it doesn't match, then do not count
        }
      else {#print("hi") #When it match, print "hi"
        ovn<-ovn+1 #When it match, then count it
        }
    }
    #print(ovn) #Use it to check for loop is working well
    ovns<-c(ovn,ovns) #Make overlap count list to create dataframe later
    #print(ovns) #Use it to check for loop is working well
    assign(i3,rev(ovns)) #For loop is applied in opposite order, so use reve()
    #print(length(ovns)) #Use it to check for loop is working well

  }
}

#Prepare the overlap dataframe for plot
ref_type<-c(lc) #Set column name for cell types in overlap dataframe
dfov<-data.frame(ref_type) #Create overlap dataframe
#Complete the overlap dataframe
bc<-c(2) #For loop will add one column after each loop and there is ref_type column already
for (i6 in lg){ #For loop in cluster types
  #print(i) #Use it to check for loop is working well
  dfov$i6<-get(i6) #Use get() to bring the object, and add it to dataframe
  names(dfov)[bc]<-i6 #Change the name to keep adding the column
  bc<-bc+1 #Columns will be added, so column should be increased too
}

#Customize the dataframe for plot
dfovn<-dfov[,2:15] #Extract numeric column for t()
tdfov<-data.frame(t(dfovn)) #Use t() to change the axis
setnames(tdfov,lc) #t() will generate new column names, so change it
tdfov$cluster<-lg

#Assess the dataframe
nzc<-c() #Empty vector for non zero columns
zc<-c() #Empty vector for zero columns
for (i7 in lc){ #For loop in cell type names
  if (all(tdfov[i7]==0)==FALSE){ #If at least one value is not 0 in specific column
    nzc<-c(i7,nzc) #Add in none zero columns
  } else { #If all values is o in specific column
    zc<-c(i7,zc)} #Add in zero columns
}
nzc #Check none zero columns
```

```
##  [1] "VLMC"       "Vip"        "Sst"        "Sncg"       "SMC"
##  [6] "Serpinf1"   "Pvalb"      "Peri"       "Oligo"      "NP"
## [11] "Meis2"      "Macrophage" "Lamp5"      "L6b"        "L6 IT"
## [16] "L6 CT"      "L5 PT"      "L5 IT"      "L4"         "L2/3 IT"
## [21] "Endo"       "CR"         "Astro"
```

```
length(nzc) #Check none zero columns number
```

```
## [1] 23
```

```
zc #Check zero columns
```

```
## NULL
```

```
length(zc) #Check zero columns number
```

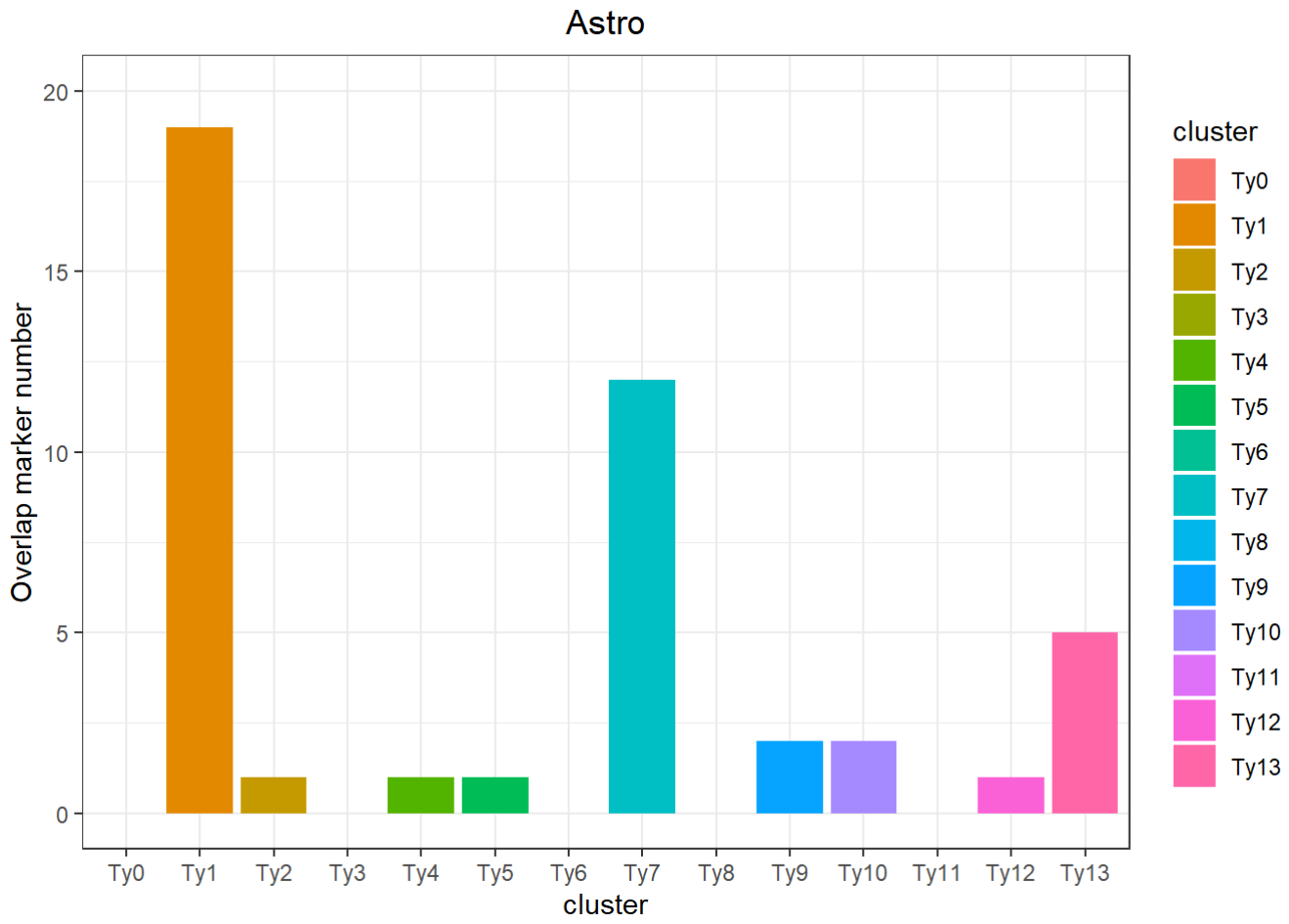
```
## [1] 0
```

```
#Check the current directory
getwd()
```

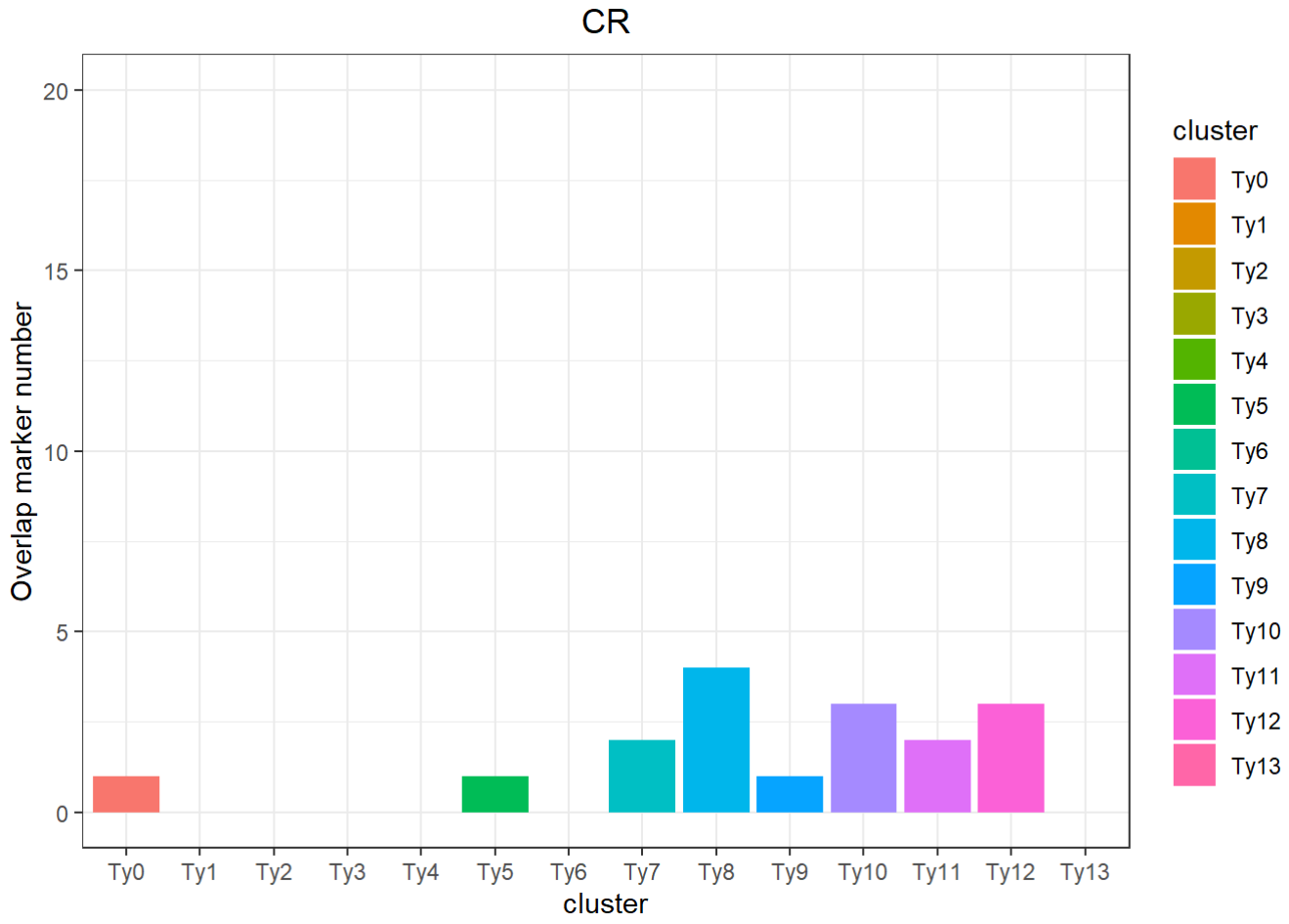
```
## [1] "D:/pankyung/intern_back_up/sub/marker_analysis"
```

```
#Draw marker plots
for (i8 in lc){ #For loop in cell type names
  pl<-print(ggplot(data=tdfov, aes(x=cluster, y=get(i8), fill=cluster))+ #Fill is for legend
    geom_bar(stat = "identity")+ #Stat is the format of barplot
    scale_x_discrete(limits=lg)+ #Use scale_x_discrete to reorder x values
    scale_y_continuous(limits=c(0,tmg))+ #Use scale_y_continuous to set range fo y values
    scale_fill_discrete(limits=lg)+ ##Use scale_fill_discrete to reorder legend values
    ggtitle(i8)+ #Create plot title
    ylab("Overlap marker number")+ #Create the y title
    theme_bw()+ #Set the background color as white
    theme(plot.title = element_text(hjust = 0.5))) #Situate the title in center
  nsi<-gsub("/","_",i8) #Use gsub() to convert / to _
  fn<-paste0("D:/pankyung/intern_back_up/sub/marker_analysis/plot/",nsi,".png")
  ggsave(filename = fn,plot = pl, device = "png")
  Sys.sleep(1) #Show the plot one by one
}
```

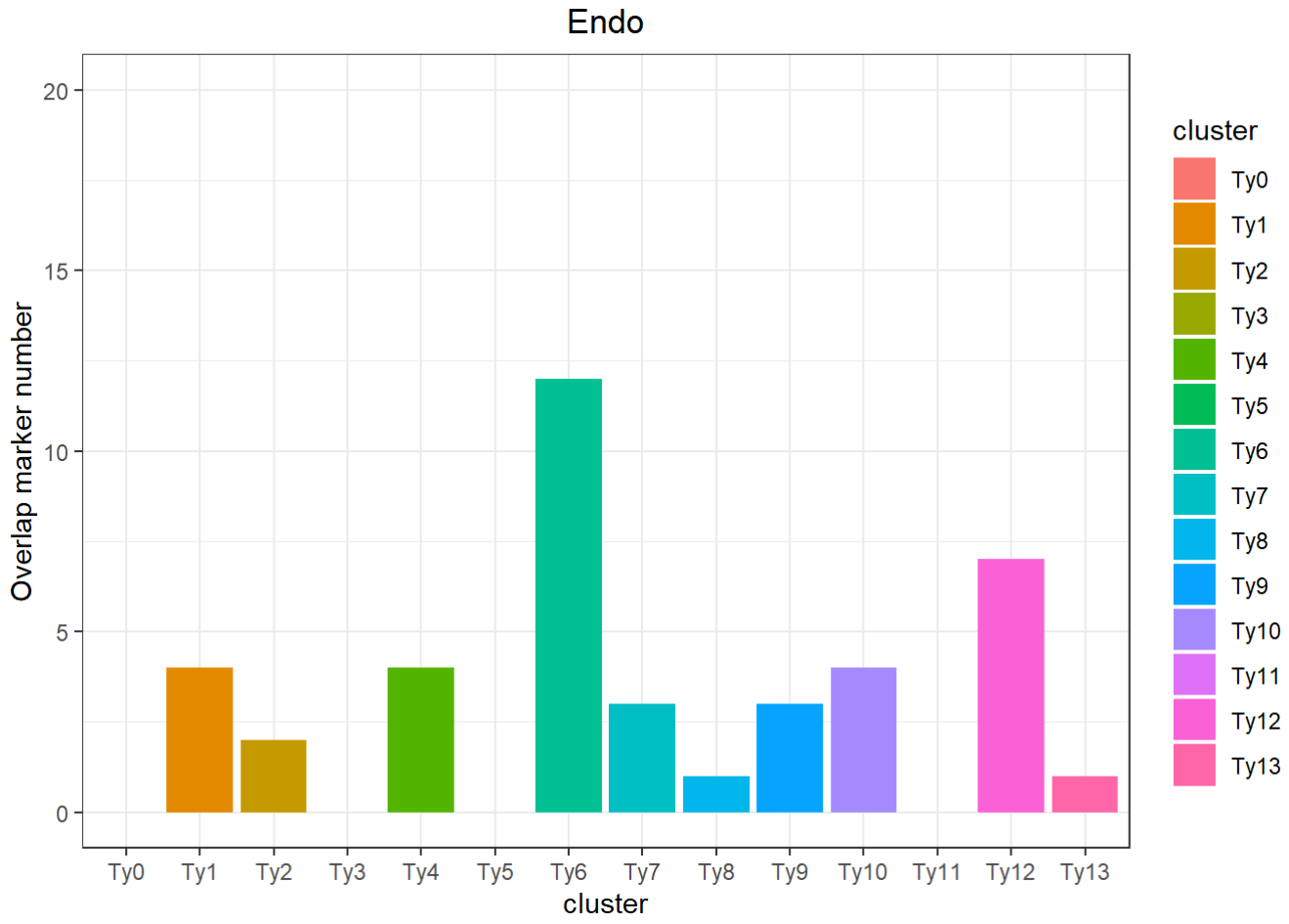
```
## Saving 7 x 5 in image
```



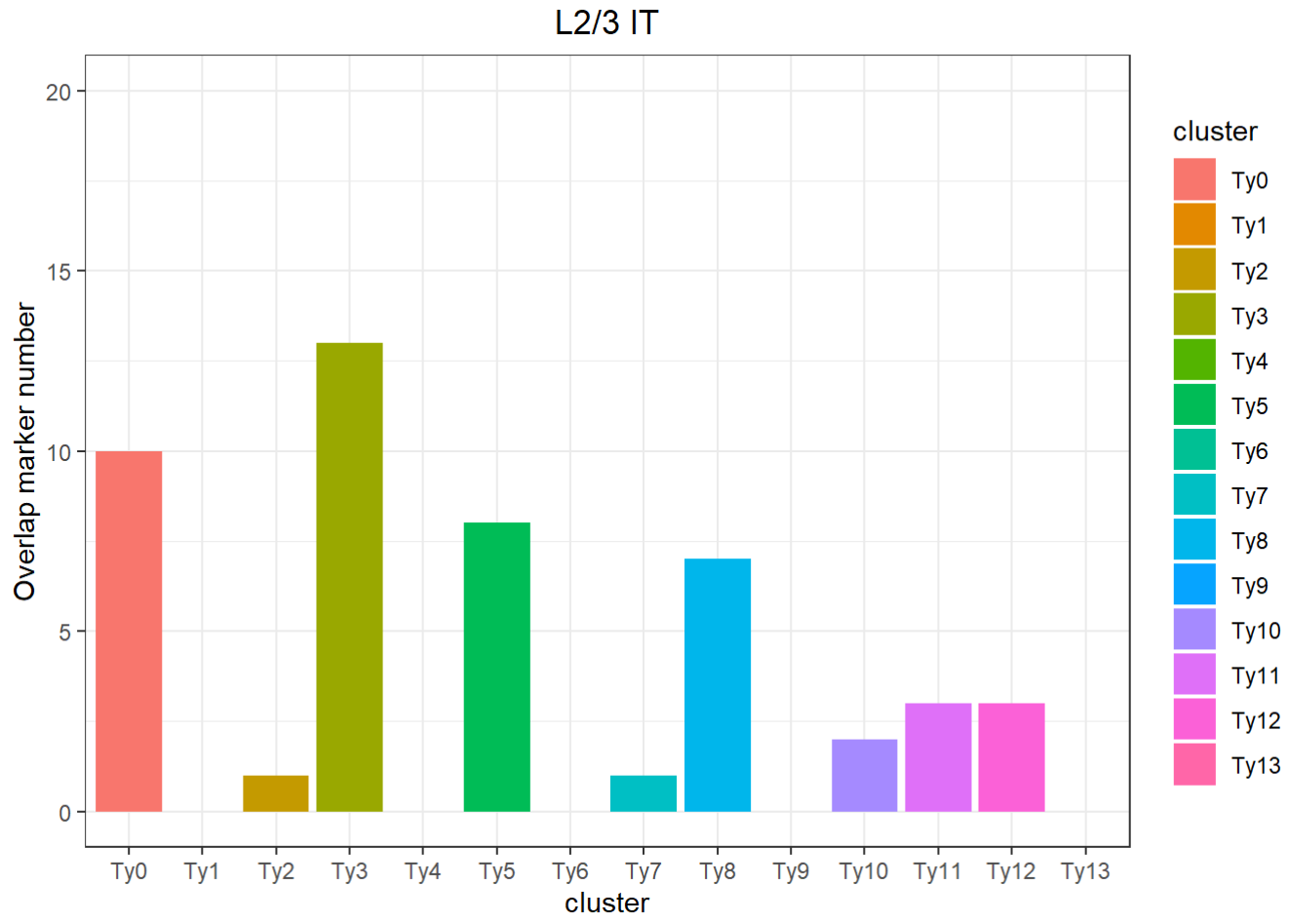

```
## Saving 7 x 5 in image
```



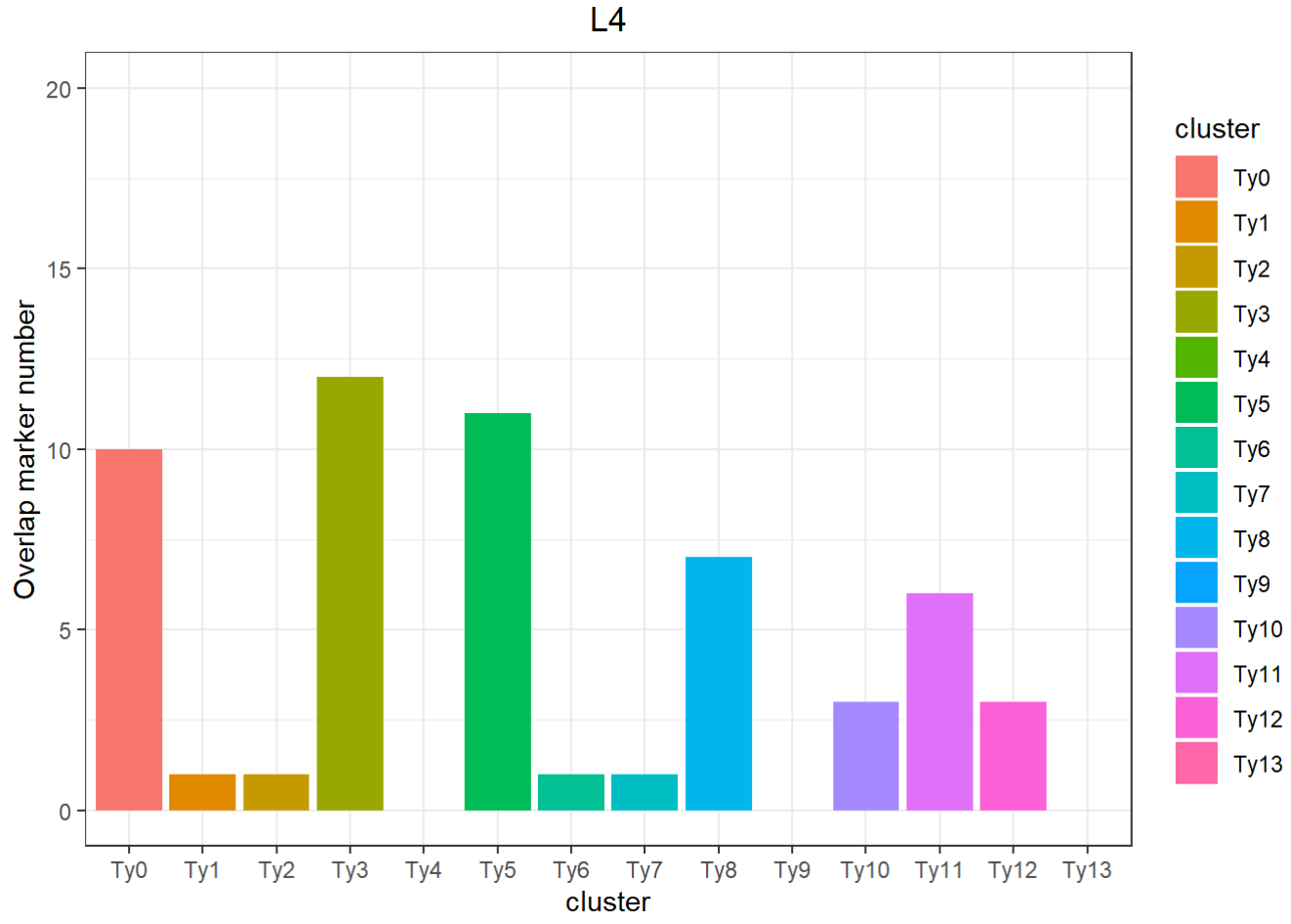
```
## Saving 7 x 5 in image
```



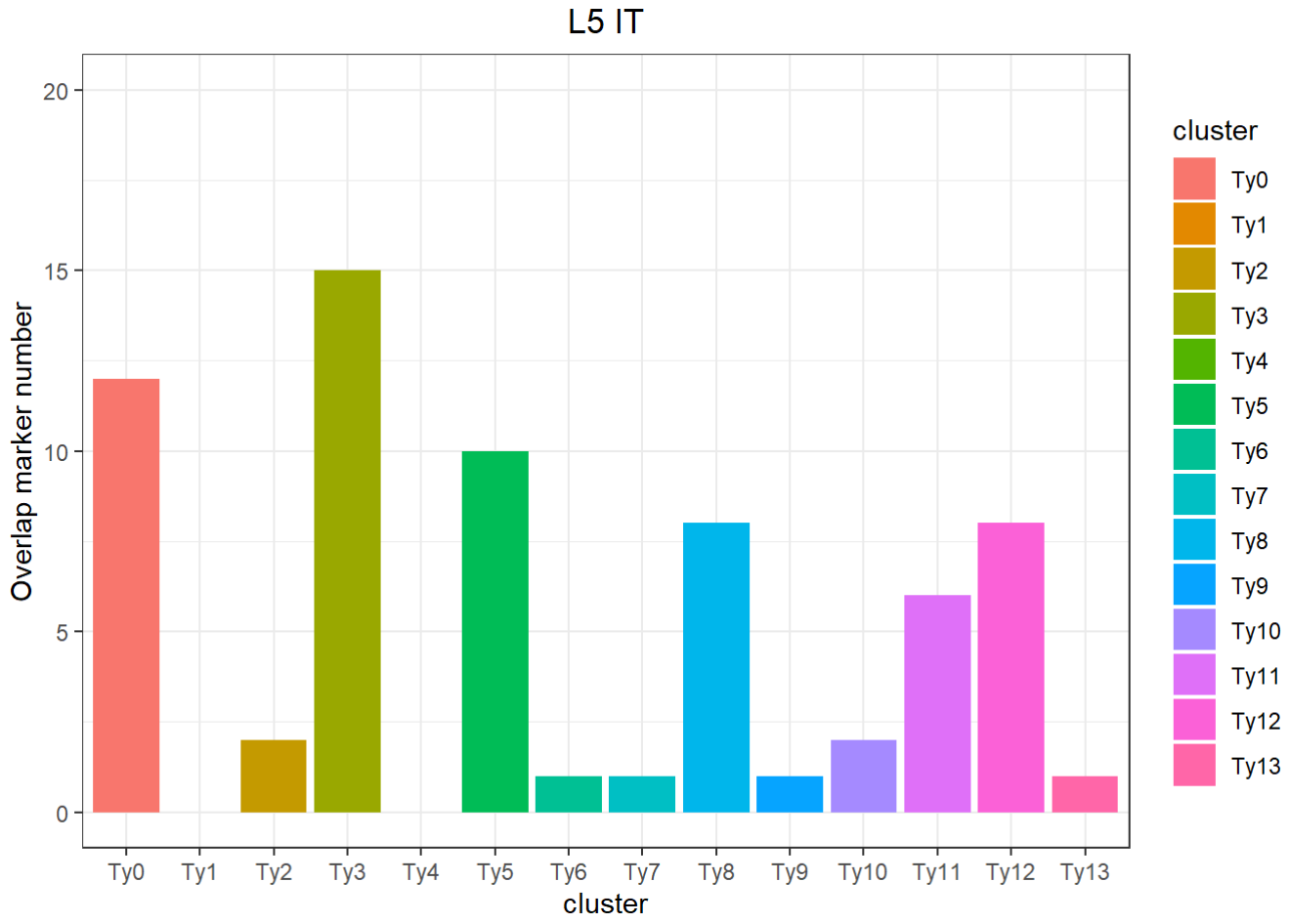

```
## Saving 7 x 5 in image
```



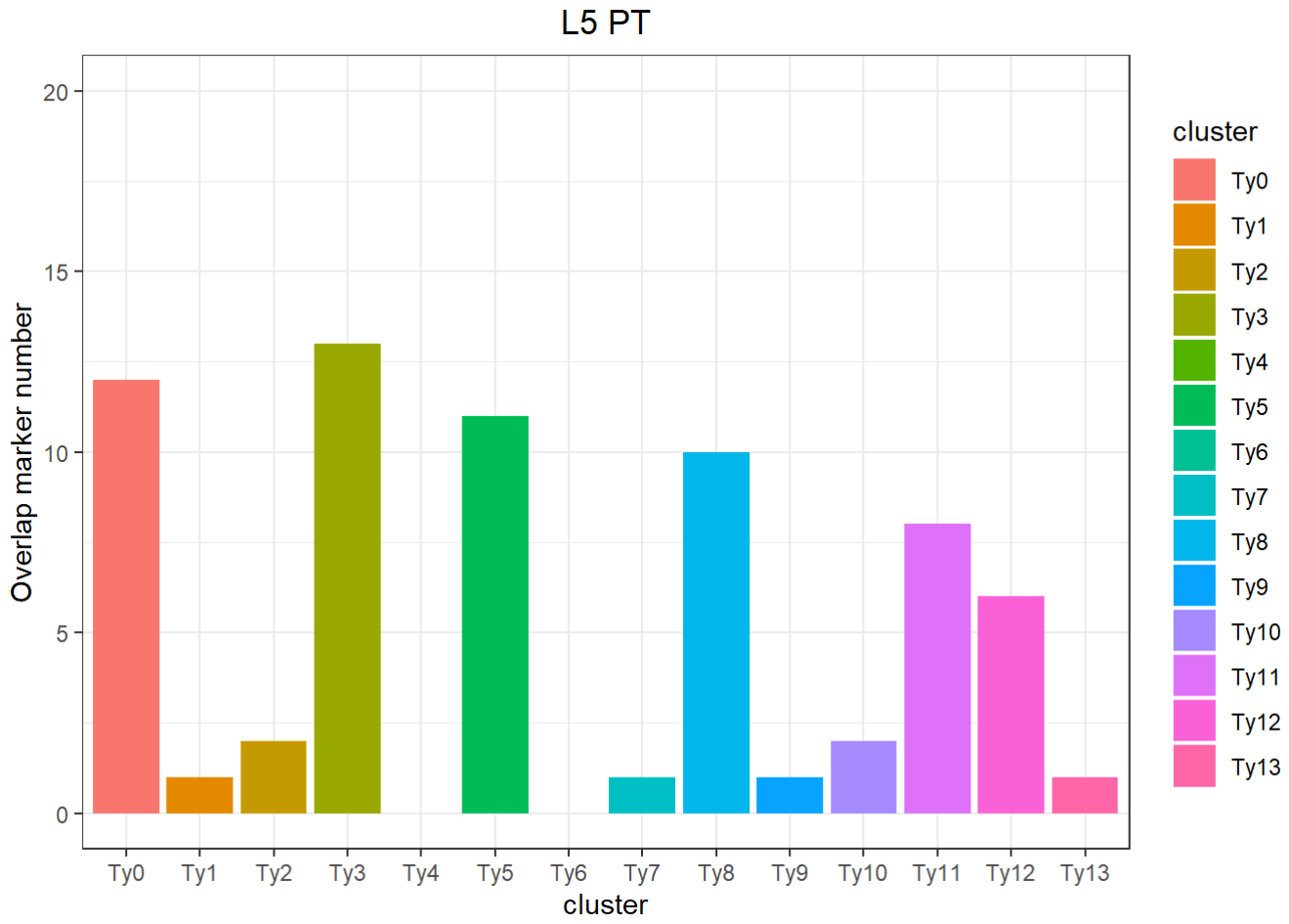

```
## Saving 7 x 5 in image
```



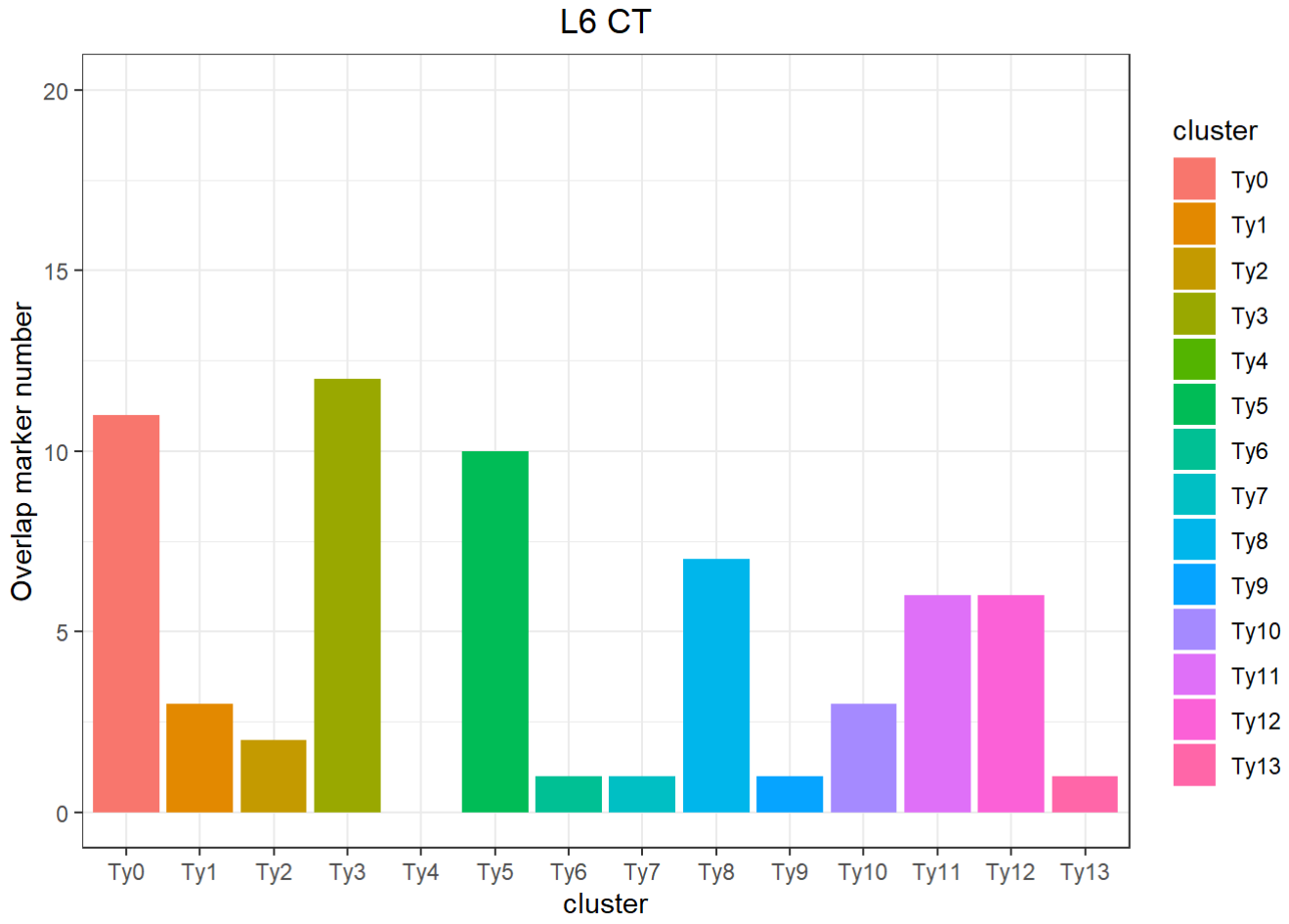
```
## Saving 7 x 5 in image
```



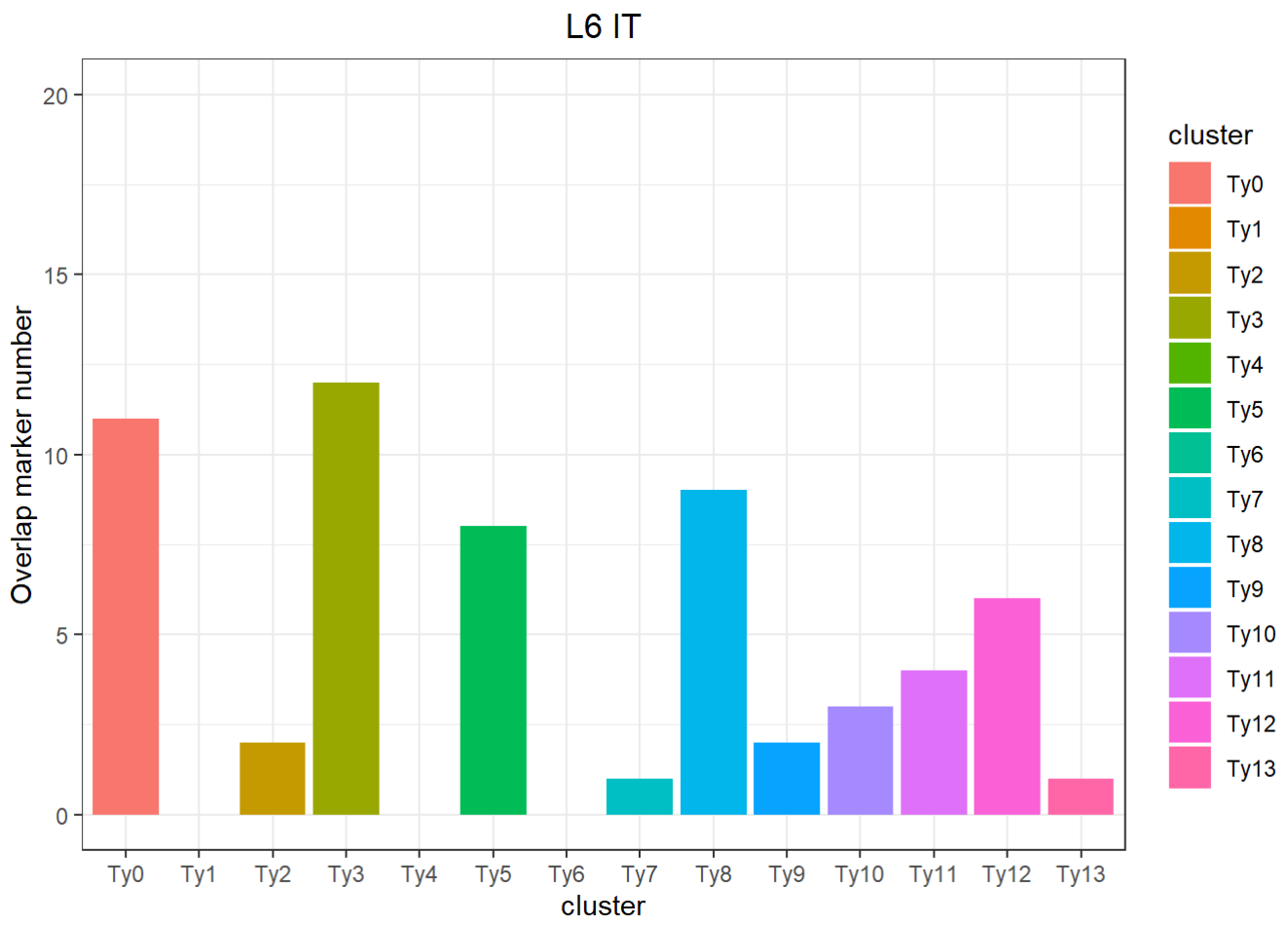

```
## Saving 7 x 5 in image
```



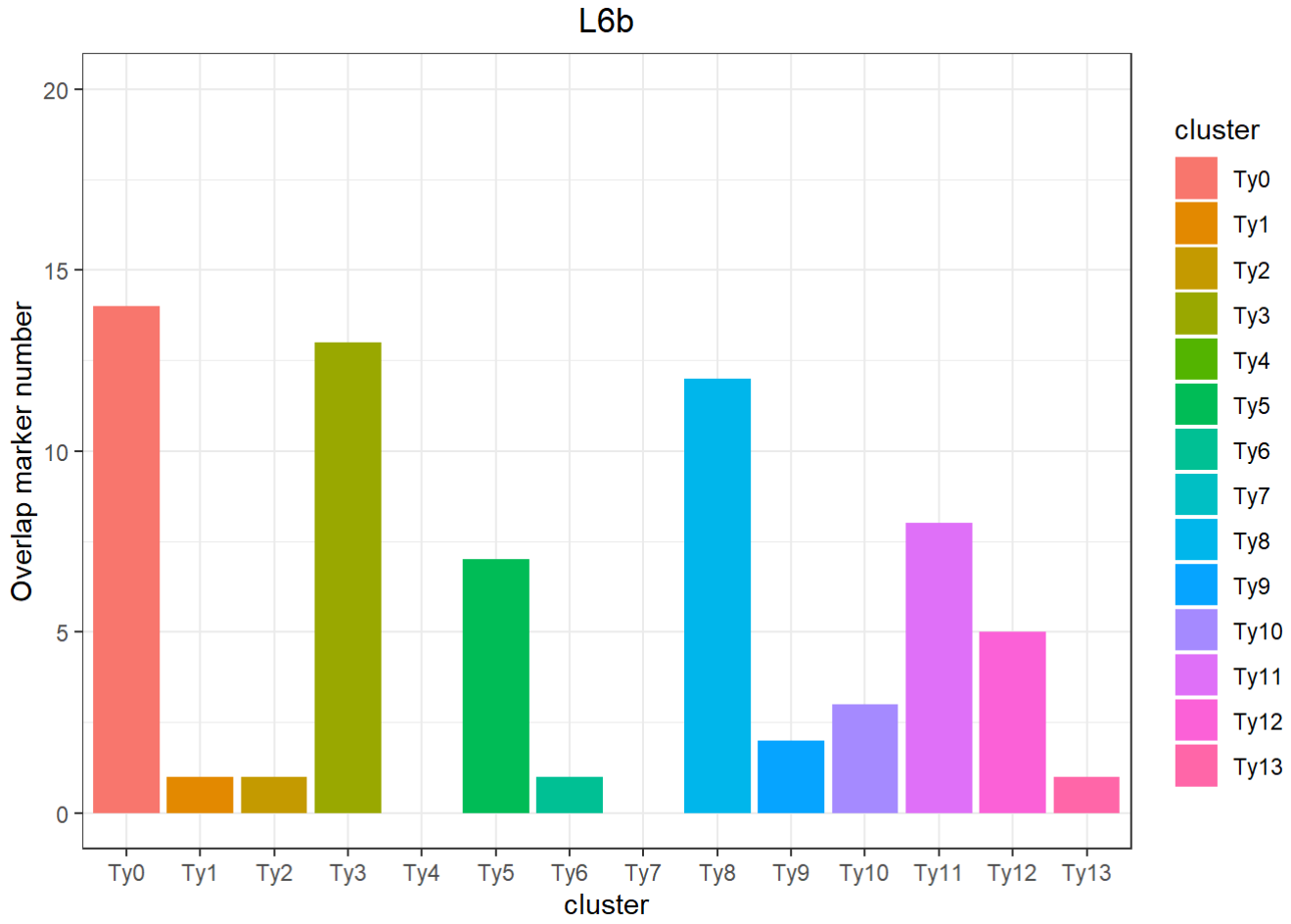

```
## Saving 7 x 5 in image
```



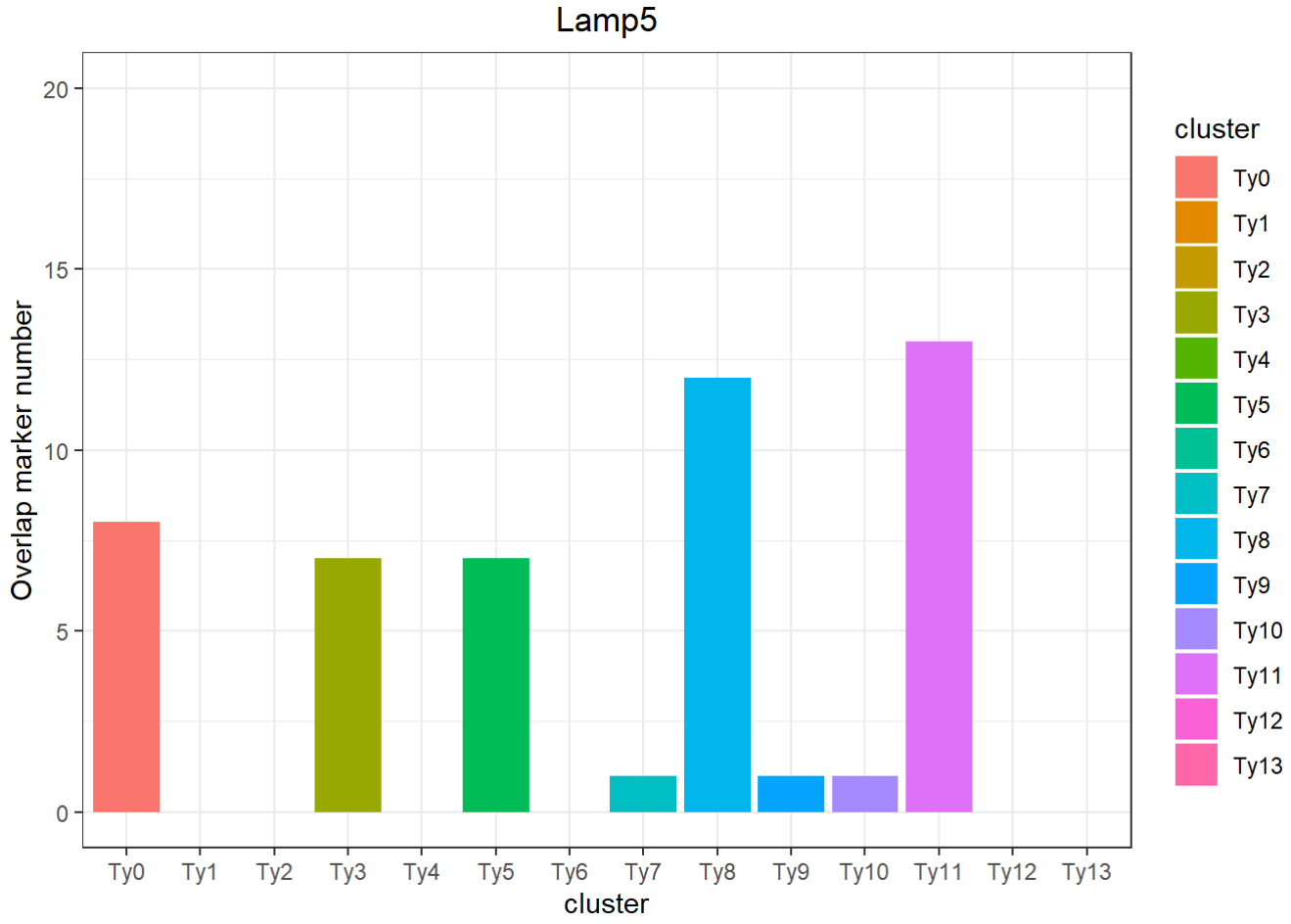
```
## Saving 7 x 5 in image
```



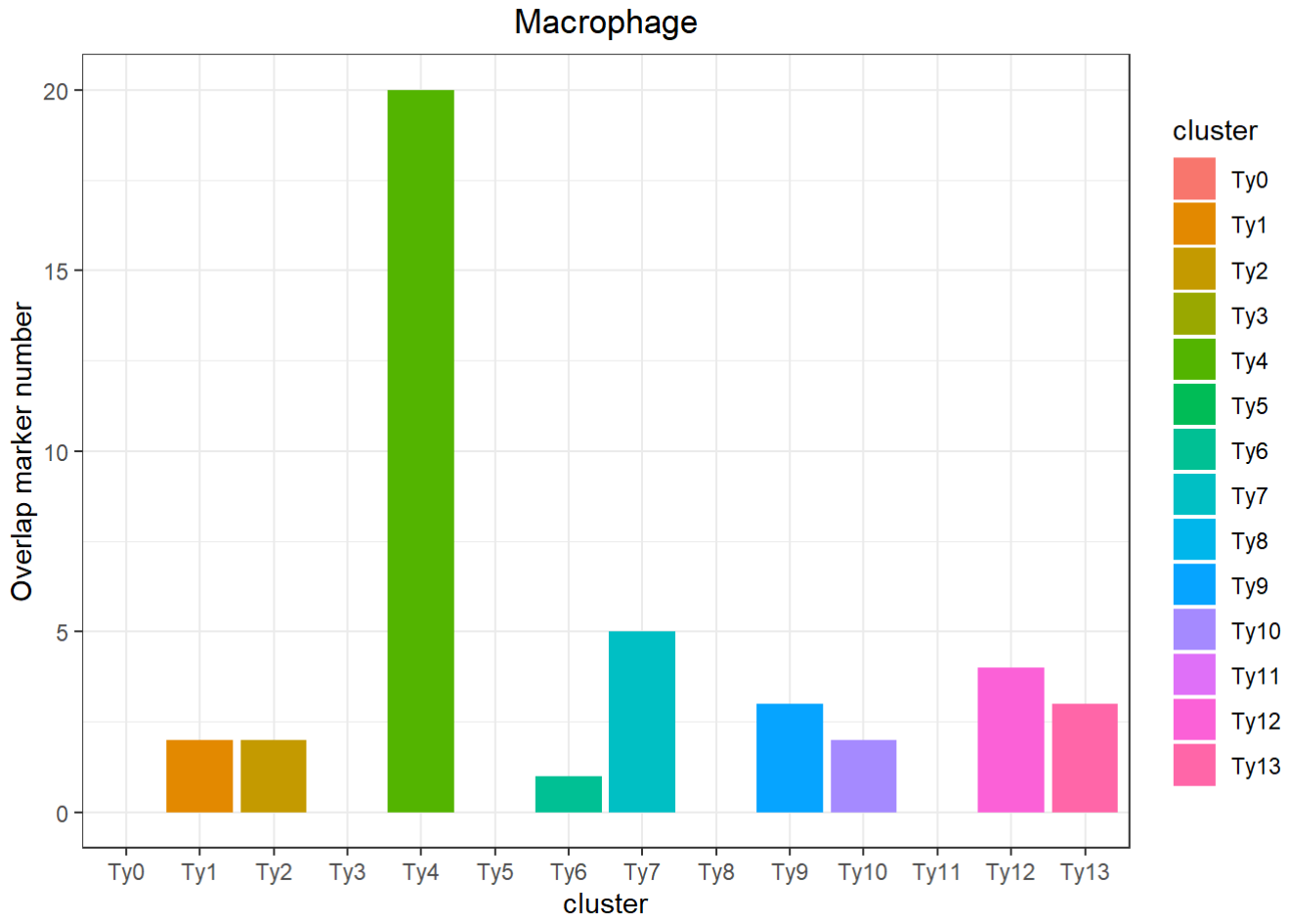

```
## Saving 7 x 5 in image
```



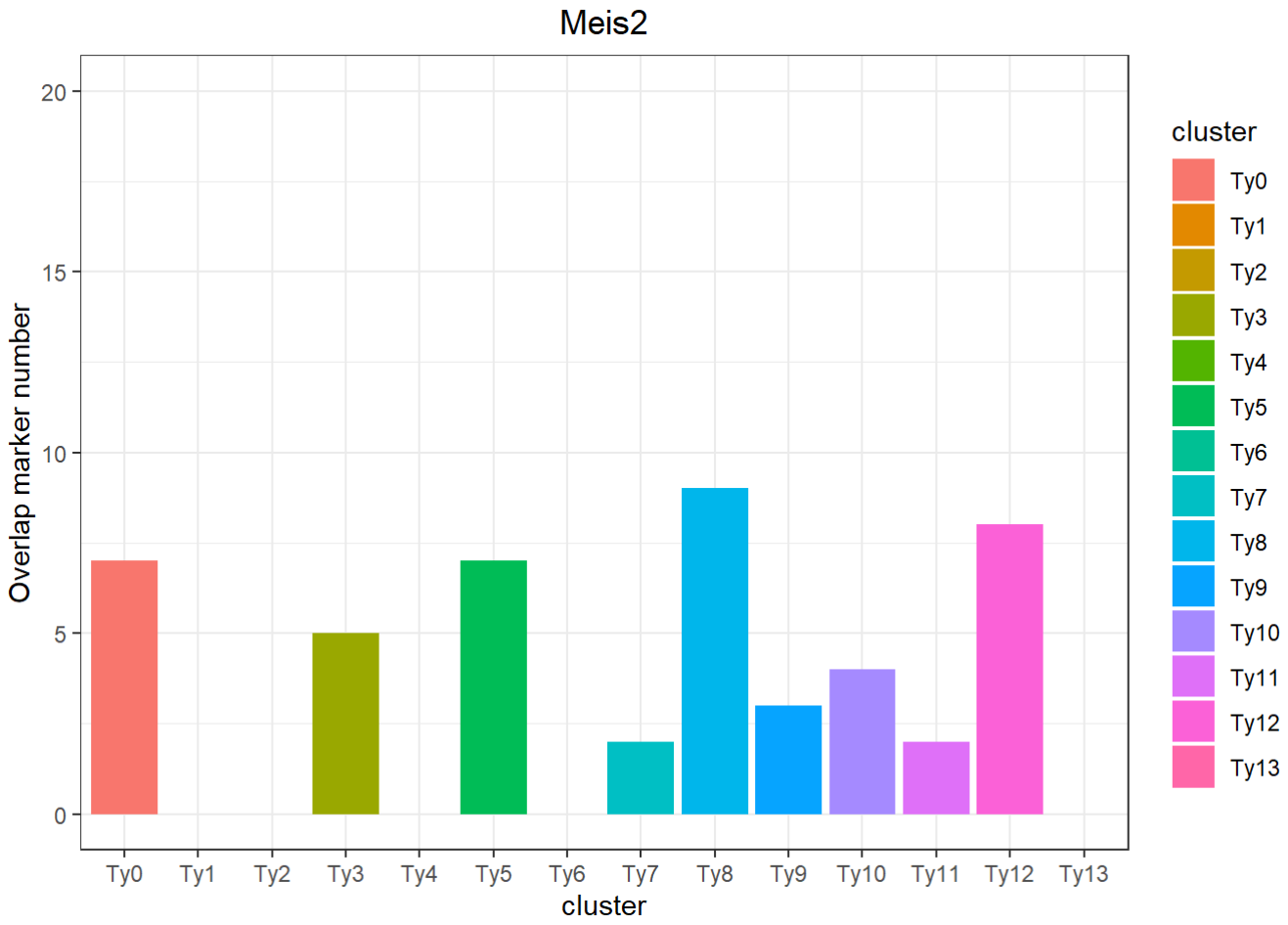

```
## Saving 7 x 5 in image
```



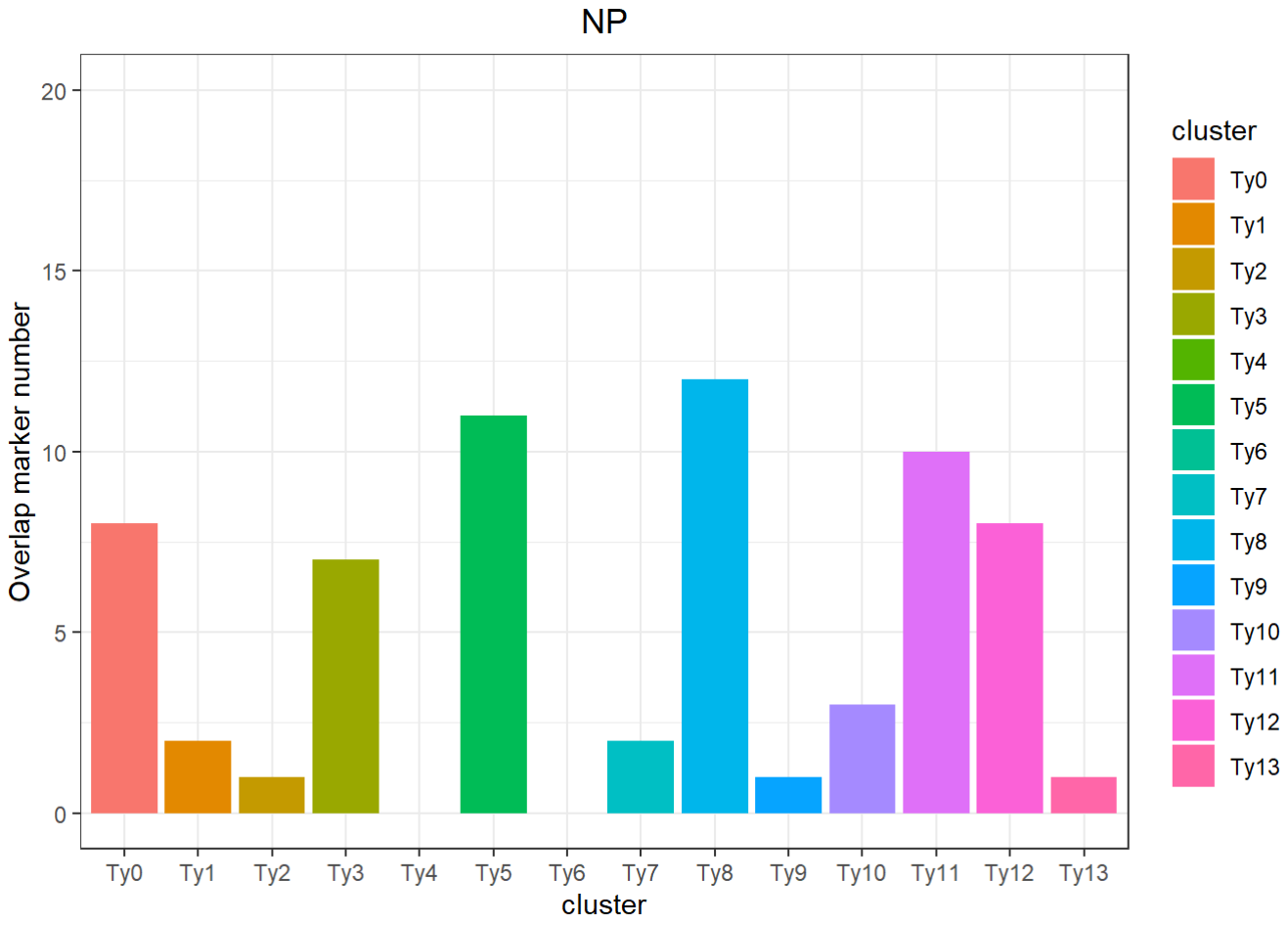
```
## Saving 7 x 5 in image
```



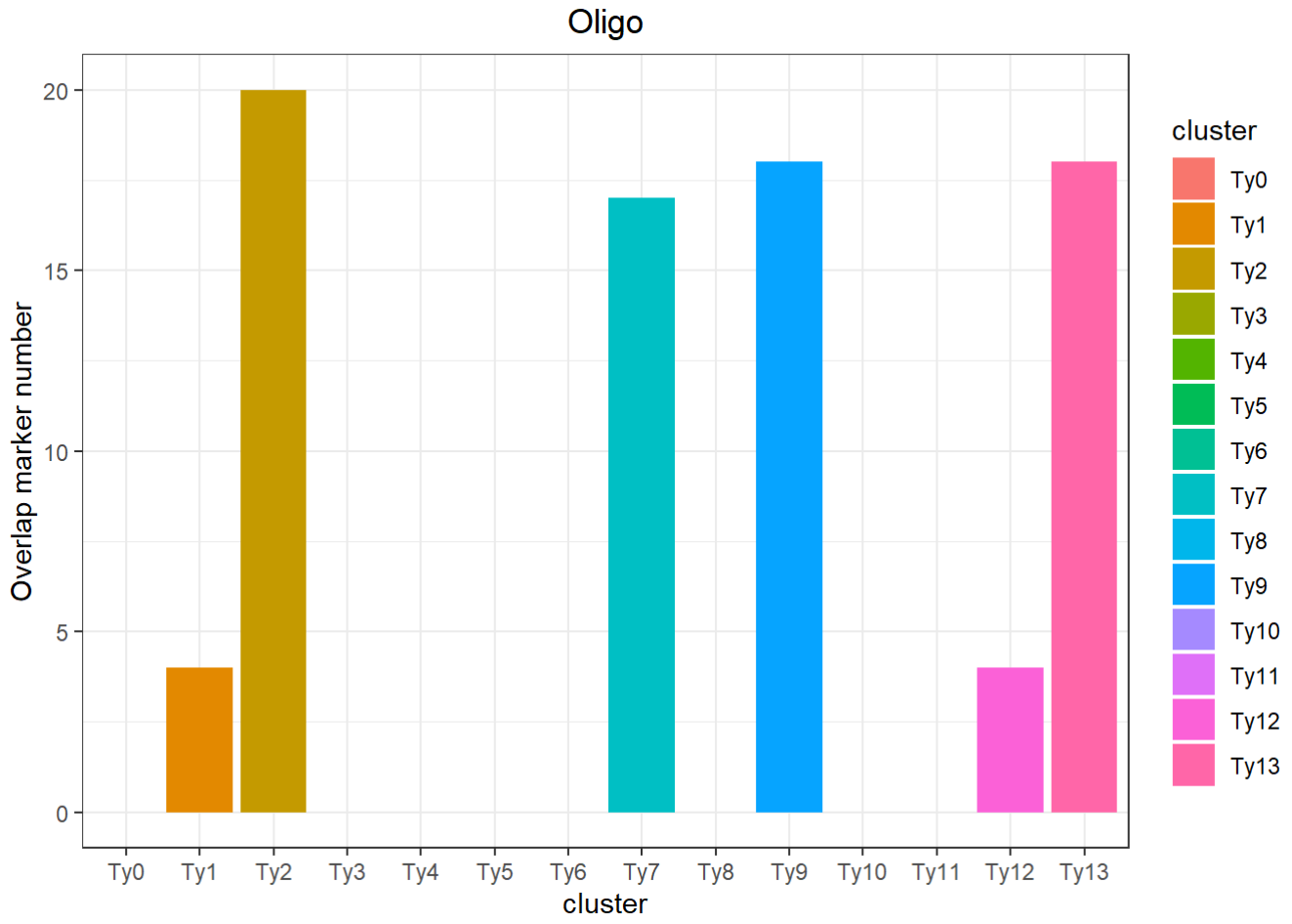

```
## Saving 7 x 5 in image
```



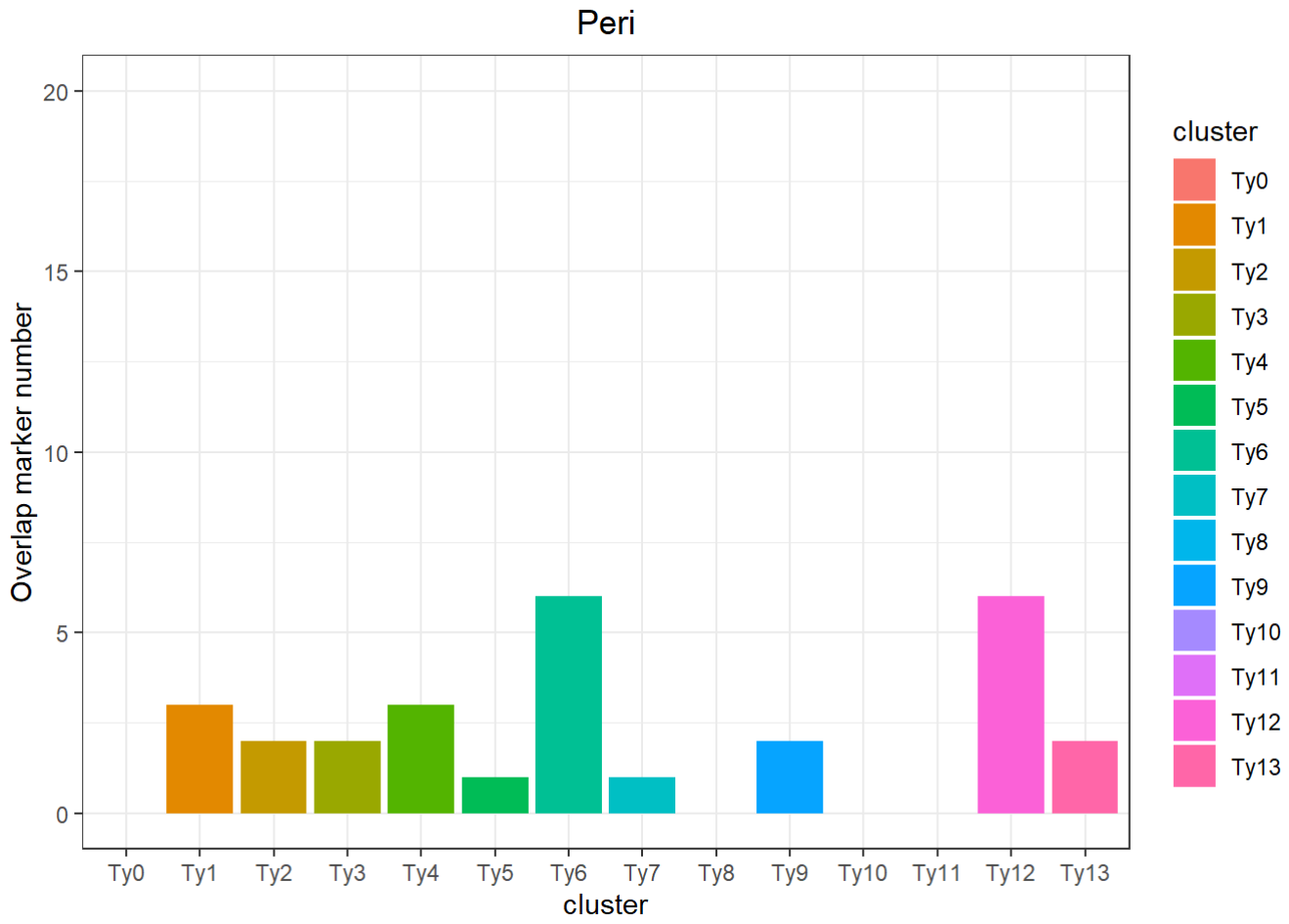

```
## Saving 7 x 5 in image
```



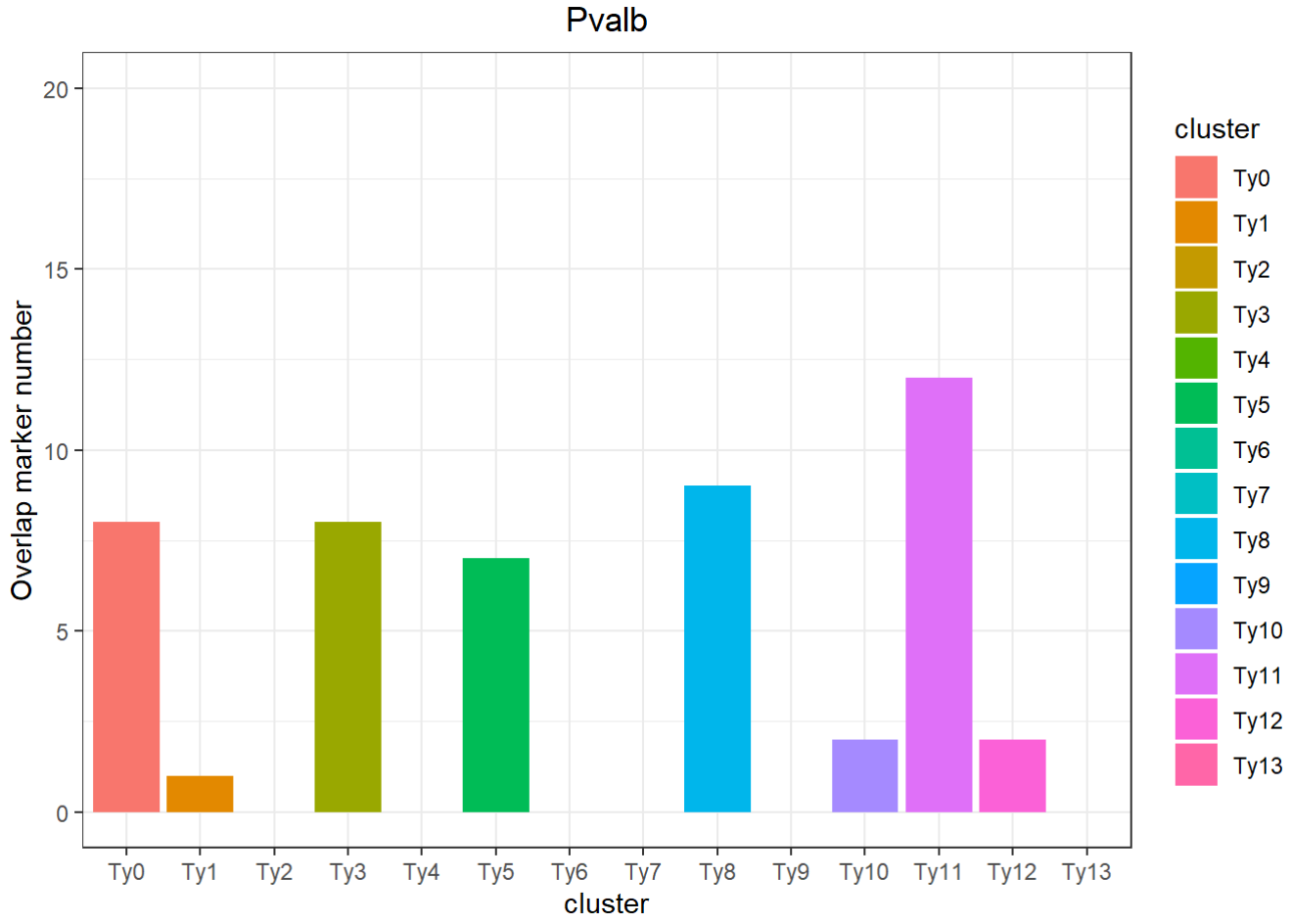
```
## Saving 7 x 5 in image
```



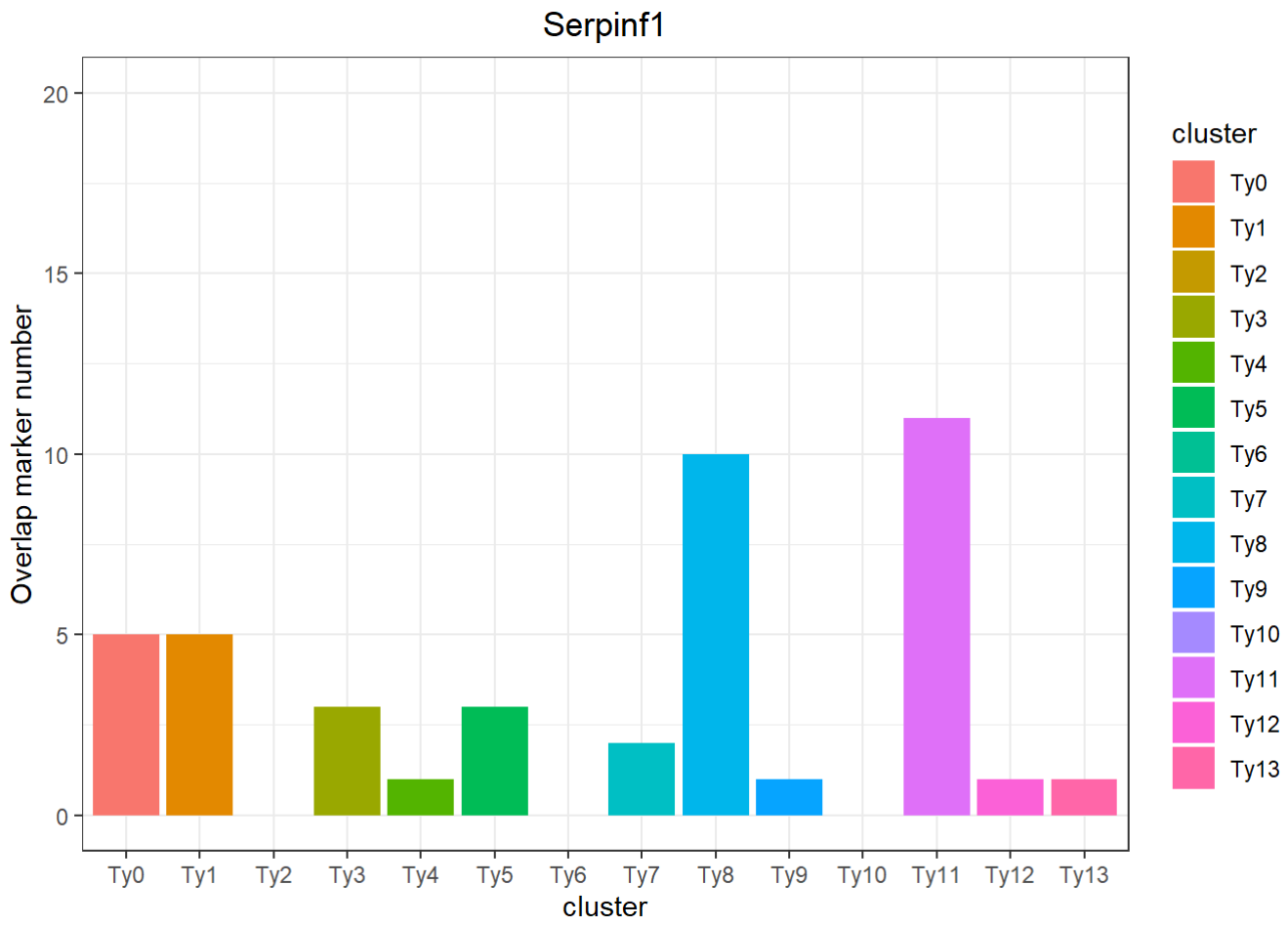

```
## Saving 7 x 5 in image
```



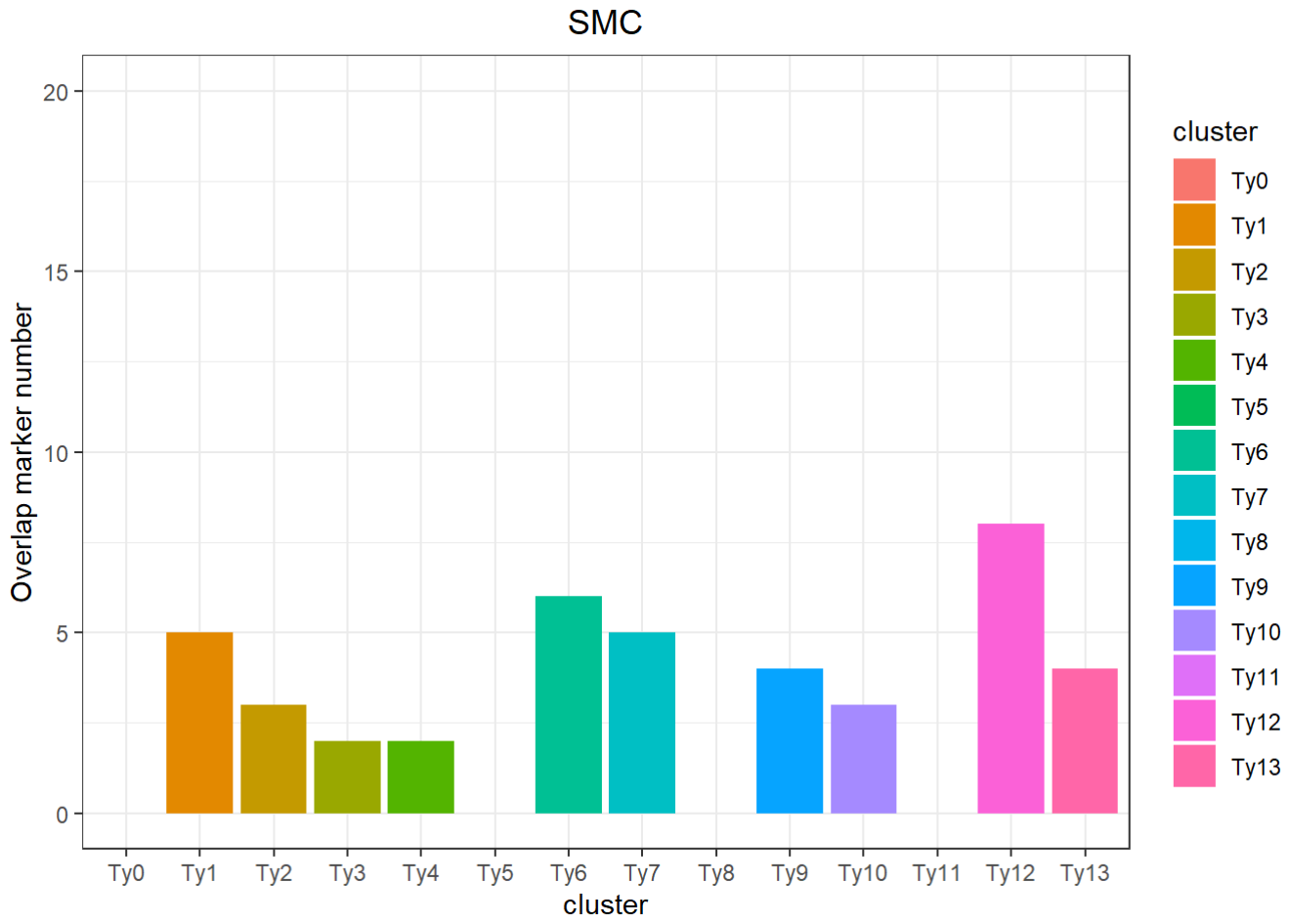

```
## Saving 7 x 5 in image
```



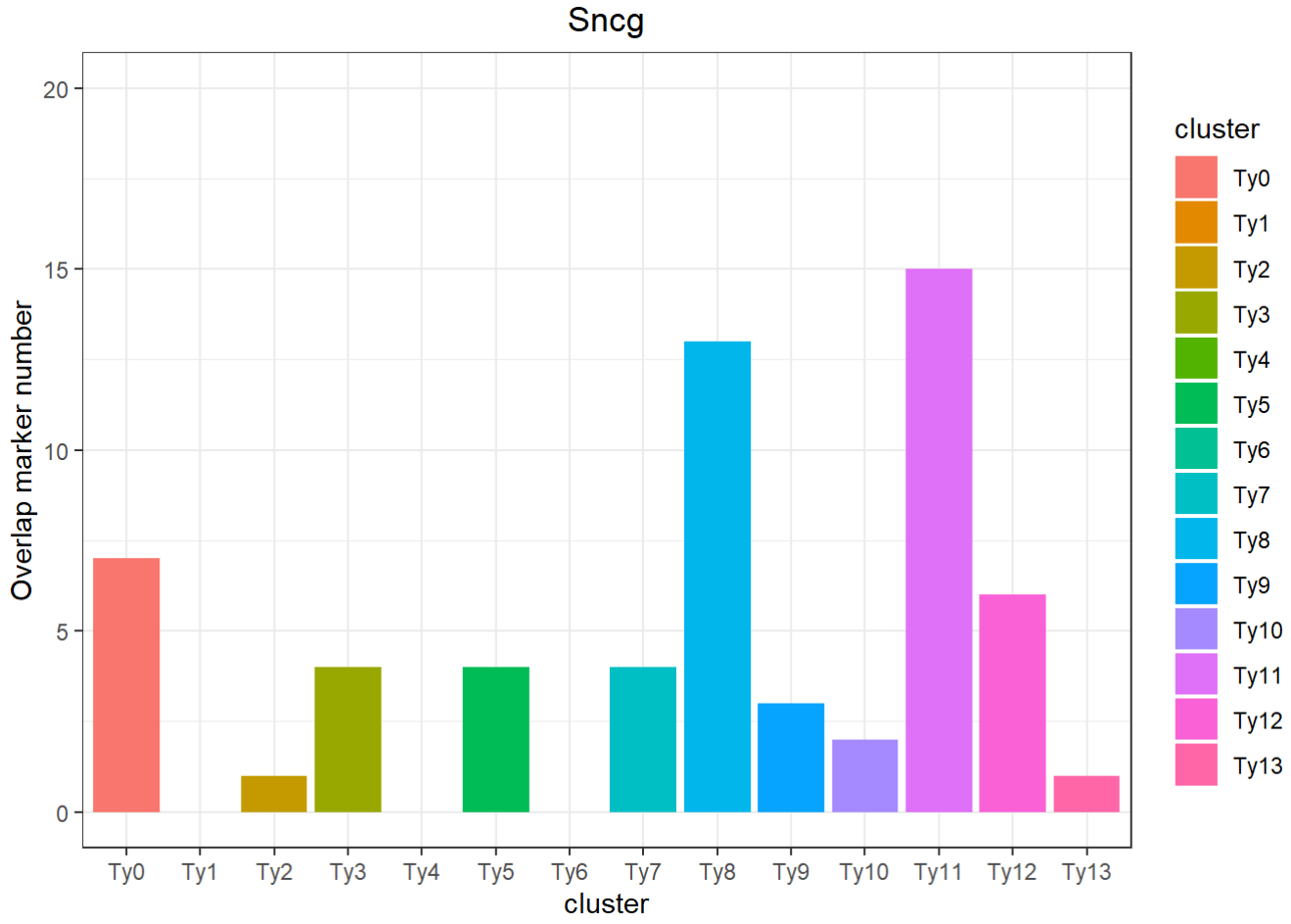
```
## Saving 7 x 5 in image
```



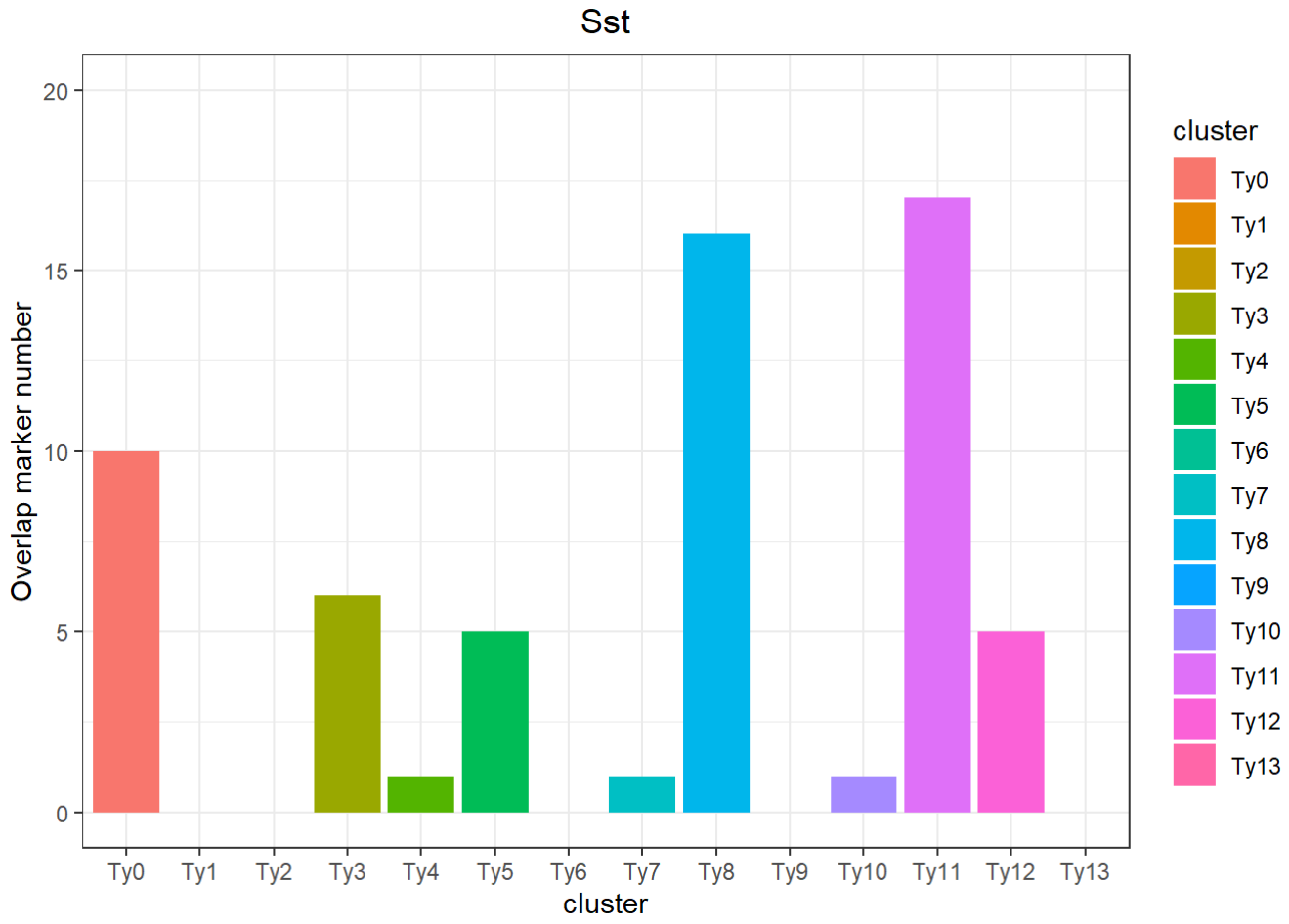

```
## Saving 7 x 5 in image
```



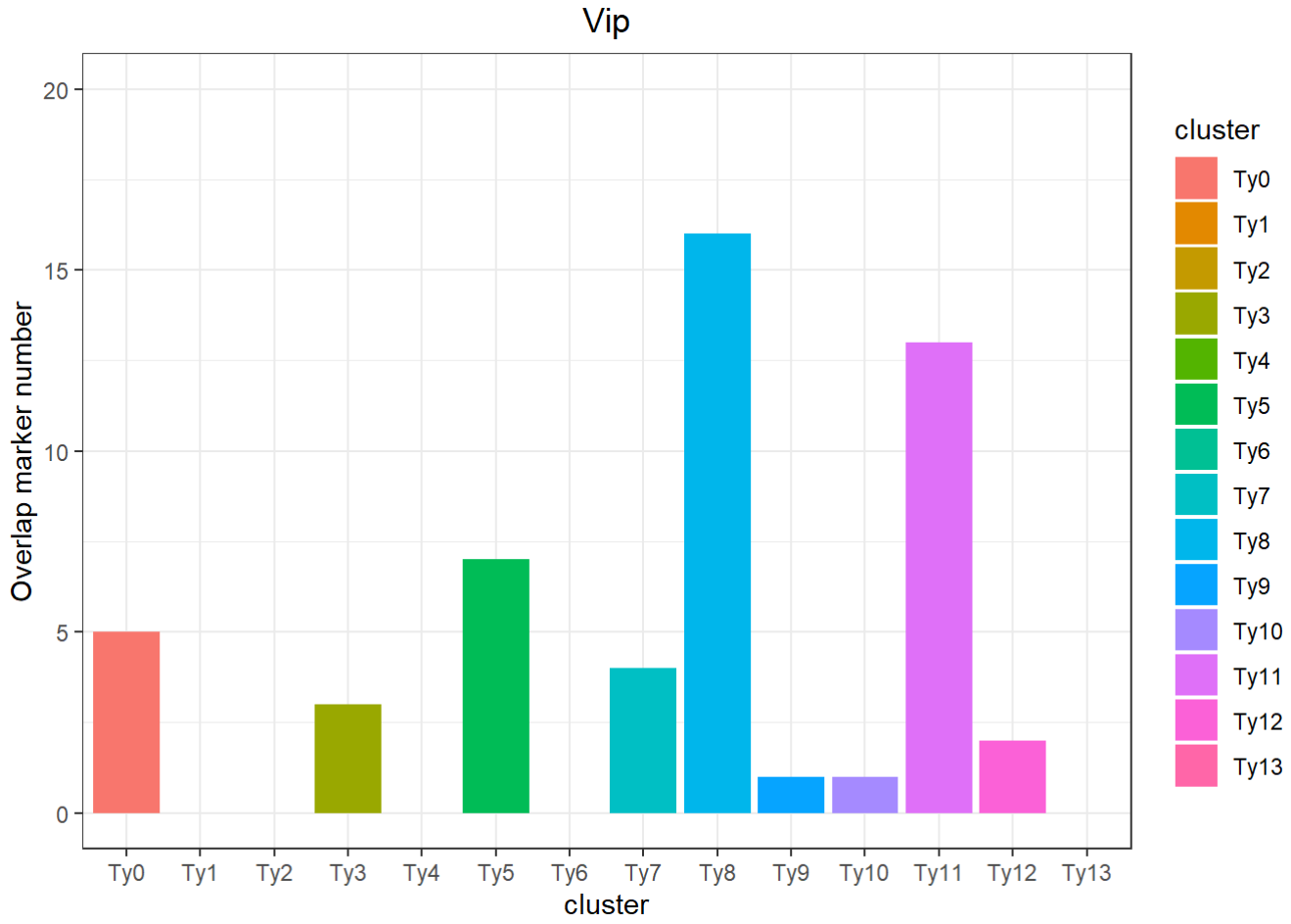

```
## Saving 7 x 5 in image
```



```
## Saving 7 x 5 in image
```




```
## Saving 7 x 5 in image
```




```
## Saving 7 x 5 in image
```