Example2: MAE

Getting started:

```
## load libraries
library(cBioPortalData)
library(httr)
library(dplyr)
library(stringr)
library(ggplot2)
library(MultiAssayExperiment)
library(S4Vectors)
library(UpSetR)
```

Brief explanation what an MAE is:

The essence of a MultiAssayExperiment: * The essence of MultiAssayExperiment is the following: Imagine you have a bookshelf;
* The whole bookshelf is one particular study; then image the levels within the bookshelf * every levels in this bookshelf is one particular assay or experiment (e.g. CopyNumber Calls)
* Those Assays/Experiments are stored as lists within the MultiAssayExperiment Object. *you can combine different experiments on one sample/study and commonly work with them*
experiments may be measures of mRNA, copyNumber Alterations, and mutations calls (among others)

Example:

We concentrate on Lung Invasisve Adenocarcinomas (LUAD) from TCGA We start by creating a MAE object with luad_tcga data from cBIO

We now work with **'luad_tcga'** and retrieve all the information needed to create MultiAssayExperiment

```
## list all Assays/Experiments for this particular study:
Assays_available = molecularProfiles(api = cbio,
                                     studyId = 'luad_tcga',
                                     projection = 'SUMMARY')
Assay_Ids = Assays_available$molecularProfileId # in this case we have 15 Assays available
Assay_Ids # print
##  [1] "luad_tcga_rppa"
##  [2] "luad_tcga_rppa_Zscores"
##  [3] "luad_tcga_gistic"
##  [4] "luad_tcga_mrna"
##  [5] "luad_tcga_mrna_median_Zscores"
##  [6] "luad_tcga_rna_seq_v2_mrna"
##  [7] "luad_tcga_rna_seq_v2_mrna_median_Zscores"
##  [8] "luad_tcga_linear_CNA"
##  [9] "luad_tcga_methylation_hm27"
## [10] "luad_tcga_methylation_hm450"
## [11] "luad_tcga_mutations"
```

We start with a convinience function which downloads and create an MAE automatically;

```
LUAD_Multiassay = cBioDataPack(cancer_study_id = 'luad_tcga')
LUAD_Multiassay # look at all the experiments which are available
## A MultiAssayExperiment object of 15 listed
##  experiments with user-defined names and respective classes.
##  Containing an ExperimentList class object of length 15:
##  [1] CNA: SummarizedExperiment with 24776 rows and 516 columns
##  [2] RNA_Seq_v2_expression_median: SummarizedExperiment with 20531 rows and 517 columns
##  [3] RNA_Seq_v2_mRNA_median_Zscores: SummarizedExperiment with 20531 rows and 517 columns
##  [4] cna_hg19.seg: RaggedExperiment with 81799 rows and 518 columns
##  [5] expression_median: SummarizedExperiment with 17814 rows and 32 columns
##  [6] linear_CNA: SummarizedExperiment with 24776 rows and 516 columns
##  [7] mRNA_median_Zscores: SummarizedExperiment with 17814 rows and 32 columns
##  [8] methylation_hm27: SummarizedExperiment with 1788 rows and 126 columns
##  [9] methylation_hm27_normals: SummarizedExperiment with 1788 rows and 24 columns
##  [10] methylation_hm450: SummarizedExperiment with 16237 rows and 460 columns
##  [11] methylation_hm450_normals: SummarizedExperiment with 16237 rows and 32 columns
##  [12] mutations_extended: RaggedExperiment with 72541 rows and 230 columns
##  [13] mutations_mskcc: RaggedExperiment with 72541 rows and 230 columns
##  [14] rppa: SummarizedExperiment with 223 rows and 365 columns
##  [15] rppa_Zscores: SummarizedExperiment with 223 rows and 365 columns
## Features:
##  experiments() - obtain the ExperimentList instance
##  colData() - the primary/phenotype DFrame
##  sampleMap() - the sample availability DFrame
##  ‘$‘, ‘[‘, ‘[[‘ - extract colData columns, subset, or experiment
##  *Format() - convert into a long or wide DFrame
##  assays() - convert ExperimentList to a SimpleList of matrices
```

Let’s investigate the MAE object closer

A. Assays: a list of 15 objects (subsetted above); experimental datasets

```
experiments(LUAD_Multiassay)
## ExperimentList class object of length 15:
##  [1] CNA: SummarizedExperiment with 24776 rows and 516 columns
##  [2] RNA_Seq_v2_expression_median: SummarizedExperiment with 20531 rows and 517 columns
##  [3] RNA_Seq_v2_mRNA_median_Zscores: SummarizedExperiment with 20531 rows and 517 columns
##  [4] cna_hg19.seg: RaggedExperiment with 81799 rows and 518 columns
##  [5] expression_median: SummarizedExperiment with 17814 rows and 32 columns
##  [6] linear_CNA: SummarizedExperiment with 24776 rows and 516 columns
##  [7] mRNA_median_Zscores: SummarizedExperiment with 17814 rows and 32 columns
##  [8] methylation_hm27: SummarizedExperiment with 1788 rows and 126 columns
##  [9] methylation_hm27_normals: SummarizedExperiment with 1788 rows and 24 columns
##  [10] methylation_hm450: SummarizedExperiment with 16237 rows and 460 columns
##  [11] methylation_hm450_normals: SummarizedExperiment with 16237 rows and 32 columns
##  [12] mutations_extended: RaggedExperiment with 72541 rows and 230 columns
##  [13] mutations_mskcc: RaggedExperiment with 72541 rows and 230 columns
##  [14] rppa: SummarizedExperiment with 223 rows and 365 columns
##  [15] rppa_Zscores: SummarizedExperiment with 223 rows and 365 columns
```

B. colData: characteristics of samples (rows) e.g. clinical, pathological data (columns)

```
colData(LUAD_Multiassay)[1:10, 1:10]
## DataFrame with 10 rows and 10 columns
##                PATIENT_ID       SAMPLE_ID                      OTHER_SAMPLE_ID
##               <character>     <character>                          <character>
## TCGA-05-4384 TCGA-05-4384 TCGA-05-4384-01 e4416303-50b0-4316-bfee-030c7b29fac6
## TCGA-05-4390 TCGA-05-4390 TCGA-05-4390-01 c7f76210-d0f2-4fb8-80f1-35098dbe03de
## TCGA-05-4425 TCGA-05-4425 TCGA-05-4425-01 98bee17e-0b07-40d8-a70f-3163dbfd9c79
## TCGA-38-4631 TCGA-38-4631 TCGA-38-4631-01 13b0d336-205a-42ac-8a50-db0eb065013d
## TCGA-38-4632 TCGA-38-4632 TCGA-38-4632-01 d994817a-474e-4f92-8a78-7586caa85d52
## TCGA-38-6178 TCGA-38-6178 TCGA-38-6178-01 81fb5fa9-d901-4faa-b909-376b8676d2bc
## TCGA-44-6144 TCGA-44-6144 TCGA-44-6144-01 aae346a0-532d-48a3-89b2-d0a614097fd5
## TCGA-44-6145 TCGA-44-6145 TCGA-44-6145-01 007783b3-a1b5-4e9b-8add-6d01fefb5697
## TCGA-44-6146 TCGA-44-6146 TCGA-44-6146-01 ad66185f-983d-4416-b705-25d3d4d8b5a5
## TCGA-44-6147 TCGA-44-6147 TCGA-44-6147-01 43f5270a-f2f2-4dd9-83b5-486b805e35b8
##              SPECIMEN_CURRENT_WEIGHT DAYS_TO_COLLECTION
##                          <character>        <character>
## TCGA-05-4384         [Not Available]    [Not Available]
## TCGA-05-4390         [Not Available]    [Not Available]
## TCGA-05-4425         [Not Available]    [Not Available]
## TCGA-38-4631         [Not Available]    [Not Available]
## TCGA-38-4632         [Not Available]    [Not Available]
## TCGA-38-6178         [Not Available]    [Not Available]
## TCGA-44-6144         [Not Available]    [Not Available]
## TCGA-44-6145         [Not Available]    [Not Available]
## TCGA-44-6146         [Not Available]    [Not Available]
## TCGA-44-6147         [Not Available]    [Not Available]
##              DAYS_TO_SPECIMEN_COLLECTION SPECIMEN_FREEZING_METHOD
##                              <character>              <character>
## TCGA-05-4384             [Not Available]          [Not Available]
## TCGA-05-4390             [Not Available]          [Not Available]
## TCGA-05-4425             [Not Available]          [Not Available]
## TCGA-38-4631             [Not Available]          [Not Available]
## TCGA-38-4632             [Not Available]          [Not Available]
## TCGA-38-6178             [Not Available]          [Not Available]
## TCGA-44-6144             [Not Available]          [Not Available]
## TCGA-44-6145             [Not Available]          [Not Available]
## TCGA-44-6146             [Not Available]          [Not Available]
## TCGA-44-6147             [Not Available]          [Not Available]
##              SAMPLE_INITIAL_WEIGHT SPECIMEN_SECOND_LONGEST_DIMENSION
##                        <character>                       <character>
## TCGA-05-4384       [Not Available]                               0.8
## TCGA-05-4390       [Not Available]                               0.9
## TCGA-05-4425       [Not Available]                                 1
## TCGA-38-4631       [Not Available]                               0.4
## TCGA-38-4632       [Not Available]                               0.7
## TCGA-38-6178       [Not Available]                               0.6
## TCGA-44-6144       [Not Available]                               0.6
## TCGA-44-6145       [Not Available]                                 1
## TCGA-44-6146                   420                               0.8
## TCGA-44-6147                    80                               0.8
##                  IS_FFPE
##              <character>
## TCGA-05-4384          NO
## TCGA-05-4390          NO
## TCGA-05-4425          NO
## TCGA-38-4631          NO
## TCGA-38-4632          NO
## TCGA-38-6178          NO
## TCGA-44-6144          NO
## TCGA-44-6145          NO
## TCGA-44-6146          NO
## TCGA-44-6147          NO
```

C. sampleMap:

```
sampleMap(LUAD_Multiassay)
## DataFrame with 4480 rows and 3 columns
##             assay      primary         colname
##          <factor>  <character>     <character>
## 1             CNA TCGA-05-4244 TCGA-05-4244-01
## 2             CNA TCGA-05-4249 TCGA-05-4249-01
## 3             CNA TCGA-05-4250 TCGA-05-4250-01
## 4             CNA TCGA-05-4382 TCGA-05-4382-01
## 5             CNA TCGA-05-4384 TCGA-05-4384-01
## ...           ...          ...             ...
## 4476 rppa_Zscores TCGA-NJ-A550 TCGA-NJ-A550-01
## 4477 rppa_Zscores TCGA-NJ-A55R TCGA-NJ-A55R-01
## 4478 rppa_Zscores TCGA-NJ-A7XG TCGA-NJ-A7XG-01
## 4479 rppa_Zscores TCGA-O1-A52J TCGA-O1-A52J-01
## 4480 rppa_Zscores TCGA-S2-AA1A TCGA-S2-AA1A-01
```

D. MetaData: additional data

```
metadata(LUAD_Multiassay)$name # Name of the study
## [1] "Lung Adenocarcinoma (TCGA, Firehose Legacy)"
metadata(LUAD_Multiassay)$description # where is the data coming from
## [1] "TCGA Lung Adenocarcinoma; raw data at the <A HREF=\"https://tcga-data.nci.nih.gov/\">NCI</A>; s
```

E. assays; to retrieve data for specific Assays/experiments; let's look at CNA

```
LUAD_CNA = assays(LUAD_Multiassay)[’CNA’]

LUAD_CNA_compact = as.data.frame(LUAD_CNA@listData$CNA)
LUAD_CNA_compact[1:5, 1:5] # genes are rows; patients are columns
##         TCGA-05-4244-01 TCGA-05-4249-01 TCGA-05-4250-01 TCGA-05-4382-01
## ACAP3                -1              -1              -1               0
## ACTRT2               -1              -1              -1               0
## AGRN                 -1              -1              -1               0
## ANKRD65              -1              -1              -1               0
## ATAD3A               -1              -1              -1               0
##         TCGA-05-4384-01
## ACAP3                 0
## ACTRT2                0
## AGRN                  0
## ANKRD65               0
## ATAD3A                0
```

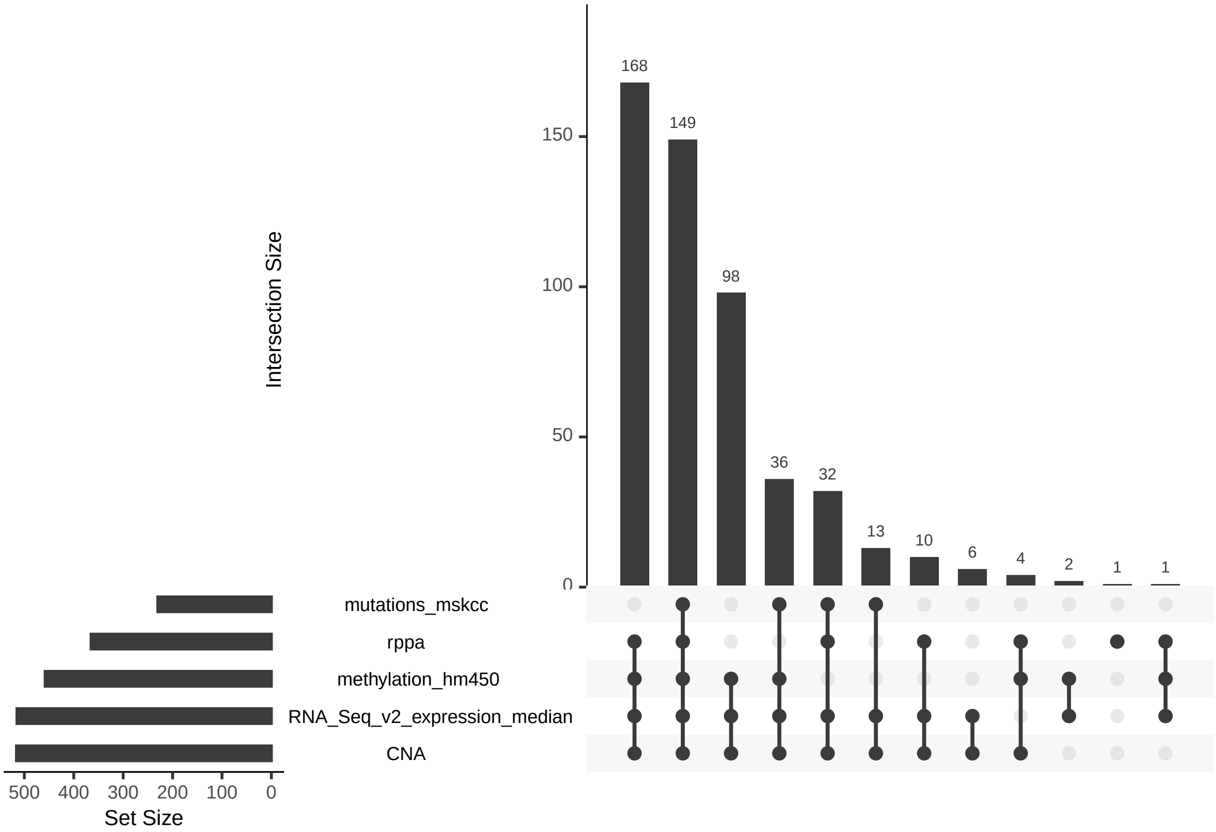
Before we have seen that the whole object consisted of 15 assays; let’s reduce this to the experiments we want to keep

```
LUAD_MAE = LUAD_Multiassay[,, c(’CNA’,
                                ’RNA_Seq_v2_expression_median’,
                                ’methylation_hm450’,
                                ’mutations_mskcc’,
                                ’rppa’)]
## harmonizing input:
##   removing 2392 sampleMap rows not in names(experiments)
##   removing 61 colData rownames not in sampleMap ’primary’

LUAD_MAE
## A MultiAssayExperiment object of 5 listed
##  experiments with user-defined names and respective classes.
##  Containing an ExperimentList class object of length 5:
##  [1] CNA: SummarizedExperiment with 24776 rows and 516 columns
##  [2] RNA_Seq_v2_expression_median: SummarizedExperiment with 20531 rows and 517 columns
##  [3] methylation_hm450: SummarizedExperiment with 16237 rows and 460 columns
##  [4] mutations_mskcc: RaggedExperiment with 72541 rows and 230 columns
##  [5] rppa: SummarizedExperiment with 223 rows and 365 columns
## Features:
##  experiments() - obtain the ExperimentList instance
##  colData() - the primary/phenotype DFrame
##  sampleMap() - the sample availability DFrame
##  ‘$‘, ‘[‘, ‘[[‘ - extract colData columns, subset, or experiment
##  *Format() - convert into a long or wide DFrame
##  assays() - convert ExperimentList to a SimpleList of matrices
```

Let’s work on some examples; what we can do with this MAE object

This plot creates an overview, on how many samples have which measurement

```
upsetSamples(LUAD_MAE)
```

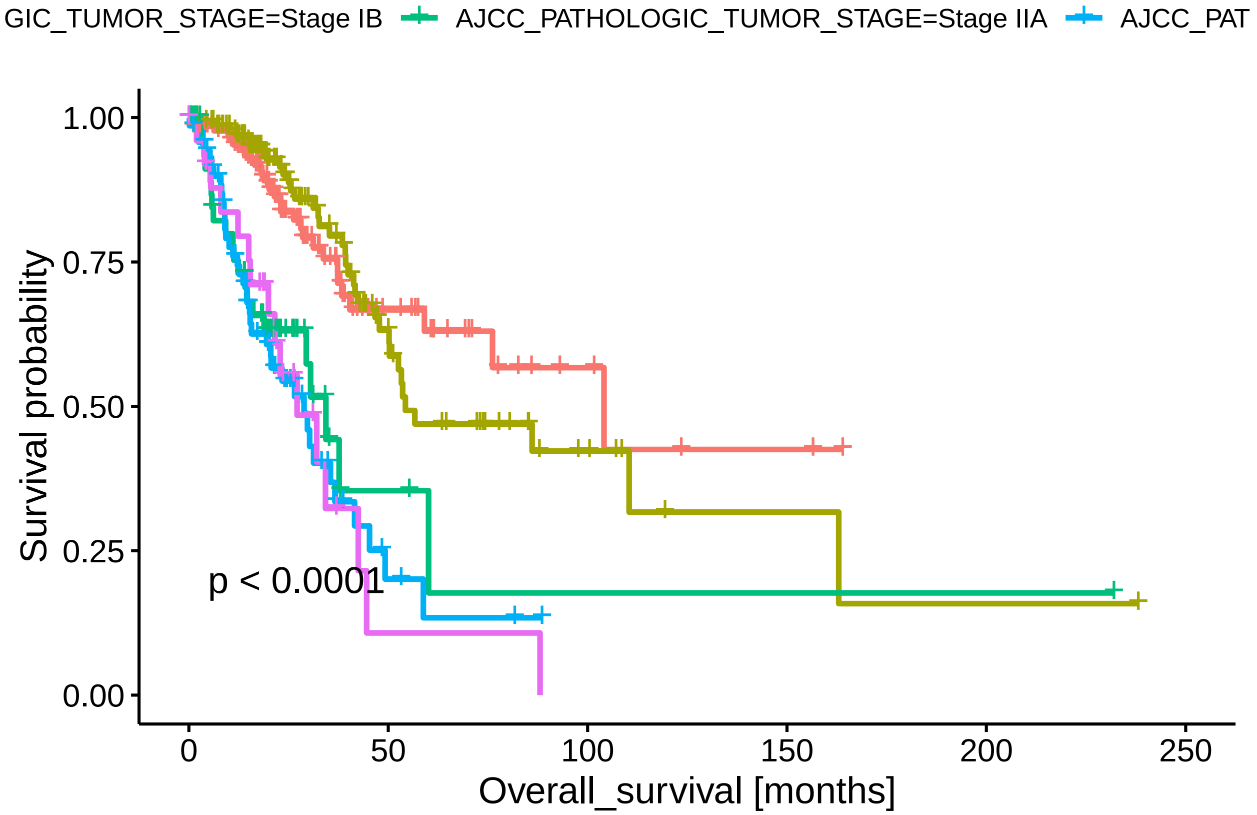


We see that we have full data (selected assays) for 149 samples/patients We also see that (apart from rppa) we have full data (n = 586 patients) for the rest of assays

Example: Kaplan-Meier plot stratified by a clinical variable

The colData can provide clinical data for survival analysis: First, we fetch some clinical data for this cohort and then we create a 'surfit' object:

```
suppressPackageStartupMessages({library(survival); library(survminer)})

## retrieve the clinical data associated with selected cohort;
LUAD_clinical_data = clinicalData(cbio,
                                  studyId = 'luad_tcga')

LUAD_MAE = LUAD_Multiassay[,, c('CNA',
                                'RNA_Seq_v2_expression_median',
                                'methylation_hm450',
                                'mutations_mskcc',
                                'rppa')]
## harmonizing input:
##   removing 2392 sampleMap rows not in names(experiments)
##   removing 61 colData rownames not in sampleMap 'primary'
## now we need to transform and merge clinical data with colData in MAE object
coldata = as.data.frame(colData(LUAD_MAE)) ## transform the colData of MAE object into dataframe
coldata_extended = merge(LUAD_clinical_data[,c(’patientId’,
                                               ’AJCC_PATHOLOGIC_TUMOR_STAGE’,
                                               ’ETHNICITY’,
                                               ’OS_MONTHS’,
                                               ’VITAL_STATUS’)],
                         coldata,
                         by.x = ’patientId’,
                         by.y = ’PATIENT_ID’,
                         all = T)

## remove duplicated samples
coldata_extended = coldata_extended[!duplicated(coldata_extended$patientId), ]
## transform VITAL_STATUS variable (for survival analysis)
coldata_extended$VITAL_STATUS = ifelse(coldata_extended$VITAL_STATUS == ’Alive’, 0, 1)

## now we move to survival analysis
coldata_extended$y = Surv(as.numeric(coldata_extended$OS_MONTHS), coldata_extended$VITAL_STATUS)

## Lets backtransform dataframe to S4 object (neccessary for further handling)
## We are doing this backtransformation, because we can apply handy functions to this object
#- data extraction, subsetting, etc.
colData(LUAD_MAE) = DataFrame(coldata_extended)
## harmonizing input:
##   removing 2088 sampleMap rows with ’primary’ not in colData

## if we now look at the colData for MAE object (compared to before), we see the extension;
## the attributes we just added
colData(LUAD_MAE)[1:6, 1:6]
## DataFrame with 6 rows and 6 columns
##      patientId AJCC_PATHOLOGIC_TUMOR_STAGE   ETHNICITY   OS_MONTHS VITAL_STATUS
##    <character>                 <character> <character> <character>    <numeric>
## 1 TCGA-05-4244                    Stage IV          NA           0            0
## 2 TCGA-05-4245                  Stage IIIA          NA       23.98            0
## 3 TCGA-05-4249                    Stage IB          NA       50.03            0
## 4 TCGA-05-4250                  Stage IIIA          NA        3.98            1
## 5 TCGA-05-4382                    Stage IB          NA       19.94            0
## 6 TCGA-05-4384                  Stage IIIA          NA       13.99            0
##         SAMPLE_ID
##       <character>
## 1 TCGA-05-4244-01
## 2              NA
## 3 TCGA-05-4249-01
## 4 TCGA-05-4250-01
## 5 TCGA-05-4382-01
## 6 TCGA-05-4384-01

## however, for the survival analysis we will again create a data frame (technical reasons)
survival_data = as(colData(LUAD_MAE), ’data.frame’)

## fit an survival object
fit = survfit(y ~ AJCC_PATHOLOGIC_TUMOR_STAGE,
              data = subset(survival_data,
                            AJCC_PATHOLOGIC_TUMOR_STAGE %in% c('Stage IA',
                                                               'Stage IB',
                                                               'Stage IIA',
                                                               'Stage IIIA',
                                                               'Stage IV')))

## and make a plot
ggsurvplot(fit = fit,
           data = survival_data,
           risk.table = F,
           pval = T) +
  labs(x = 'Overall_survival [months]')
```


GIC_TUMOR_STAGE=Stage IB
AJCC_PATHOLOGIC_TUMOR_STAGE=Stage IIA
AJCC_PAT
Survival probability
p < 0.0001
Overall_survival [months]