

Unified classification and risk-stratification in Acute Myeloid Leukemia

Tutorial for Main Figures :

```
In [1]: R.version.string
```

```
'R version 4.1.0 (2021-05-18)'
```

```
In [3]: options(warn=-1)
```

```
# Step 1: Install and load missing libraries
# start_time <- Sys.time()
some_packages <- c("mstate", "survminer", "colorspace", "gridExtra", "stringr")
for (package in some_packages){
#   if(!require(package, character.only = TRUE)){
#     install.packages(package, character.only = TRUE)
#   }
  library(package, character.only=TRUE)
}
# end_time <- Sys.time()
# paste0("It takes :", round(end_time-start_time, 2), " minutes to install all the libraries (survminer takes a LOT of time!)")
```

```
In [4]: source('src/tools_analysis_bis.R')
```

```
statecols <- heat_hcl(6, c = c(80, 30), l = c(30, 90), power = c(1/5, 2))
[c(6,5,4,3,2,1)]
ord <- c(1,2,3,4,5,6)

# CI for sediment plot and barplot :
conf.int <- 0.95
crit <- qnorm((1 - conf.int) / 2, lower.tail = FALSE)
```

```
In [ ]:
```

Helper Functions:

- 1. Classify patients in components and proposal**
- 2. Find principal class**
- 3. ELN classifier**

```
In [5]: classify_components <- function(user_data, keep=c(demo, clin, comp, "ITD"), proposal=T) {
  factors <- c("ZRSR2", "U2AF1_p.S34", "U2AF1_p.Q157", "SRSF2", "SF3B1", "ASXL1", "STAG2", "BCOR", "RUNX1", "EZH2", "MLL", "PHF6", "SF1", "NF1", "CUX1", "SETBP1")

  layer1 <- c("full_component_t_15_17", "full_component_inv_16", "full_component_t_8_21",
             "full_component_t_11", "full_component_t_6_9", "full_component_inv_3",
             "full_component_NPM1", "full_component_CEBPA_bi", "full_component_TP53_complex",
             "full_component_sAML2")

  user_data["full_component_t_15_17", "value"] <- ifelse(user_data["t_15_17", "value"] == 1, 1, 0)

  user_data["full_component_inv_16", "value"] <- ifelse(user_data["inv_16", "value"] == 1, 1, 0)

  user_data["full_component_t_8_21", "value"] <- ifelse(user_data["t_8_21", "value"] == 1, 1, 0)

  user_data["full_component_t_11", "value"] <- ifelse((user_data["t_v_11", "value"] == 1 | user_data["t_9_11", "value"] == 1), 1, 0)

  user_data["full_component_t_6_9", "value"] <- ifelse(user_data["t_6_9", "value"] == 1, 1, 0)

  user_data["full_component_inv_3", "value"] <- ifelse(user_data["inv_3", "value"] == 1, 1, 0)

  user_data["full_component_NPM1", "value"] <- ifelse(user_data["NPM1", "value"] == 1, 1, 0)

  user_data["full_component_CEBPA_bi", "value"] <- ifelse(user_data["CEBPA_bi", "value"] == 1, 1, 0)

#    user_data["full_component_Trisomies", "value"] <- ifelse((user_data["add_8", "value"] == 1 | user_data["add_11", "value"] == 1 |
#    user_data["add_13", "value"] == 1 | user_data["add_21", "value"] == 1 |
#    user_data["add_22", "value"] == 1), 1, 0)

  user_data["full_component_TP53_complex", "value"] <- ifelse((user_data["TP53", "value"] == 1 | user_data["complex", "value"] == 1), 1, 0)

  # Remove from TP53-complex the ones that are complex, no TP53 and no deletions.
  # get names of TP53 complex without TP53 and no deletions so that we move them later to Trisomies

  num_deletions <- sum(user_data[grep("minus", rownames(user_data)), grep("del_", rownames(user_data))], "value")
```

```

no_deletions_for_complex_no_TP53 <- ifelse((user_data["TP53","value"]==0 & user_data["complex","value"]==1 & num_deletions==0),T,F)

user_data["full_component_TP53_complex","value"] <- ifelse((user_data["TP53","value"]==0 & user_data["complex","value"]==1 & num_deletions==0) ,0,user_data["full_component_TP53_complex","value"])

# Layer 2

user_data["full_component_sAML2","value"] <- ifelse(sum(user_data[factors,"value"],na.rm=T)>1 ,1,0)

user_data["full_component_sAML1","value"] <- ifelse(sum(user_data[factors,"value"],na.rm=T)==1 & sum(user_data[layer1,"value"])==0 ,1,0)

user_data["full_component_sAML2","value"] <- ifelse(user_data["full_component_sAML1","value"]==1 &
(user_data["DNMT3A","value"]==1 | user_data["TET2","value"]==1 ),1,
user_data["full_component_sAML2","value"])

user_data["full_component_sAML1","value"] <- ifelse(user_data["full_component_sAML1","value"]==1 &
(user_data["DNMT3A","value"]==1 | user_data["TET2","value"]==1 ),0,
user_data["full_component_sAML1","value"])

user_data["full_component_WT1","value"] <- ifelse( sum(user_data[c(layer1,"full_component_sAML1"),"value"])==0 & user_data["WT1","value"]==1 ,1,0)

##### Special rules Trisomies#####
# 1) one of main Trisomies (+8,+11,+13,+21,+22) and did not belong to other components
user_data["full_component_Trisomies","value"] <- ifelse(sum(user_data[c(layer1,"full_component_sAML1","full_component_WT1"),"value"])==0 & (user_data["add_8","value"]==1 | user_data["add_11","value"]==1 |
user_data["add_13","value"]==1 | user_data["add_21","value"]==1 |
user_data["add_22","value"]==1),1,0)

# 2) Add in Trisomies the ones that are complex, no TP53 and no deletions i.e

user_data["full_component_Trisomies","value"] <- ifelse(no_deletions_for_complex_no_TP53,1,user_data["full_component_Trisomies","value"])

# 3) Finally, remove previous overlaps

user_data["full_component_Trisomies","value"] <- ifelse(sum(user_data[c(layer1,"full_component_sAML1","full_component_WT1"),"value"])>0,0,user_data["full_component_Trisomies","value"])

```

```

    user_data[ "full_component_DNMT3A_IDH1_2", "value" ] <- ifelse((user_da
ta[ "DNMT3A", "value" ]==1 & (user_data[ "IDH1", "value" ]==1 | user_data[ "IDH
2_p.R140", "value" ]==1 | user_data[ "IDH2_p.R172", "value" ]==1)) &
                           sum(user
_data[c(layer1,"full_component_sAML1","full_component_WT1","full_compo
ne_Trisomies"),"value"])==0,1,0)

# Layer 3

    user_data[ "full_component_mNOS", "value" ] <- ifelse((sum(user_data[c(
all_gen,all_cyto),"value"],na.rm=T)>0 & sum(user_data[comp,"value"],na.r
m=T)==0),1,0)

    user_data[ "full_component_no_events", "value" ] <- ifelse(sum(user_dat
a[c(all_gen,all_cyto),"value"],na.rm=T)==0 ,1,0)

    user_data[ "overlap", "value" ] <- ifelse(sum(user_data[comp,"value"],n
a.rm=T) >1 ,1,0)

# Get princ comp for the risk proposal

    comp_order <- c("full_component_NPM1", "full_component_inv_16", "full_
component_t_8_21", "full_component_t_15_17", "full_component_TP53_complex"
, "full_component_inv_3", "full_component_t_11", "full_component_t_6_9",
                           "full_component_sAML1", "full_component_sAML2", "ful
l_component_CEBPA_bi",
                           "full_component_WT1",
                           "full_component_Trisomies",
                           "full_component_DNMT3A_IDH1_2",
                           "full_component_mNOS", "full_component_no_events")

princ_comp <- comp_order[which(user_data[comp_order,"value"]==1)][1]

    if (proposal==T){

        user_data[c( "NEW_favorable", "NEW_intermediate", "NEW_adverse" ), "v
alue" ] <- 0
    }

```

```

    user_data[ "NEW_favorable", "value" ] <- ifelse(princ_comp %in% c(
  "full_component_NPM1" , "full_component_inv_16" , "full_component_t_8_2
  1" , "full_component_t_15_17",
  "full_component_CEBPA_bi" , "full_component_no_events"),1,0)

    user_data[ "NEW_intermediate", "value" ] <- ifelse(princ_comp %in%
  c( "full_component_sAML1" , "full_component_t_6_9" , "full_component_DNM
  T3A_IDH1_2" ,
  "full_component_Trisomies" , "full_component_t_11" , "full_c
  omponent_WT1" , "full_component_mNOS"),1,0)

    user_data[ "NEW_adverse", "value" ] <- ifelse(princ_comp %in% c( "f
  ull_component_sAML2" , "full_component_TP53_complex" , "full_component_i
  nv_3"==1),1,0)

#order is important! adverse then intermediate otherwise we ove
rlap

    user_data[ "NEW_adverse", "value" ] <- ifelse(user_data[ "NEW_interm
  ediate", "value" ]==1 & user_data[ "ITD", "value" ]==1 ,1,user_data[ "NEW_adve
  rse", "value" ] )
    user_data[ "NEW_intermediate", "value" ] <- ifelse(user_data[ "NEW_i
  ntermediate", "value" ]==1 & user_data[ "ITD", "value" ]==1 ,0,user_data[ "NEW
  _intermediate", "value" ])

    user_data[ "NEW_intermediate", "value" ] <- ifelse(user_data[ "NEW_f
  avorable", "value" ]==1 & user_data[ "NPM1", "value" ]==1 & user_data[ "ITD",
  "value" ]==1 ,
  1,user_data[ "NEW
  _intermediate", "value" ]) # changing with NPM1

    user_data[ "NEW_favorable", "value" ] <- ifelse(user_data[ "NEW_favo
  rable", "value" ]==1 & user_data[ "NPM1", "value" ]==1 & user_data[ "ITD", "val
  ue" ]==1 ,
  0,user_data[ "NEW_fa
  vorable", "value" ]) # changing with NPM1

    return(user_data[c(keep, "NEW_favorable", "NEW_intermediate", "NEW_
  adverse"),,drop=F])

  }else{
    return(user_data[keep,,drop=F])
  }

}

```

```
In [6]: find_principal_component <- function(user_data_simplified){  
  comp_order <- comp_order <- c("full_component_NPM1", "full_component_inv_16", "full_component_t_8_21", "full_component_t_15_17", "full_component_TP53_complex", "full_component_inv_3", "full_component_t_11", "full_component_t_6_9",  
    "full_component_sAML1", "full_component_sAML2", "full_component_CEBPA_bi",  
    "full_component_WT1",  
    "full_component_Trisomies",  
    "full_component_DNMT3A_IDH1_2",  
    "full_component_mNOS", "full_component_no_events")  
  
  # Check if ITD:  
  ITD_mut <- ifelse(user_data_simplified["ITD", "value"]==1, "_ITD", "")  
  
  tmp <- user_data_simplified  
  princ_component <- paste(comp_order[which(tmp[comp_order, , drop=F]==1)  
  )[1], ITD_mut, sep="")  
  return (princ_component)  
}
```

```
In [7]: eln_classification_webapp_simplified <- function(df){
  df$eln_2017 <- "null"
  df[df$eln_2017=="null" & df$inv_16==1,"eln_2017"] <- "favorable"
  df[df$eln_2017=="null" & df$t_15_17==1,"eln_2017"] <- "favorable"
  df[df$eln_2017=="null" & df$t_9_11==1,"eln_2017"] <- "intermediate"
  df[df$eln_2017=="null" & df$complex==1,"eln_2017"] <- "adverse"
  df[df$eln_2017=="null" & df$TP53==1,"eln_2017"] <- "adverse"
  df[df$eln_2017=="null" & df$t_6_9==1,"eln_2017"] <- "adverse"

  if ("MLL_PTD" %in% colnames(df)){
    df[df$eln_2017=="null" & df$MLL_PTD==1,"eln_2017"] <- "adverse"
  } else {
    df[df$eln_2017=="null" & df$MLL==1,"eln_2017"] <- "adverse"
  }
  df[df$eln_2017=="null" & df$t_9_22==1,"eln_2017"] <- "adverse"
  df[df$eln_2017=="null" & df$inv_3==1,"eln_2017"] <- "adverse"

  for (col in colnames(df)[grep("^-|del", colnames(df)) & grep("5|7|17", colnames(df))]){
    df[df$eln_2017=="null" & df[,col]==1,"eln_2017"] <- "adverse"
  }
  df[df$eln_2017=="null" & df$NPM1==0 & df$ITD==1,"eln_2017"] <- "adverse"

  df[df$eln_2017=="null" & df$t_8_21==1,"eln_2017"] <- "favorable"
  df[df$eln_2017=="null" & df$CEBPA_bi==1,"eln_2017"] <- "favorable"
  df[df$eln_2017=="null" & df$NPM1==1 & df$ITD==0,"eln_2017"] <- "favorable"

  df[df$eln_2017=="null" & df$RUNX1==1,"eln_2017"] <- "adverse"
  df[df$eln_2017=="null" & df$ASXL1==1,"eln_2017"] <- "adverse"

  for (col in colnames(df)[grep("^-|del", colnames(df))]){
    df[df$eln_2017=="null" & df[,col]==1,"eln_2017"] <- "adverse"
  }

  df[df$eln_2017=="null" & df$NPM1==0 & df$ITD==0,"eln_2017"] <- "intermediate"
  df[df$eln_2017=="null" & df$NPM1==1 & df$ITD==1,"eln_2017"] <- "intermediate"

  df[df$eln_2017=="null" & df$t_v_11==1,"eln_2017"] <- "adverse"

  df[df$eln_2017=="null" , "eln_2017"] <- "intermediate"

  return (df)
}
```

End Helper

```
In [8]: list_intensively_treated <- readRDS("data/list_intensively_treated.rds")
df_nejm <- read.table("data/df_nejm_personalization.tsv")
df_nejm$patient_id <- rownames(df_nejm)
dim(df_nejm)
df_cardiff <- read.table("data/df_TRAINING_personalization_updated.tsv")
df_cardiff <- df_cardiff[rownames(df_cardiff) %in% list_intensively_treated,]
df_cardiff$patient_id <- rownames(df_cardiff)
dim(df_cardiff)
all_gen <- colnames(df_cardiff)[c(34:117)]
all_cyto <- colnames(df_cardiff)[c(118:187)]


## add other additions and deletions ( it will be useful for class Trisomies and TP53-complex differentiation )
for (i in 1:22) {
  if(!paste("add_",i,sep="") %in% all_cyto){
    all_cyto <- c(all_cyto,paste("add_",i,sep=""))
  }
  if(!paste("del_",i,sep="") %in% all_cyto){
    all_cyto <- c(all_cyto,paste("del_",i,sep=""))
  }
}

comp <- c(colnames(df_cardiff)[startsWith(colnames(df_cardiff),"full_component_")],"overlap")
clin <- c("ahd","perf_status","bm_blasts","secondary","wbc","hb","plt")
demo <- c("gender","age")

cols_to_keep <- c("patient_id","CR","Relapse","Death_in_CR",
                   "Death_in_Relapse","Death_without",
                   "CR_stat","Relapse_stat","Death_in_CR_stat",
                   "Death_in_Relapse_stat","Death_without_stat",
                   c("ITD",comp,clin,demo,"NEW_favorable","NEW_intermediate"
                     ,"NEW_adverse",
                     "eln_2017_favorable","eln_2017_intermediate","eln_2017_adverse"))
library(mstate)
data <- rbind(df_nejm[,cols_to_keep],df_cardiff[,cols_to_keep])
data[data$bm_blasts<5,]$bm_blasts <- mean(data[data$bm_blasts>=5,]$bm_blasts)
# Create Transition Matrix
tmat_data <- transMat(x = list(c(2,6), c(3, 4), c(5), c(),c(),c()),
                       names = c("Alive", "Alive in CR", "Alive in Relapse",
                                "Death in CR", "Death in Relapse", "Death without CR"))

msdata <- msprep(data = data, trans = tmat_data,
                  time = c(NA, "CR", "Relapse", "Death_in_CR", "Death_in_Relapse", "Death_without"),
                  status = c(NA, "CR_stat", "Relapse_stat", "Death_in_CR_stat",
                            "Death_in_Relapse_stat", "Death_without_stat"),
                  ),
```

```
keep=c("patient_id", "ITD", comp, clin, demo,
      "NEW_favorable", "NEW_intermediate", "NEW_adverse"
      ,
      "eln_2017_favorable", "eln_2017_intermediate", "eln
      _2017_adverse"))
# Age as time dependent covariate
msdata$age <- msdata$age + msdata$Tstart

events(msdata)$Frequencies
events(msdata)$Proportions
```

1540 · 146

1661 · 221

from	to	Alive	Alive in CR	Alive in Relapse	Death in CR
Alive		0	2653		0
Alive in CR		0	0		1331
Alive in Relapse		0	0		0
Death in CR		0	0		0
Death in Relapse		0	0		0
Death without CR		0	0		0
from	to	Death in Relapse	Death without CR	no event	total ent
Alive		0		529	19
3201					
Alive in CR		0		0	970
2653					
Alive in Relapse		1025		0	306
1331					
Death in CR		0		0	352
352					
Death in Relapse		0		0	1025
1025					
Death without CR		0		0	529
529					
from	to	Alive	Alive in CR	Alive in Relapse	Death in CR
Alive		0.000000000	0.828803499		0.000000000 0.000000000
Alive in CR		0.000000000	0.000000000		0.501696193 0.132679985
Alive in Relapse		0.000000000	0.000000000		0.000000000 0.000000000
Death in CR		0.000000000	0.000000000		0.000000000 0.000000000
Death in Relapse		0.000000000	0.000000000		0.000000000 0.000000000
Death without CR		0.000000000	0.000000000		0.000000000 0.000000000
from	to	Death in Relapse	Death without CR	no event	
Alive		0.000000000	0.165260856	0.005935645	
Alive in CR		0.000000000	0.000000000	0.365623822	
Alive in Relapse		0.770097671	0.000000000	0.229902329	
Death in CR		0.000000000	0.000000000	1.000000000	
Death in Relapse		0.000000000	0.000000000	1.000000000	
Death without CR		0.000000000	0.000000000	1.000000000	

Store clinical covariate for imputation of missing data

```
In [9]: covariates_median <- sapply(msdata[,c(clin,demo)], FUN=median, )  
t(covariates_median)
```

A matrix: 1 × 9 of type dbl

ahd	perf_status	bm_blasts	secondary	wbc	hb	plt	gender	age
0	1	70	1	16	9.2	56	1	52.07945

Filter warning example

```
In [10]: user_data <- data.frame(covariates = c(demo,clin,all_gen,all_cyto))
user_data$value <- "Not specified"
rownames(user_data) <- user_data$covariates

user_data$covariates <- NULL
user_data[c("age","wbc","plt","bm_blasts","hb"), "value"] <- c(75,30,43,1
2,2000)

importance_sig_clin <- c("age","wbc","plt","bm_blasts")

for (var in importance_sig_clin){
    if (quantile(data[,var],probs=c(.025,.975))[[1]] > as.numeric(user_d
ata[var,"value"])){
        print (paste("Patient Model Sediment Plot Output 's confidence i
s low because 97.5% of the patients have",var,"values greater than tha
t."))
    }
    if (quantile(data[,var],probs=c(.025,.975))[[2]] < as.numeric(user_d
ata[var,"value"])){
        print (paste("Patient Model Sediment Plot Output 's confidence i
s low because 97.5% of the patients have",var,"values smaller than tha
t."))
    }
}

if (quantile(data[,"hb"],probs=c(.025,.975))[[1]] > as.numeric(user_data
["hb","value"])){
    print ("97.5% of the patients have hemoglobin values greater than th
at.")
}
if (quantile(data[,"hb"],probs=c(.025,.975))[[2]] < as.numeric(user_
data["hb","value"])){
    print ("97.5% of the patients have hemoglobin values smaller than th
at.")
}
```

[1] "Patient Model Sediment Plot Output 's confidence is low because 9
7.5% of the patients have age values smaller than that."
[1] "Patient Model Sediment Plot Output 's confidence is low because 9
7.5% of the patients have bm_blasts values greater than that."
[1] "97.5% of the patients have hemoglobin values smaller than that."

Forest Plot for Cox transitions

```
In [11]: # Normalize Continuous Covariates for interpretation of the pre computed
transition hazard ratio

normalized<-function(y) {

  x<-y[!is.na(y)]

  x<-(x - min(x)) / (max(x) - min(x))

  y[ !is.na(y)]<-x

  return(y)
}

set_notebook_plot_size(15,10)
msdata_precomputed <- msdata[!(msdata$trans==1 & msdata$perf_status==4
),] ### Handle perf status =4 for 2 rows in trans 1 that mess up with t
he ggforest because of one coefficient

msdata_precomputed$gender <- factor(msdata_precomputed$gender,labels=c(
"Female","Male"))
msdata_precomputed$ahd <- factor(msdata_precomputed$ahd,labels=c("no","y
es"))
msdata_precomputed$perf_status <- factor(msdata_precomputed$perf_status,
labels=c("Fully active","Restricted in physical activity","capable of se
lfcare but no work activities","limited selfcare","completely disabled"
))
msdata_precomputed$secondary <- factor(msdata_precomputed$secondary,labe
ls=c("de novo","secondary","tAML"))
# msdata_precomputed[,c("secondary")] <- lapply(msdata_precomputed[,c("s
econdary")],as.factor)

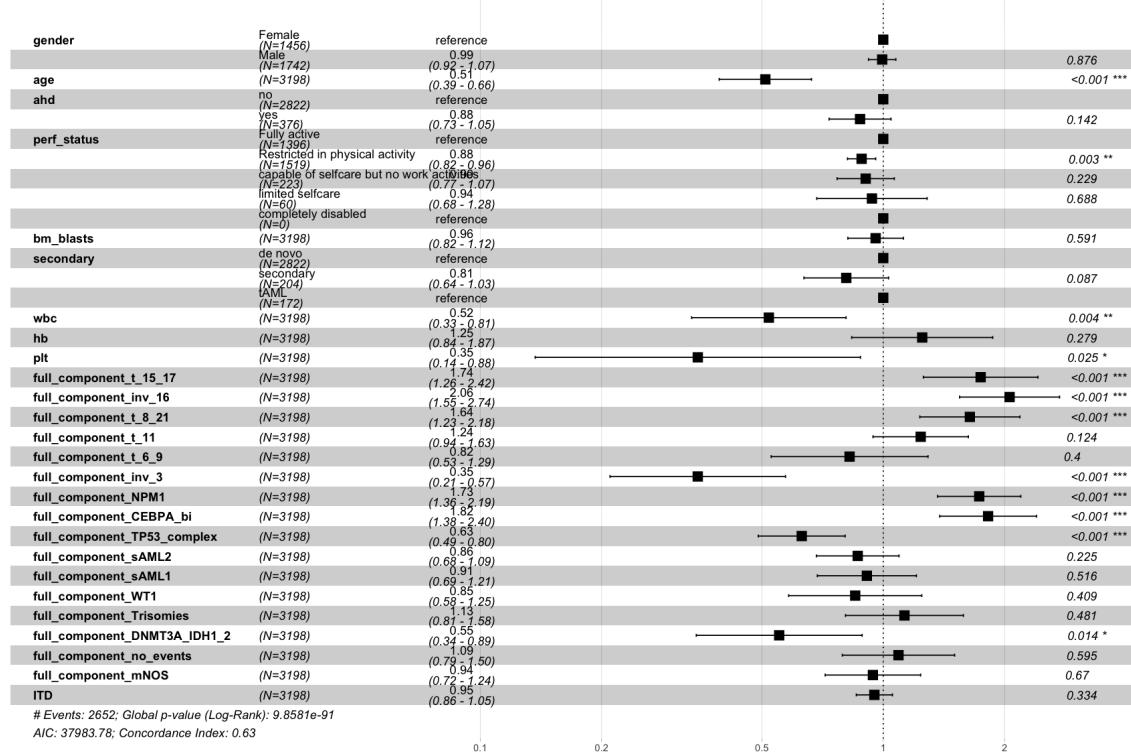
msdata_precomputed[,c("age","bm_blasts","wbc","hb","plt")] <- apply(msda
ta_precomputed[,c("age","bm_blasts","wbc","hb","plt")],2,normalized)
# msdata_precomputed[,c(clin,demo)] <- apply(msdata_precomputed[,c(clin,
demo)],2,normalized)
covariates <- c(demo,clin,comp[comp!="overlap"],"ITD")
form <- "Surv(Tstart, Tstop, status) ~ "
for (co in c(covariates)){
  form <- paste (form,co,sep = " + ")
}

# Plot Forest Global

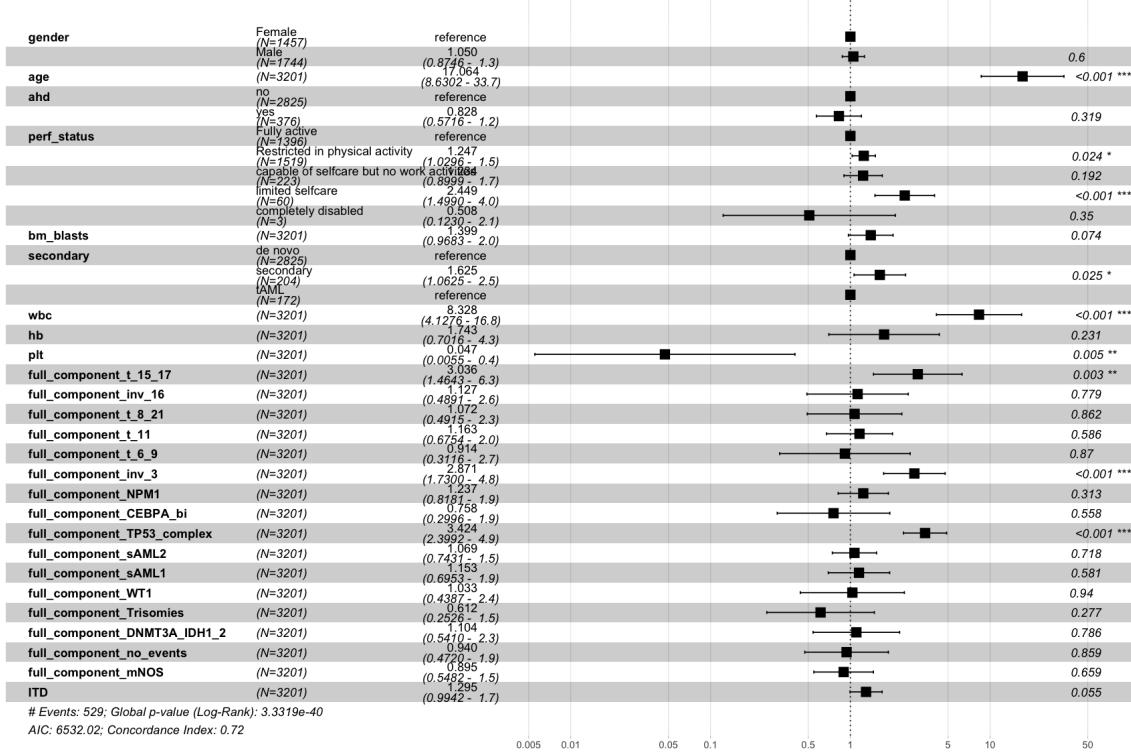
data_trans <- list()
cox_trans <- list()
plot_forest <- list()
set.seed(17)
for (i in c(1:5)){
  data_trans[[i]] <- msdata_precomputed[msdata_precomputed$trans==i,]
  formula <- form
  set.seed(9)
  if(i %in% c(3,4)){
    data_trans[[i]] <- data_trans[[i]][data_trans[[i]]$perf_status!=
```

```
"completely disabled",]  
    #data_trans[[i]]$perf_status <- factor(data_trans[[i]]$perf_status,  
                                         #labels=c("Fully active",  
                                         "Restricted in physical activity", "capable of selfcare but no work activities", "limited selfcare"))  
}  
cox_trans[[i]] <- coxph(as.formula(formula), data = data_trans[[i]],  
method = "breslow")  
title <- ifelse(i==1,"Alive --> Alive in CR",  
                 ifelse(i==2,"Alive --> Death Without CR",  
                       ifelse(i==3,"Alive in CR -->Alive with Relapse"  
,                           ifelse(i==4,"Alive in CR --> Death in CR"  
,"Alive with Relapse --> Death With Relapse"))))  
plot_forest[[i]] <- ggforest(cox_trans[[i]],main=paste("Hazard ratio  
for ",title),fontsize=1)  
}  
plot_forest
```

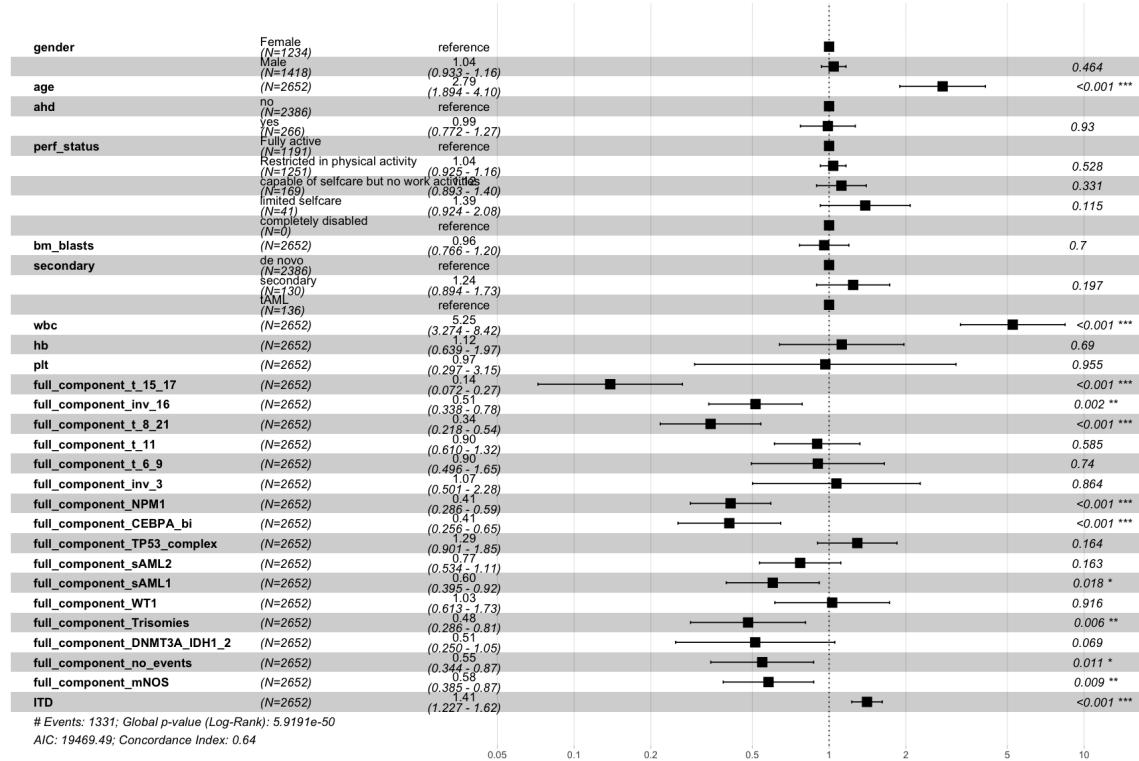
Hazard ratio for Alive --> Alive in CR



Hazard ratio for Alive --> Death Without CR



Hazard ratio for Alive in CR -->Alive with Relapse



[[1]]

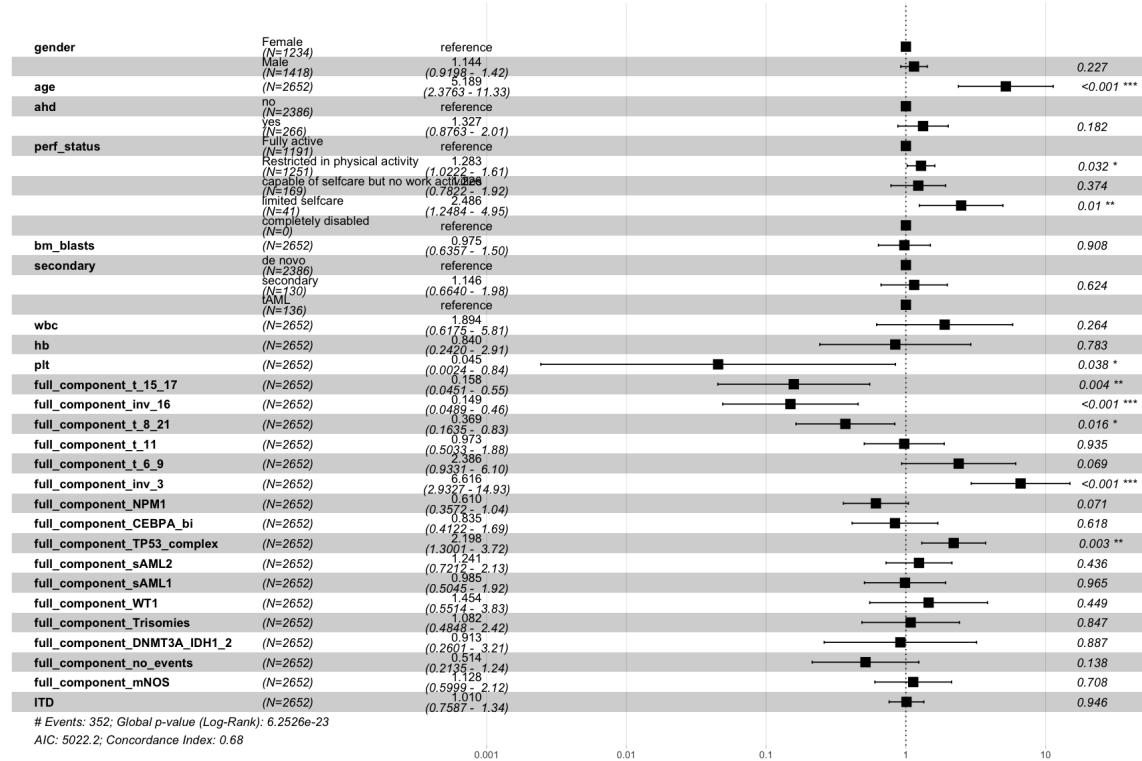
[[2]]

[[3]]

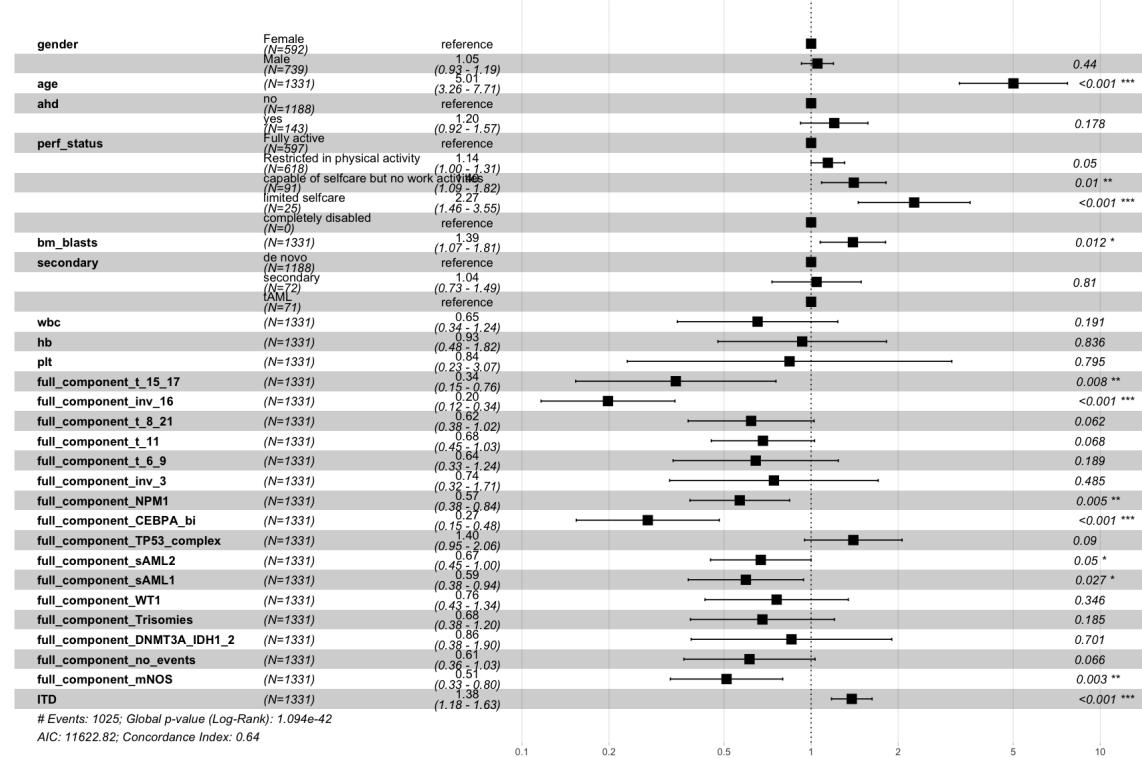
[[4]]

[[5]]

Hazard ratio for Alive in CR --> Death in CR



Hazard ratio for Alive with Relapse --> Death With Relapse



I) Prepare the Model for computation and patient-centric inference.

```
In [12]: msdata <- msprep(data = data, trans = tmat_data,
                           time = c(NA, "CR", "Relapse", "Death_in_CR", "Death_in_Relapse", "Death_without"),
                           status = c(NA, "CR_stat", "Relapse_stat", "Death_in_CR_stat",
                                     "Death_in_Relapse_stat", "Death_without_stat"),
                           ),
                           keep=c("patient_id", "ITD", comp, clin, demo,
                                  "NEW_favorable", "NEW_intermediate", "NEW_adverse",
                                  ,
                                  "eln_2017_favorable", "eln_2017_intermediate", "eln_2017_adverse"))
# Age as time dependent covariate
msdata$age <- msdata$age + msdata$Tstart

msdata <- expand.covs(msdata, covs = c("ITD", comp, clin, demo,
                                         "NEW_favorable", "NEW_intermediate", "NEW_adverse",
                                         "eln_2017_favorable", "eln_2017_intermediate", "eln_2017_adverse"),
                                         longnames = FALSE)
expanded_itd <- colnames(msdata)[grep("ITD.", colnames(msdata))]
expanded_component <- colnames(msdata)[grep("full_component|overlap", colnames(msdata)) & grep("\\.", colnames(msdata))]
expanded_clin <- colnames(msdata)[grep("ahd|perf_status|bm_blasts|secondary|wbc|hb|plt", colnames(msdata)) & grep("\\.", colnames(msdata))]
expanded_demo <- colnames(msdata)[grep("gender|age", colnames(msdata)) & grep("\\.", colnames(msdata))]
expanded_proposal <- colnames(msdata)[grep("NEW_", colnames(msdata)) & grep("\\.", colnames(msdata))]
expanded_eln <- colnames(msdata)[grep("eln_", colnames(msdata)) & grep("\\.", colnames(msdata))]
```

1. Prepare full model for patient calculator

```
In [13]: # Prepare Full Model
covariates <- c(demo,clin,comp[comp!="overlap"],"ITD")
form <- "Surv(Tstart, Tstop, status) ~ "
for (co in c(covariates)){
    for (sub in c(".1",".2",".3",".4",".5")){
        form <- paste (form,paste(co,sub,sep=""),sep = " + ")
    }
}
form <- paste(form,"+ strata(trans)")
set.seed(9)
cfull <- coxph(as.formula(form), data = msdata, method = "breslow")
cfull
```

Call:

coxph(formula = as.formula(form), data = msdata, method = "breslow")

		coef	exp(coef)	se(coef)	z
p					
gender.1	0.887893	-5.629e-03	9.944e-01	3.993e-02	-0.141
gender.2	0.738788	3.103e-02	1.032e+00	9.307e-02	0.333
gender.3	0.391535	4.845e-02	1.050e+00	5.655e-02	0.857
gender.4	0.210725	1.388e-01	1.149e+00	1.109e-01	1.252
gender.5	0.431460	5.117e-02	1.052e+00	6.504e-02	0.787
age.1	5.61e-07	-7.847e-03	9.922e-01	1.568e-03	-5.004
age.2	6.62e-16	3.277e-02	1.033e+00	4.057e-03	8.077
age.3	1.30e-07	1.212e-02	1.012e+00	2.296e-03	5.279
age.4	4.23e-05	1.900e-02	1.019e+00	4.640e-03	4.095
age.5	2.42e-13	1.875e-02	1.019e+00	2.561e-03	7.323
ahd.1	0.004883	-5.606e-01	5.708e-01	1.992e-01	-2.815
ahd.2	0.008992	8.063e-01	2.240e+00	3.087e-01	2.612
ahd.3	0.117886	4.194e-01	1.521e+00	2.682e-01	1.564
ahd.4	0.243962	5.229e-01	1.687e+00	4.487e-01	1.165
ahd.5	0.338153	2.756e-01	1.317e+00	2.878e-01	0.958
perf_status.1	0.013236	-7.468e-02	9.280e-01	3.015e-02	-2.477
perf_status.2	0.011718	1.419e-01	1.152e+00	5.628e-02	2.521
perf_status.3	0.210717	5.129e-02	1.053e+00	4.098e-02	1.252
perf_status.4	0.013047	1.939e-01	1.214e+00	7.811e-02	2.482
perf_status.5	0.000119	1.869e-01	1.205e+00	4.856e-02	3.848
bm_blasts.1	0.551913	-5.074e-04	9.995e-01	8.529e-04	-0.595
bm_blasts.2	0.030229	4.239e-03	1.004e+00	1.956e-03	2.167
bm_blasts.3	0.645503	-5.496e-04	9.995e-01	1.195e-03	-0.460
bm_blasts.4	0.882003	-3.406e-04	9.997e-01	2.294e-03	-0.148
bm_blasts.5	0.014620	3.410e-03	1.003e+00	1.396e-03	2.442

	Patient_Centric_Web_App				
secondary.1	2.149e-01	1.240e+00	1.234e-01	1.742	
0.081587					
secondary.2	-4.891e-01	6.132e-01	2.166e-01	-2.258	
0.023967					
secondary.3	-2.150e-01	8.065e-01	1.683e-01	-1.278	
0.201392					
secondary.4	-1.243e-01	8.831e-01	2.784e-01	-0.446	
0.655280					
secondary.5	-5.237e-02	9.490e-01	1.811e-01	-0.289	
0.772427					
wbc.1	-1.153e-03	9.988e-01	4.207e-04	-2.742	
0.006114					
wbc.2	4.027e-03	1.004e+00	6.609e-04	6.093	
1.11e-09					
wbc.3	3.167e-03	1.003e+00	4.507e-04	7.026	
2.12e-12					
wbc.4	1.255e-03	1.001e+00	1.072e-03	1.171	
0.241786					
wbc.5	-7.102e-04	9.993e-01	6.066e-04	-1.171	
0.241726					
hb.1	1.400e-03	1.001e+00	1.171e-03	1.196	
0.231692					
hb.2	3.244e-03	1.003e+00	2.640e-03	1.229	
0.219104					
hb.3	7.035e-04	1.001e+00	1.630e-03	0.432	
0.665948					
hb.4	-1.107e-03	9.989e-01	3.611e-03	-0.306	
0.759239					
hb.5	-2.225e-04	9.998e-01	1.949e-03	-0.114	
0.909084					
plt.1	-5.304e-04	9.995e-01	2.360e-04	-2.248	
0.024582					
plt.2	-1.454e-03	9.985e-01	5.425e-04	-2.680	
0.007357					
plt.3	-7.156e-06	1.000e+00	2.989e-04	-0.024	
0.980897					
plt.4	-1.552e-03	9.984e-01	7.429e-04	-2.089	
0.036688					
plt.5	-6.970e-05	9.999e-01	3.270e-04	-0.213	
0.831232					
full_component_t_15_17.1	5.443e-01	1.723e+00	1.668e-01	3.264	
0.001098					
full_component_t_15_17.2	1.134e+00	3.108e+00	3.714e-01	3.053	
0.002263					
full_component_t_15_17.3	-1.969e+00	1.395e-01	3.324e-01	-5.925	
3.12e-09					
full_component_t_15_17.4	-1.813e+00	1.631e-01	6.378e-01	-2.843	
0.004468					
full_component_t_15_17.5	-1.043e+00	3.524e-01	4.058e-01	-2.570	
0.010160					
full_component_inv_16.1	7.144e-01	2.043e+00	1.457e-01	4.904	
9.39e-07					
full_component_inv_16.2	1.491e-01	1.161e+00	4.255e-01	0.350	
0.726072					
full_component_inv_16.3	-6.573e-01	5.182e-01	2.149e-01	-3.059	
0.002217					

	Patient_Centric_Web_App				
full_component_inv_16.4 0.000918	-1.885e+00	1.518e-01	5.687e-01	-3.315	
full_component_inv_16.5 4.34e-09	-1.579e+00	2.062e-01	2.689e-01	-5.871	
full_component_t_8_21.1 0.000839	4.871e-01	1.628e+00	1.459e-01	3.340	
full_component_t_8_21.2 0.797024	1.022e-01	1.108e+00	3.973e-01	0.257	
full_component_t_8_21.3 4.30e-06	-1.064e+00	3.452e-01	2.314e-01	-4.596	
full_component_t_8_21.4 0.017386	-9.875e-01	3.725e-01	4.152e-01	-2.378	
full_component_t_8_21.5 0.085815	-4.343e-01	6.477e-01	2.528e-01	-1.718	
full_component_t_11.1 0.123636	2.140e-01	1.239e+00	1.390e-01	1.540	
full_component_t_11.2 0.546518	1.672e-01	1.182e+00	2.773e-01	0.603	
full_component_t_11.3 0.628305	-9.529e-02	9.091e-01	1.968e-01	-0.484	
full_component_t_11.4 0.985687	-6.024e-03	9.940e-01	3.358e-01	-0.018	
full_component_t_11.5 0.086887	-3.539e-01	7.020e-01	2.067e-01	-1.712	
full_component_t_6_9.1 0.383020	-1.996e-01	8.191e-01	2.288e-01	-0.872	
full_component_t_6_9.2 0.947457	-3.612e-02	9.645e-01	5.481e-01	-0.066	
full_component_t_6_9.3 0.774348	-8.773e-02	9.160e-01	3.060e-01	-0.287	
full_component_t_6_9.4 0.068324	8.725e-01	2.393e+00	4.786e-01	1.823	
full_component_t_6_9.5 0.245530	-3.869e-01	6.792e-01	3.331e-01	-1.161	
full_component_inv_3.1 3.42e-05	-1.060e+00	3.466e-01	2.557e-01	-4.144	
full_component_inv_3.2 4.98e-05	1.048e+00	2.852e+00	2.584e-01	4.057	
full_component_inv_3.3 0.841459	7.720e-02	1.080e+00	3.860e-01	0.200	
full_component_inv_3.4 3.67e-06	1.915e+00	6.790e+00	4.138e-01	4.629	
full_component_inv_3.5 0.535809	-2.616e-01	7.698e-01	4.226e-01	-0.619	
full_component_NPM1.1 7.43e-06	5.434e-01	1.722e+00	1.213e-01	4.481	
full_component_NPM1.2 0.241128	2.466e-01	1.280e+00	2.104e-01	1.172	
full_component_NPM1.3 2.01e-06	-8.784e-01	4.155e-01	1.848e-01	-4.752	
full_component_NPM1.4 0.084179	-4.706e-01	6.247e-01	2.725e-01	-1.727	
full_component_NPM1.5 0.008059	-5.267e-01	5.905e-01	1.988e-01	-2.650	
full_component_CEBPA_bi.1 2.90e-05	5.897e-01	1.803e+00	1.410e-01	4.181	

full_component_CEBPA.bi.2 0.556414	-2.783e-01	7.571e-01	4.731e-01	-0.588
full_component_CEBPA.bi.3 0.000152	-8.942e-01	4.090e-01	2.361e-01	-3.787
full_component_CEBPA.bi.4 0.608318	-1.846e-01	8.314e-01	3.602e-01	-0.512
full_component_CEBPA.bi.5 1.10e-05	-1.272e+00	2.804e-01	2.892e-01	-4.397
full_component_TP53_complex.1 0.000189	-4.727e-01	6.233e-01	1.266e-01	-3.733
full_component_TP53_complex.2 1.06e-11	1.234e+00	3.433e+00	1.814e-01	6.799
full_component_TP53_complex.3 0.187529	2.411e-01	1.273e+00	1.830e-01	1.318
full_component_TP53_complex.4 0.003121	7.923e-01	2.208e+00	2.681e-01	2.956
full_component_TP53_complex.5 0.047825	3.841e-01	1.468e+00	1.941e-01	1.979
full_component_sAML2.1 0.209003	-1.511e-01	8.597e-01	1.203e-01	-1.256
full_component_sAML2.2 0.559549	1.079e-01	1.114e+00	1.850e-01	0.584
full_component_sAML2.3 0.172354	-2.551e-01	7.748e-01	1.870e-01	-1.365
full_component_sAML2.4 0.379444	2.429e-01	1.275e+00	2.764e-01	0.879
full_component_sAML2.5 0.071748	-3.615e-01	6.966e-01	2.008e-01	-1.801
full_component_sAML1.1 0.482646	-1.013e-01	9.036e-01	1.443e-01	-0.702
full_component_sAML1.2 0.476055	1.833e-01	1.201e+00	2.571e-01	0.713
full_component_sAML1.3 0.019140	-5.015e-01	6.056e-01	2.141e-01	-2.343
full_component_sAML1.4 0.988013	5.127e-03	1.005e+00	3.413e-01	0.015
full_component_sAML1.5 0.036669	-4.845e-01	6.160e-01	2.319e-01	-2.089
full_component_WT1.1 0.437390	-1.506e-01	8.602e-01	1.940e-01	-0.777
full_component_WT1.2 0.843233	8.614e-02	1.090e+00	4.356e-01	0.198
full_component_WT1.3 0.881818	3.922e-02	1.040e+00	2.638e-01	0.149
full_component_WT1.4 0.423059	3.961e-01	1.486e+00	4.945e-01	0.801
full_component_WT1.5 0.402044	-2.424e-01	7.847e-01	2.893e-01	-0.838
full_component_Trisomies.1 0.493121	1.178e-01	1.125e+00	1.718e-01	0.685
full_component_Trisomies.2 0.321194	-4.476e-01	6.392e-01	4.512e-01	-0.992
full_component_Trisomies.3 0.006239	-7.240e-01	4.848e-01	2.647e-01	-2.735
full_component_Trisomies.4 0.821662	9.227e-02	1.097e+00	4.093e-01	0.225

full_component_Trisomies.5 0.217752	-3.584e-01	6.988e-01	2.908e-01	-1.233
full_component_DNMT3A_IDH1_2.1 0.012873	-6.012e-01	5.481e-01	2.417e-01	-2.487
full_component_DNMT3A_IDH1_2.2 0.705188	1.374e-01	1.147e+00	3.632e-01	0.378
full_component_DNMT3A_IDH1_2.3 0.070317	-6.631e-01	5.153e-01	3.664e-01	-1.810
full_component_DNMT3A_IDH1_2.4 0.930036	-5.620e-02	9.454e-01	6.401e-01	-0.088
full_component_DNMT3A_IDH1_2.5 0.775155	-1.160e-01	8.905e-01	4.060e-01	-0.286
full_component_no_events.1 0.625117	7.998e-02	1.083e+00	1.637e-01	0.489
full_component_no_events.2 0.994951	-2.216e-03	9.978e-01	3.502e-01	-0.006
full_component_no_events.3 0.011263	-6.003e-01	5.486e-01	2.369e-01	-2.534
full_component_no_events.4 0.145869	-6.518e-01	5.211e-01	4.482e-01	-1.454
full_component_no_events.5 0.087566	-4.509e-01	6.371e-01	2.639e-01	-1.708
full_component_mNOS.1 0.628047	-6.736e-02	9.349e-01	1.390e-01	-0.484
full_component_mNOS.2 0.724837	-8.806e-02	9.157e-01	2.502e-01	-0.352
full_component_mNOS.3 0.009349	-5.411e-01	5.821e-01	2.082e-01	-2.599
full_component_mNOS.4 0.674500	1.352e-01	1.145e+00	3.219e-01	0.420
full_component_mNOS.5 0.004616	-6.412e-01	5.266e-01	2.264e-01	-2.833
ITD.1 0.350104	-4.916e-02	9.520e-01	5.262e-02	-0.934
ITD.2 0.071517	2.423e-01	1.274e+00	1.345e-01	1.802
ITD.3 1.89e-06	3.340e-01	1.397e+00	7.010e-02	4.765
ITD.4 0.992596	1.348e-03	1.001e+00	1.453e-01	0.009
ITD.5 0.000122	3.168e-01	1.373e+00	8.246e-02	3.842

Likelihood ratio test=1524 on 130 df, p=< 2.2e-16
n= 13039, number of events= 5890

II. Get coefficients for all other models to display(intermediate fixed plots in Web app).

1. Train Risk Proposal Model under cfull_proposal.

```
In [14]: form <- "Surv(Tstart, Tstop, status) ~ "
for (co in expanded_proposal){
    form <- paste (form,co,sep = " + ")
}
form <- paste(form,"+ strata(trans)")
set.seed(9)
cfull_proposal <- coxph(as.formula(form), data = msdata, method = "breslow")
cfull_proposal
```

Call:

```
coxph(formula = as.formula(form), data = msdata, method = "breslow")
```

	coef	exp(coef)	se(coef)	z	p
NEW_favorable.1	0.88387	2.42024	0.04929	17.933	< 2e-16
NEW_favorable.2	-0.49336	0.61057	0.12255	-4.026	5.68e-05
NEW_favorable.3	-0.87800	0.41561	0.06793	-12.925	< 2e-16
NEW_favorable.4	-1.21347	0.29716	0.13355	-9.086	< 2e-16
NEW_favorable.5	-0.70993	0.49168	0.08184	-8.674	< 2e-16
NEW_intermediate.1	0.46111	1.58583	0.05064	9.106	< 2e-16
NEW_intermediate.2	-0.50078	0.60606	0.10382	-4.823	1.41e-06
NEW_intermediate.3	-0.40545	0.66668	0.06691	-6.060	1.37e-09
NEW_intermediate.4	-0.60179	0.54783	0.12652	-4.756	1.97e-06
NEW_intermediate.5	-0.24678	0.78131	0.07377	-3.345	0.000822
NEW_adverse.1	NA	NA	0.00000	NA	NA
NEW_adverse.2	NA	NA	0.00000	NA	NA
NEW_adverse.3	NA	NA	0.00000	NA	NA
NEW_adverse.4	NA	NA	0.00000	NA	NA
NEW_adverse.5	NA	NA	0.00000	NA	NA

```
Likelihood ratio test=691.7 on 10 df, p=< 2.2e-16
n= 13039, number of events= 5890
```

2. Train ELN Model under cfull_eln.

```
In [15]: form <- "Surv(Tstart, Tstop, status) ~ "
for (co in expanded_eln){
    form <- paste (form,co,sep = " + ")
}
form <- paste(form,"+ strata(trans)")
set.seed(9)
cfull_eln <- coxph(as.formula(form), data = msdata, method = "breslow")
cfull_eln
```

Call:

```
coxph(formula = as.formula(form), data = msdata, method = "breslow")
```

	coef	exp(coef)	se(coef)	z	p
eln_2017_favorable.1	0.84027	2.31699	0.04683	17.941	< 2e-16
eln_2017_favorable.2	-0.30180	0.73949	0.11467	-2.632	0.00849
eln_2017_favorable.3	-0.67705	0.50812	0.06474	-10.459	< 2e-16
eln_2017_favorable.4	-1.18640	0.30532	0.12561	-9.445	< 2e-16
eln_2017_favorable.5	-0.52042	0.59427	0.07598	-6.850	7.41e-12
eln_2017_intermediate.1	0.28416	1.32865	0.05317	5.344	9.09e-08
eln_2017_intermediate.2	-0.53594	0.58512	0.10936	-4.901	9.55e-07
eln_2017_intermediate.3	-0.18636	0.82998	0.07054	-2.642	0.00824
eln_2017_intermediate.4	-0.55407	0.57461	0.13417	-4.129	3.64e-05
eln_2017_intermediate.5	-0.15903	0.85297	0.07741	-2.054	0.03993
eln_2017_adverse.1	NA	NA	0.00000	NA	NA
eln_2017_adverse.2	NA	NA	0.00000	NA	NA
eln_2017_adverse.3	NA	NA	0.00000	NA	NA
eln_2017_adverse.4	NA	NA	0.00000	NA	NA
eln_2017_adverse.5	NA	NA	0.00000	NA	NA

```
Likelihood ratio test=628.4 on 10 df, p=< 2.2e-16
n= 13039, number of events= 5890
```

3. Train Component model under cfull_comp.

```
In [16]: form <- "Surv(Tstart, Tstop, status) ~ "
for (co in c(expanded_component,expanded_itd)){
    form <- paste (form,co,sep = " + ")
}
form <- paste(form,"+ strata(trans)")
set.seed(9)
cfull_comp <- coxph(as.formula(form), data = msdata, method = "breslow")
cfull_comp
```

Call:

```
coxph(formula = as.formula(form), data = msdata, method = "breslow")
```

		coef	exp(coef)	se(coef)	z
p					
full_component_t_15_17.1	0.06547	1.115e+00	3.051e+00	6.055e-01	1.842
full_component_t_15_17.2	0.82302	-1.818e-01	8.338e-01	8.129e-01	-0.224
full_component_t_15_17.3	0.15680	-1.112e+00	3.290e-01	7.851e-01	-1.416
full_component_t_15_17.4	0.95793	-1.262e+01	3.319e-06	2.392e+02	-0.053
full_component_t_15_17.5	0.01758	-2.573e+00	7.632e-02	1.084e+00	-2.374
full_component_inv_16.1	0.03390	1.270e+00	3.562e+00	5.989e-01	2.121
full_component_inv_16.2	0.19997	-1.070e+00	3.431e-01	8.347e-01	-1.282
full_component_inv_16.3	0.65402	3.334e-01	1.396e+00	7.439e-01	0.448
full_component_inv_16.4	0.95748	-1.275e+01	2.902e-06	2.392e+02	-0.053
full_component_inv_16.5	0.00165	-3.273e+00	3.789e-02	1.040e+00	-3.146
full_component_t_8_21.1	0.07017	1.087e+00	2.964e+00	6.001e-01	1.811
full_component_t_8_21.2	0.14411	-1.208e+00	2.987e-01	8.273e-01	-1.461
full_component_t_8_21.3	0.81521	-1.750e-01	8.395e-01	7.487e-01	-0.234
full_component_t_8_21.4	0.96048	-1.185e+01	7.136e-06	2.392e+02	-0.050
full_component_t_8_21.5	0.04458	-2.086e+00	1.242e-01	1.039e+00	-2.009
full_component_t_11.1	0.21905	7.320e-01	2.079e+00	5.956e-01	1.229
full_component_t_11.2	0.20309	-9.971e-01	3.689e-01	7.834e-01	-1.273
full_component_t_11.3	0.25474	8.493e-01	2.338e+00	7.457e-01	1.139
full_component_t_11.4	0.96377	-1.086e+01	1.916e-05	2.392e+02	-0.045
full_component_t_11.5	0.05513	-1.959e+00	1.410e-01	1.021e+00	-1.918
full_component_t_6_9.1	0.52114	4.021e-01	1.495e+00	6.267e-01	0.642
full_component_t_6_9.2	0.08735	-1.544e+00	2.135e-01	9.032e-01	-1.710
full_component_t_6_9.3	0.33319	7.524e-01	2.122e+00	7.775e-01	0.968
full_component_t_6_9.4	0.96628	-1.011e+01	4.066e-05	2.392e+02	-0.042
full_component_t_6_9.5	0.04591	-2.113e+00	1.209e-01	1.058e+00	-1.996

full_component_inv_3.1 0.32654	-6.138e-01	5.413e-01	6.256e-01	-0.981
full_component_inv_3.2 0.76276	-2.221e-01	8.008e-01	7.359e-01	-0.302
full_component_inv_3.3 0.22185	1.007e+00	2.737e+00	8.242e-01	1.222
full_component_inv_3.4 0.96949	-9.148e+00	1.064e-04	2.392e+02	-0.038
full_component_inv_3.5 0.08308	-1.899e+00	1.496e-01	1.096e+00	-1.733
full_component_NPM1.1 0.09224	1.002e+00	2.723e+00	5.950e-01	1.684
full_component_NPM1.2 0.34986	-7.054e-01	4.939e-01	7.546e-01	-0.935
full_component_NPM1.3 0.83427	1.535e-01	1.166e+00	7.336e-01	0.209
full_component_NPM1.4 0.96258	-1.122e+01	1.341e-05	2.392e+02	-0.047
full_component_NPM1.5 0.04710	-2.034e+00	1.308e-01	1.025e+00	-1.985
full_component_CEBPA_bi.1 0.05970	1.138e+00	3.122e+00	6.045e-01	1.883
full_component_CEBPA_bi.2 0.10632	-1.420e+00	2.417e-01	8.793e-01	-1.615
full_component_CEBPA_bi.3 0.91931	7.634e-02	1.079e+00	7.535e-01	0.101
full_component_CEBPA_bi.4 0.96316	-1.105e+01	1.595e-05	2.392e+02	-0.046
full_component_CEBPA_bi.5 0.00518	-2.935e+00	5.314e-02	1.050e+00	-2.795
full_component_TP53_complex.1 0.97373	-1.950e-02	9.807e-01	5.922e-01	-0.033
full_component_TP53_complex.2 0.82764	1.627e-01	1.177e+00	7.473e-01	0.218
full_component_TP53_complex.3 0.09204	1.237e+00	3.445e+00	7.342e-01	1.685
full_component_TP53_complex.4 0.96697	-9.903e+00	5.002e-05	2.392e+02	-0.041
full_component_TP53_complex.5 0.28260	-1.102e+00	3.321e-01	1.026e+00	-1.074
full_component_SAML2.1 0.67439	2.496e-01	1.284e+00	5.941e-01	0.420
full_component_SAML2.2 0.32950	-7.293e-01	4.823e-01	7.479e-01	-0.975
full_component_SAML2.3 0.29125	7.744e-01	2.169e+00	7.338e-01	1.055
full_component_SAML2.4 0.96524	-1.042e+01	2.975e-05	2.392e+02	-0.044
full_component_SAML2.5 0.07857	-1.806e+00	1.643e-01	1.027e+00	-1.759
full_component_SAML1.1 0.52591	3.797e-01	1.462e+00	5.987e-01	0.634
full_component_SAML1.2 0.23681	-9.058e-01	4.042e-01	7.657e-01	-1.183
full_component_SAML1.3 0.52612	4.703e-01	1.600e+00	7.418e-01	0.634

full_component_sAML1.4 0.96409	-1.077e+01	2.106e-05	2.392e+02	-0.045
full_component_sAML1.5 0.05962	-1.942e+00	1.434e-01	1.031e+00	-1.884
full_component_WT1.1 0.46949	4.429e-01	1.557e+00	6.123e-01	0.723
full_component_WT1.2 0.14500	-1.224e+00	2.942e-01	8.396e-01	-1.457
full_component_WT1.3 0.25522	8.608e-01	2.365e+00	7.566e-01	1.138
full_component_WT1.4 0.96443	-1.066e+01	2.336e-05	2.392e+02	-0.045
full_component_WT1.5 0.06818	-1.906e+00	1.487e-01	1.045e+00	-1.824
full_component_Trisomies.1 0.31083	6.141e-01	1.848e+00	6.060e-01	1.013
full_component_Trisomies.2 0.10372	-1.383e+00	2.509e-01	8.499e-01	-1.627
full_component_Trisomies.3 0.80446	1.876e-01	1.206e+00	7.576e-01	0.248
full_component_Trisomies.4 0.96419	-1.074e+01	2.175e-05	2.392e+02	-0.045
full_component_Trisomies.5 0.07898	-1.838e+00	1.592e-01	1.046e+00	-1.757
full_component_DNMT3A_IDH1_2.1 0.87641	-9.784e-02	9.068e-01	6.291e-01	-0.156
full_component_DNMT3A_IDH1_2.2 0.30162	-8.319e-01	4.352e-01	8.054e-01	-1.033
full_component_DNMT3A_IDH1_2.3 0.68879	3.201e-01	1.377e+00	7.992e-01	0.400
full_component_DNMT3A_IDH1_2.4 0.96372	-1.088e+01	1.887e-05	2.392e+02	-0.045
full_component_DNMT3A_IDH1_2.5 0.13460	-1.621e+00	1.976e-01	1.084e+00	-1.496
full_component_no_events.1 0.33876	5.773e-01	1.781e+00	6.035e-01	0.957
full_component_no_events.2 0.15800	-1.129e+00	3.233e-01	7.997e-01	-1.412
full_component_no_events.3 0.59605	3.970e-01	1.487e+00	7.489e-01	0.530
full_component_no_events.4 0.96193	-1.142e+01	1.103e-05	2.392e+02	-0.048
full_component_no_events.5 0.05814	-1.968e+00	1.397e-01	1.039e+00	-1.895
full_component_mNOS.1 0.48770	4.146e-01	1.514e+00	5.975e-01	0.694
full_component_mNOS.2 0.12943	-1.157e+00	3.144e-01	7.630e-01	-1.516
full_component_mNOS.3 0.56949	4.209e-01	1.523e+00	7.399e-01	0.569
full_component_mNOS.4 0.96431	-1.070e+01	2.250e-05	2.392e+02	-0.045
full_component_mNOS.5 0.02942	-2.243e+00	1.061e-01	1.030e+00	-2.178
overlap.1 0.43636	-4.862e-01	6.150e-01	6.246e-01	-0.778

	Patient_Centric_Web_App			
overlap.2	1.097e+00	2.995e+00	8.033e-01	1.366
0.17208				
overlap.3	-1.060e+00	3.464e-01	8.000e-01	-1.325
0.18509				
overlap.4	1.087e+01	5.282e+04	2.392e+02	0.045
0.96373				
overlap.5	1.843e+00	6.314e+00	1.077e+00	1.710
0.08722				
ITD.1	-6.685e-02	9.353e-01	5.057e-02	-1.322
0.18624				
ITD.2	4.006e-01	1.493e+00	1.252e-01	3.201
0.00137				
ITD.3	4.252e-01	1.530e+00	6.735e-02	6.313
2.73e-10				
ITD.4	5.500e-02	1.057e+00	1.392e-01	0.395
0.69272				
ITD.5	3.114e-01	1.365e+00	7.841e-02	3.971
7.16e-05				

Likelihood ratio test=1133 on 90 df, p=< 2.2e-16
n= 13039, number of events= 5890

II) Prepare all saved multistate and probability plots

1. Create and Save one favorable Proposal, one intermediate proposal and one adverse proposal to display for average patient in specific risk proposal

The probability transitions are presaved to gain time because it does not depend on patient's covariate.

```
In [17]: set_notebook_plot_size(30,10)
for (j in c("NEW_favorable","NEW_intermediate","NEW_adverse")){
  # pdf(paste("Output_Patient/temp_plot_",j,".pdf",sep=""),width=30,height=7)
  par(mfrow=c(1,2))
  tmp_data <- rbind(msdata,rep( NA, ncol( msdata ) ))
  patA_proposal <- tmp_data[rep(nrow(tmp_data),each=5),c("NEW_favorable",
  "NEW_intermediate","NEW_adverse")]
  patA_proposal[,c("NEW_favorable","NEW_intermediate","NEW_adverse")]
<-0

  patA_proposal[,j] <- 1

  patA_proposal$trans <- 1:5
  attr(patA_proposal, "trans") <- tmat_data

  patA_proposal <- expand.covs(patA_proposal, c("NEW_favorable","NEW_intermediate",
  "NEW_adverse"), longnames = FALSE)
  patA_proposal$strata <- patA_proposal$trans
  msfA_proposal <- msfit(cfull_proposal, patA_proposal, trans = tmat_data)
  ord <- c(1,2,3,4,5,6)

  # we save the computations through ptA_proposal_favorable,intermediate and adverse
  if(j=="NEW_favorable"){
    ptA_proposal_favorable <- probtrans(msfA_proposal, predt = 0,variance=T)
  } else if(j=="NEW_intermediate"){
    ptA_proposal_intermediate <- probtrans(msfA_proposal, predt = 0,
  variance=T)
  } else {
    ptA_proposal_adverse <- probtrans(msfA_proposal, predt = 0,variance=T)
  }

  ptA_proposal <- probtrans(msfA_proposal, predt = 0,variance=T)

  proposal_patient <- j

  probtrans_plot_list <- list()
  for (i in 1:2){
    if(i==1){
      colors <- statecols[1:6]
    } else{
      colors <- statecols[2:5]
    }
    probtrans_plot_list[[i]] <- plot_probtrans_Conf_Interval(ptA_proposal,
  colors = colors, from=i, ord = ord,xlim=c(0,5.1),
  main = paste("Average Patient in the Proposal Molecular Classification : ",
  proposal_patient),
  xlab=ifelse(i==1,"Time (years)","Time from CR (years)"))
  }
}
```

```

grid.arrange(probtrans_plot_list[[1]],probtrans_plot_list[[2]],nrow=1)
ptrans1_proposal <- data.frame(States=c("Alive","Alive in CR","Alive with relapse","Death in CR","Death with Relapse","Death without CR"),
                                proba=unlist(ptA_proposal[[1]][ptA_proposal[[1]]$time]>=1,c("pstate1","pstate2","pstate3","pstate4","pstate5","pstate6"))[1,]*100,use.names = F),
                                se=unlist(ptA_proposal[[1]][ptA_proposal[[1]]$time]>=1,c("se1","se2","se3","se4","se5","se6"))[1,]*100,use.names = F)
diagnosis_1_proposal <- ggplot(ptrans1_proposal,aes(x=States,y=proba)) + geom_bar(stat="identity", fill=statecols)+ geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(proba) + crit * se / proba)),
width=.2, # Width of the error bars
position=position_dodge(.9))+ geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white", size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") + theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
ggttitle ("Average Outcome 1 year from Diagnosis")

ptrans3_proposal <- data.frame(States=c("Alive","Alive in CR","Alive with relapse","Death in CR","Death with Relapse","Death without CR"),
                                proba=unlist(ptA_proposal[[1]][ptA_proposal[[1]]$time]>=3,c("pstate1","pstate2","pstate3","pstate4","pstate5","pstate6"))[1,]*100,use.names = F),
                                se=unlist(ptA_proposal[[1]][ptA_proposal[[1]]$time]>=3,c("se1","se2","se3","se4","se5","se6"))[1,]*100,use.names = F)
diagnosis_3_proposal <- ggplot(ptrans3_proposal,aes(x=States,y=proba)) + geom_bar(stat="identity", fill=statecols)+ geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(proba) + crit * se / proba)),
width=.2, # Width of the error bars
position=position_dodge(.9))+ geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white", size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") + theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
ggttitle ("Average Outcome 3 years from Diagnosis")

ptrans1_CR_proposal <- data.frame(States=c("Alive in CR","Alive with relapse","Death in CR","Death with Relapse")),
                                proba=unlist(ptA_proposal[[2]][ptA_proposal[[2]]$time]>=1,c("pstate2","pstate3","pstate4","pstate5"))[1,]*100,use.names = F),
                                se=unlist(ptA_proposal[[2]][ptA_proposal[[2]]$time]>=1,c("se2","se3","se4","se5"))[1,]*100,use.names = F)
CR_1_proposal <- ggplot(ptrans1_CR_proposal,aes(x=States,y=proba)) + geom_bar(stat="identity", fill=statecols[2:5])+ geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(proba) + crit * se / proba)),
width=.2, # Width of the error bars
position=position_dodge(.9))+ geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white", size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") + theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
ggttitle ("Average Outcome 1 year from CR")

```

```

position=position_dodge(.9))+  

  geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white", size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +  

  theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +  

  ggtitle ("Average Outcome 1 year from CR")  
  

  ptrans3_CR_proposal <- data.frame(States=c("Alive in CR","Alive with  

relapse","Death in CR","Death with Relapse"),  

proba=unlist(ptA_proposal[[2]][ptA_proposal[[2]]$time>=3 ,c("pstate2","pstate3","pstate4","pstate5")][1,]*100,use.names = F),  

se=unlist(ptA_proposal[[2]][ptA_proposal[[2]]$time>=3 ,c("se2","se3","se4","se5")][1,]*100,use.names = F))  

CR_3_proposal <- ggplot(ptrans3_CR_proposal,aes(x=States,y=proba)) +  

geom_bar(stat="identity", fill=statecols[2:5]) +  

geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp  

(log(proba) + crit * se / proba)),  

width=.2, # Width of the error  

bars  

position=position_dodge(.9))+  

  geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white", size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +  

  theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20,face="bold")) +  

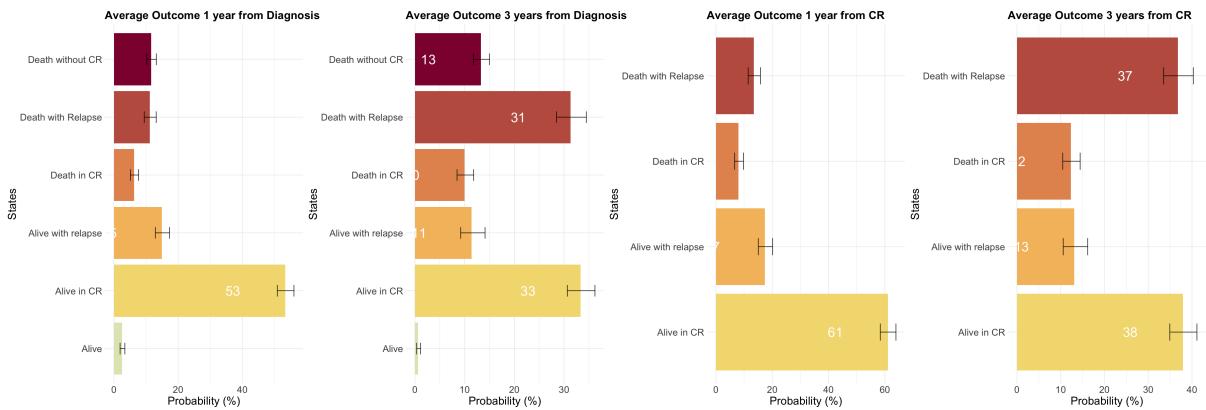
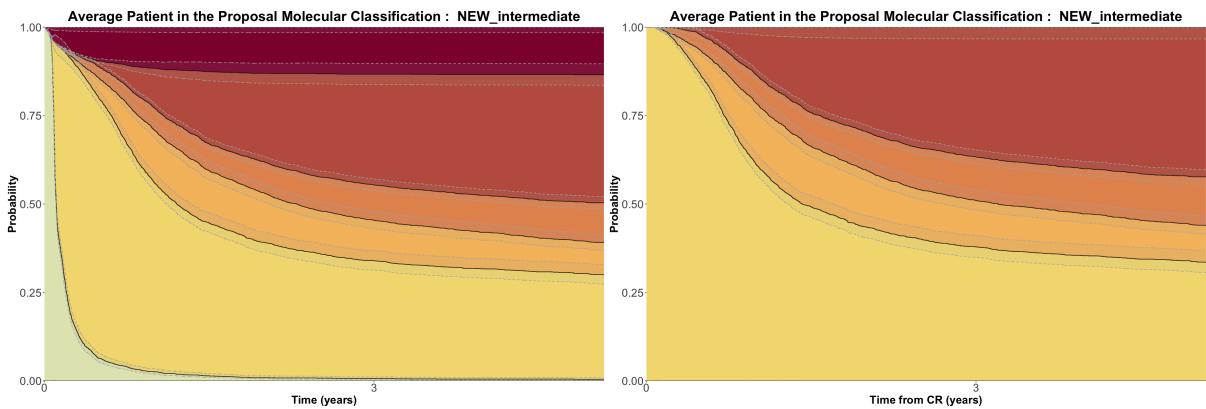
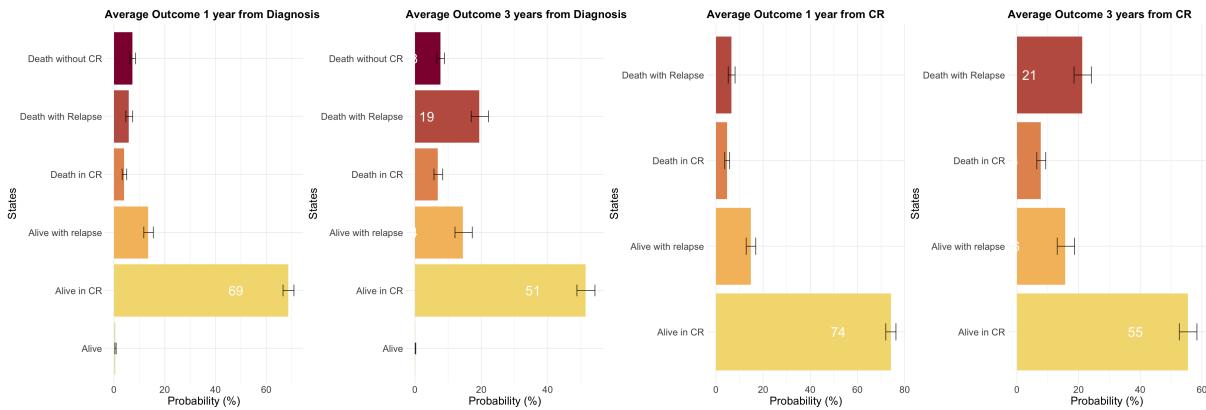
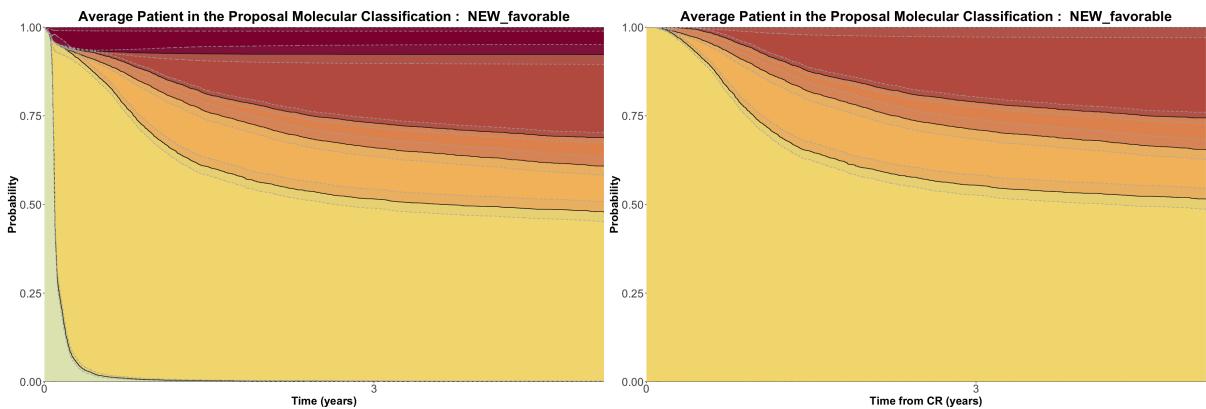
  ggtitle ("Average Outcome 3 years from CR")  
  

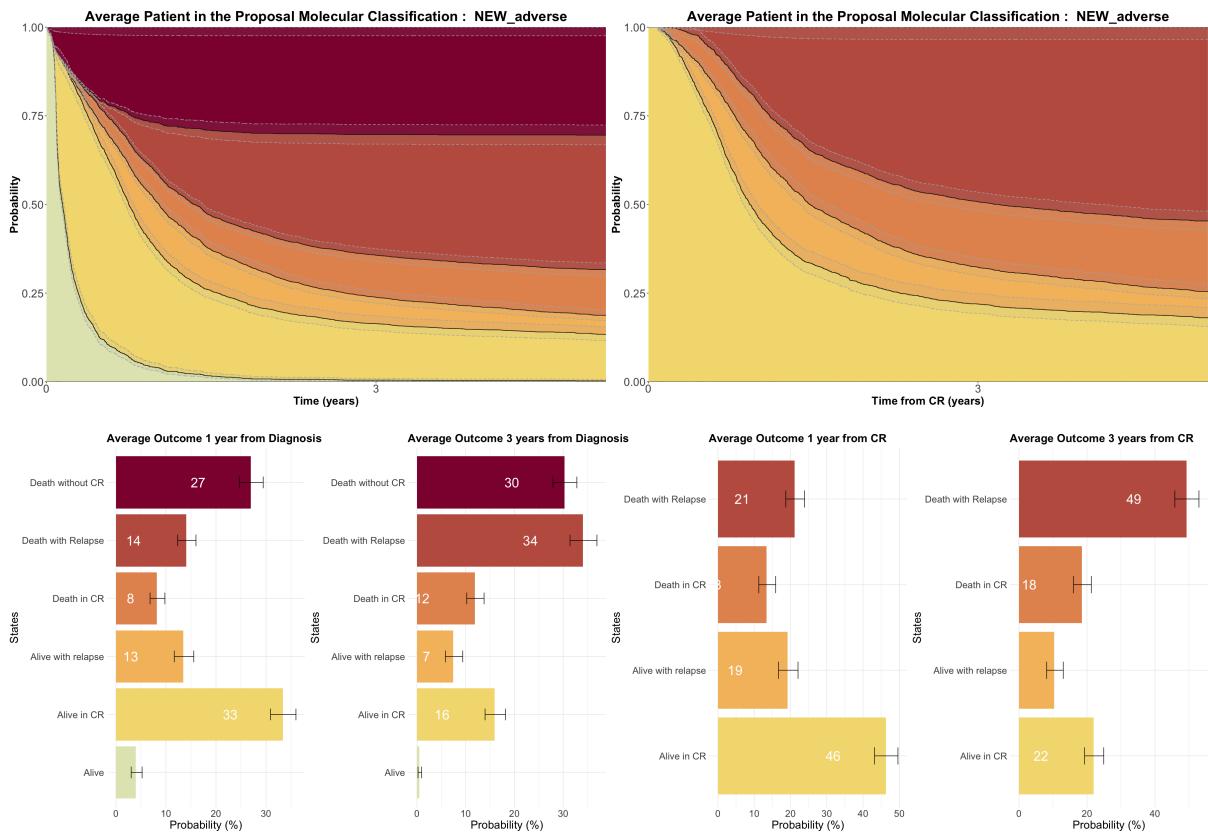
grid.arrange(diagnosis_1_proposal,diagnosis_3_proposal,CR_1_proposal  

,CR_3_proposal,nrow=1)  

# dev.off()
}

```





2. Create and Save one favorable ELN, one intermediate ELN and one adverse ELN to display for average patient in specific risk ELN

The probability transitions are preserved to gain time because it does not depend on patient's covariate.

```
In [18]: for (j in c("eln_2017_favorable", "eln_2017_intermediate", "eln_2017_adverse")){
  # pdf(paste("Output_Patient/temp_plot_", j, ".pdf", sep=""), width=30, height=7)
  par(mfrow=c(1,2))
  tmp_data <- rbind(msdata, rep( NA, ncol( msdata ) ))
  patA_eln <- tmp_data[rep(nrow(tmp_data)), each=5],
  c("eln_2017_favorable", "eln_2017_intermediate",
    "eln_2017_adverse")]
  patA_eln[,c("eln_2017_favorable", "eln_2017_intermediate", "eln_2017_adverse")] <- 0

  patA_eln[,j] <- 1

  patA_eln$trans <- 1:5
  attr(patA_eln, "trans") <- tmat_data

  patA_eln <- expand.covs(patA_eln,
    c("eln_2017_favorable", "eln_2017_intermediate",
      "eln_2017_adverse"),
    longnames = FALSE)
  patA_eln$strata <- patA_eln$trans
  msfA_eln <- msfit(cfull_eln, patA_eln, trans = tmat_data)
  ord <- c(1,2,3,4,5,6)

  # we save the computations through ptA_eln_favorable, intermediate and adverse
  if(j=="eln_2017_favorable"){
    ptA_eln_favorable <- probtrans(msfA_eln, predt = 0, variance=T)
  } else if(j=="eln_2017_intermediate"){
    ptA_eln_intermediate <- probtrans(msfA_eln, predt = 0, variance=T)
  }
  } else {
    ptA_eln_adverse <- probtrans(msfA_eln, predt = 0, variance=T)
  }

  ptA_eln <- probtrans(msfA_eln, predt = 0, variance=T)

  eln_patient <- j

  probtrans_plot_list <- list()
  for (i in 1:2){
    if(i==1){
      colors <- statecols[1:6]
    } else{
      colors <- statecols[2:5]
    }
    probtrans_plot_list[[i]] <- plot_probtrans_Conf_Interval(ptA_eln,
      colors = colors, from=i, ord = ord, xlim=c(0,5.1),
      main = paste("Average Patient in the ELN Molecular Classification : ", eln_patient),
      xlabel=ifelse(i==1, "Time (years)", "Time from CR (years)"))
  }
}
```

```

grid.arrange(probtrans_plot_list[[1]],probtrans_plot_list[[2]],nrow=1)
ptrans1_eln <- data.frame(States=c("Alive","Alive in CR","Alive with relapse","Death in CR","Death with Relapse","Death without CR"),
                           proba=unlist(ptA_eln[[1]][ptA_eln[[1]]$time>=1 ,c("pst atel","pstate2","pstate3","pstate4","pstate5","pstate6")][1,]*100,use.names = F),
                           se=unlist(ptA_eln[[1]][ptA_eln[[1]]$time>=1 ,c("se1","se2","se3","se4","se5","se6")][1,]*100,use.names = F))
diagnosis_1_eln <- ggplot(ptrans1_eln,aes(x=States,y=proba)) + geom_bar(stat="identity", fill=statecols)+geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(proba) + crit * se / proba)),
width=.2, # Width of the error
bars
position=position_dodge(.9))+geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white", size=8.5,face="bold")+coord_flip() +ylab("Probability (%)")+theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
ggtitle ("Average Outcome 1 year from Diagnosis")

ptrans3_eln <- data.frame(States=c("Alive","Alive in CR","Alive with relapse","Death in CR","Death with Relapse","Death without CR"),
                           proba=unlist(ptA_eln[[1]][ptA_eln[[1]]$time>=3 ,c("pst atel","pstate2","pstate3","pstate4","pstate5","pstate6")][1,]*100,use.names = F),
                           se=unlist(ptA_eln[[1]][ptA_eln[[1]]$time>=3 ,c("se1","se2","se3","se4","se5","se6")][1,]*100,use.names = F))
diagnosis_3_eln <- ggplot(ptrans3_eln,aes(x=States,y=proba)) + geom_bar(stat="identity", fill=statecols)+geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(proba) + crit * se / proba)),
width=.2, # Width of the error
bars
position=position_dodge(.9))+geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white", size=8.5,face="bold")+coord_flip() +ylab("Probability (%)")+theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
ggtitle ("Average Outcome 3 years from Diagnosis")

ptrans1_CR_eln <- data.frame(States=c("Alive in CR","Alive with relapse","Death in CR","Death with Relapse"),
                               proba=unlist(ptA_eln[[2]][ptA_eln[[2]]$time>=1 ,c("pst atel","pstate3","pstate4","pstate5")][1,]*100,use.names = F),
                               se=unlist(ptA_eln[[2]][ptA_eln[[2]]$time>=1 ,c("se2","se3","se4","se5")][1,]*100,use.names = F))
CR_1_eln <- ggplot(ptrans1_CR_eln,aes(x=States,y=proba)) + geom_bar(stat="identity", fill=statecols[2:5])+geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(proba) + crit * se / proba)),
width=.2, # Width of the error
bars
position=position_dodge(.9))+geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",

```

```

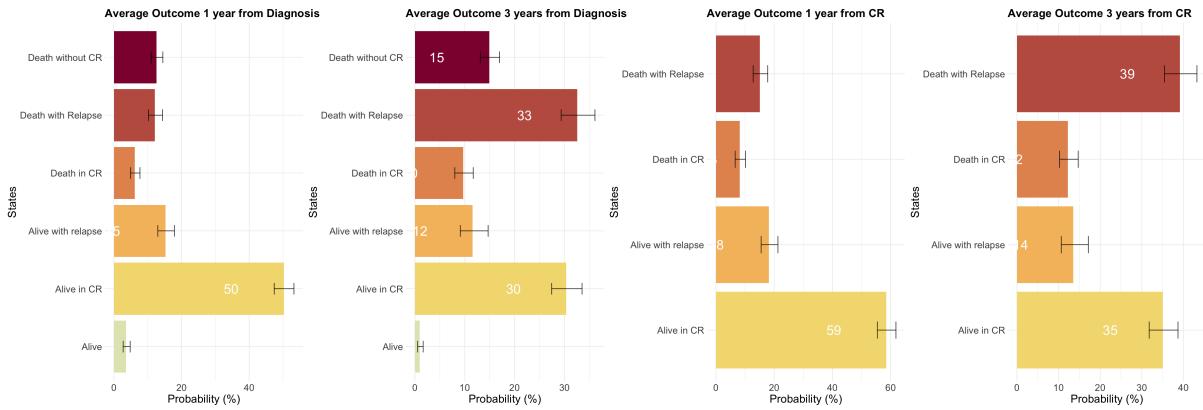
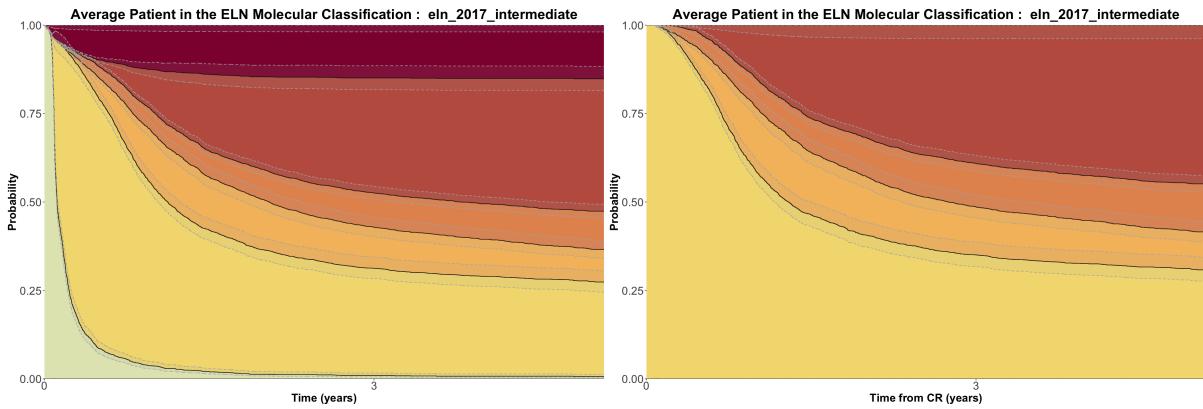
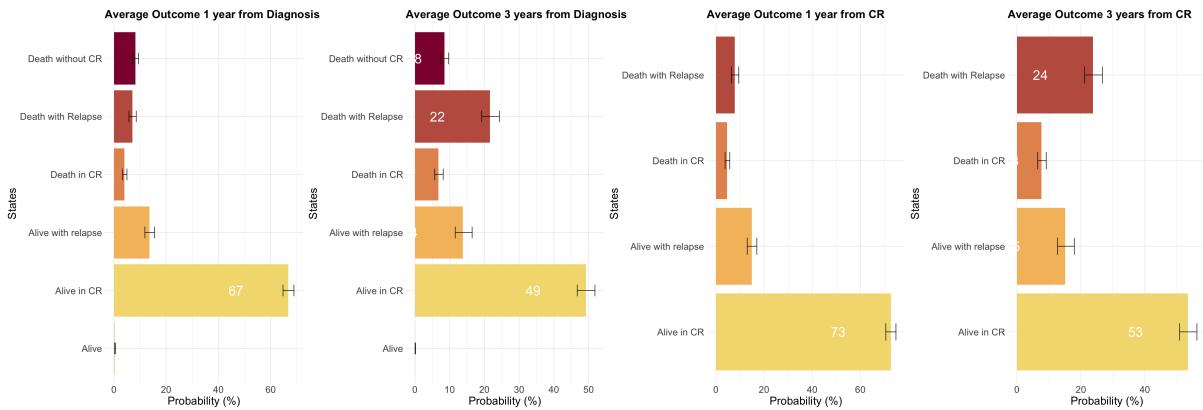
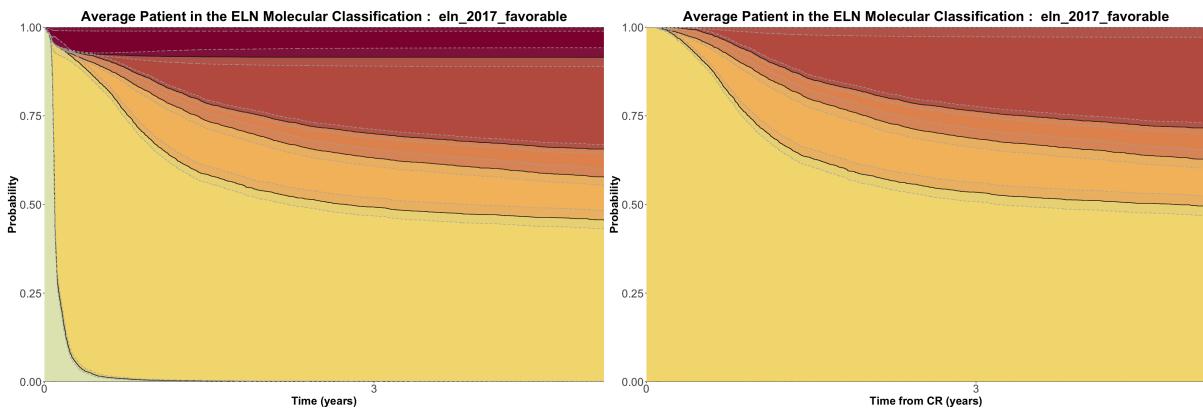
    te", size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +
      theme_minimal() + theme(axis.text=element_text(size=17),axis.title=
      element_text(size=20),plot.title=element_text(size=20,face="bold")) +
      ggtitle ("Average Outcome 1 year from CR")

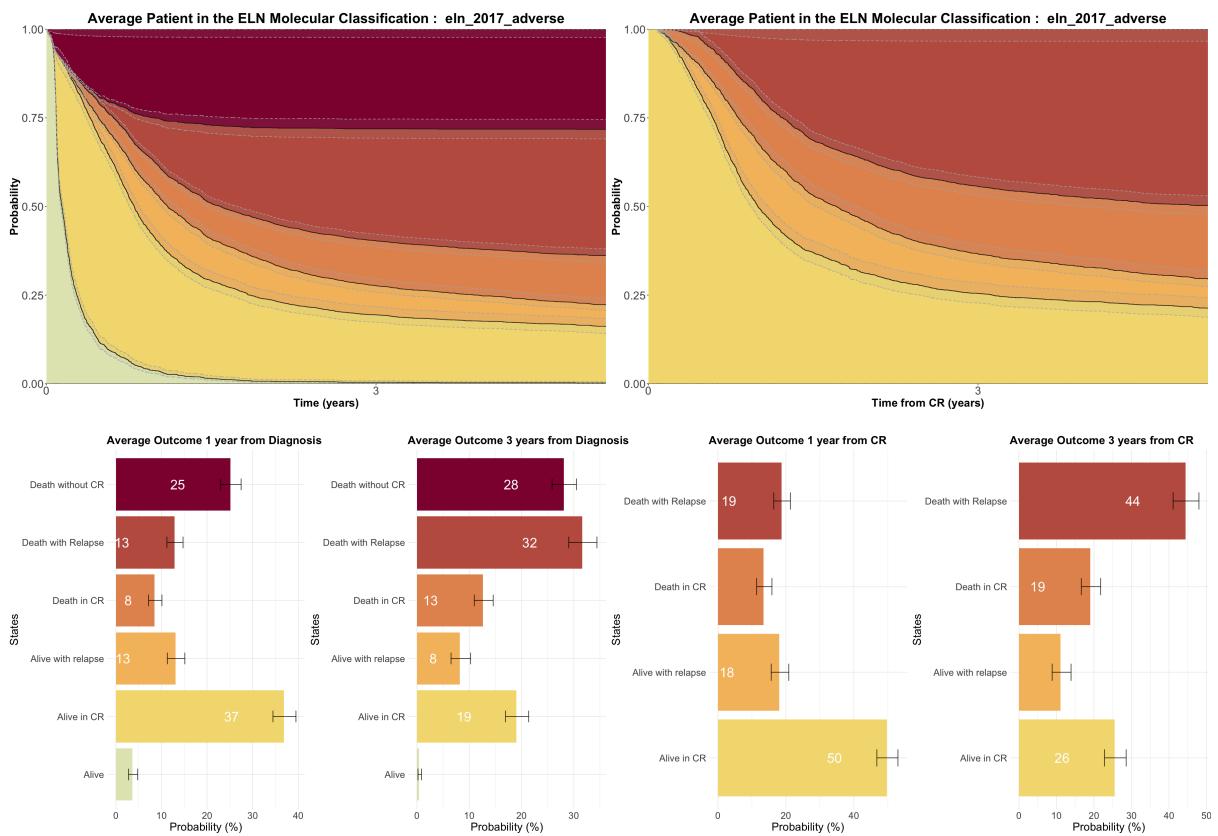
      ptrans3_CR_eln <- data.frame(States=c("Alive in CR","Alive with rela-
      pse","Death in CR","Death with Relapse"),
      proba=unlist(ptA_eln[[2]][ptA_eln[[2]]$time>=3 ,c("pst-
      ate2","pstate3","pstate4","pstate5")][1,]*100,use.names = F),
      se=unlist(ptA_eln[[2]][ptA_eln[[2]]$time>=3 ,c("se2",
      "se3","se4","se5")][1,]*100,use.names = F) )
      CR_3_eln <- ggplot(ptrans3_CR_eln,aes(x=States,y=proba)) + geom_bar(
      stat="identity", fill=statecols[2:5])+
      geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(
      log(proba) + crit * se / proba)),
      width=.2, # Width of the error
      bars
      position=position_dodge(.9))+

      geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",
      size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +
      theme_minimal() + theme(axis.text=element_text(size=17),axis.title=
      element_text(size=20),plot.title=element_text(size=20,face="bold")) +
      ggtitle ("Average Outcome 3 years from CR")

      grid.arrange(diagnosis_1_eln,diagnosis_3_eln,CR_1_eln,CR_3_eln,nrow=
      1)
      # dev.off()
}

```





3. Create and Save 2 plots for each class: one when ITD mutation is present and one when it's not present.

The probability transitions are preserved to gain time because it does not depend on patient's covariate.

```
In [19]: # ITD AND WITHOUT ITD
for (j in c(comp[comp!="overlap"])){
  tmp_data <- rbind(msdata,rep( NA, ncol( msdata ) ))
  # one version with ITD and one without
  patA_comp <- tmp_data[rep(nrow(tmp_data),each=5),
    c(comp,"ITD")]
  patA_comp[,c(comp,"ITD")] <- 0
  # Copy patA_comp for ITD
  patA_comp_ITD <- patA_comp

  # Prepare patA_comp
  patA_comp[,j] <- 1
  patA_comp$trans <- 1:5
  attr(patA_comp, "trans") <- tmat_data
  patA_comp <- expand.covs(patA_comp,
    c(comp,"ITD"),
    longnames = FALSE)
  patA_comp$strata <- patA_comp$trans
  msfA_comp <- msfit(cfull_comp, patA_comp, trans = tmat_data)

  # Add ITD Patient from patA
  patA_comp_ITD[,j] <- 1
  patA_comp_ITD$ITD <- 1
  patA_comp_ITD$trans <- 1:5
  attr(patA_comp_ITD, "trans") <- tmat_data
  patA_comp_ITD <- expand.covs(patA_comp_ITD,
    c(comp,"ITD"),
    longnames = FALSE)
  patA_comp_ITD$strata <- patA_comp_ITD$trans
  msfA_comp_ITD <- msfit(cfull_comp, patA_comp_ITD, trans = tmat_data)

  ord <- c(1,2,3,4,5,6)

  # we save the computations for each component and component+ITD

  if(j=="full_component_t_15_17"){
    ptA_full_component_t_15_17 <- probtrans(msfA_comp, predt = 0,variance=T)
    ptA_full_component_t_15_17_ITD <- probtrans(msfA_comp_ITD, predt = 0,variance=T)
  } else if(j=="full_component_inv_16"){
    ptA_full_component_inv_16 <- probtrans(msfA_comp, predt = 0,variance=T)
    ptA_full_component_inv_16_ITD <- probtrans(msfA_comp_ITD, predt = 0,variance=T)
  } else if(j=="full_component_t_8_21"){

}
```

```

        ptA_full_component_t_8_21 <- probtrans(msfA_comp, predt = 0, variance=T)
        ptA_full_component_t_8_21_ITD <- probtrans(msfA_comp_ITD, predt = 0, variance=T)
    } else if(j=="full_component_t_11"){
        ptA_full_component_t_11 <- probtrans(msfA_comp, predt = 0, variance=T)
        ptA_full_component_t_11_ITD <- probtrans(msfA_comp_ITD, predt = 0, variance=T)
    } else if(j=="full_component_t_6_9"){
        ptA_full_component_t_6_9 <- probtrans(msfA_comp, predt = 0, variance=T)
        ptA_full_component_t_6_9_ITD <- probtrans(msfA_comp_ITD, predt = 0, variance=T)
    } else if(j=="full_component_inv_3"){
        ptA_full_component_inv_3 <- probtrans(msfA_comp, predt = 0, variance=T)
        ptA_full_component_inv_3_ITD <- probtrans(msfA_comp_ITD, predt = 0, variance=T)
    } else if(j=="full_component_Trisomies"){
        ptA_full_component_Trisomies <- probtrans(msfA_comp, predt = 0, variance=T)
        ptA_full_component_Trisomies_ITD <- probtrans(msfA_comp_ITD, predt = 0, variance=T)
    } else if(j=="full_component_TP53_complex"){
        ptA_full_component_TP53_complex <- probtrans(msfA_comp, predt = 0, variance=T)
        ptA_full_component_TP53_complex_ITD <- probtrans(msfA_comp_ITD, predt = 0, variance=T)
    } else if(j=="full_component_NPM1"){
        ptA_full_component_NPM1 <- probtrans(msfA_comp, predt = 0, variance=T)
        ptA_full_component_NPM1_ITD <- probtrans(msfA_comp_ITD, predt = 0, variance=T)
    } else if(j=="full_component_CEBPA_bi"){
        ptA_full_component_CEBPA_bi <- probtrans(msfA_comp, predt = 0, variance=T)
        ptA_full_component_CEBPA_bi_ITD <- probtrans(msfA_comp_ITD, predt = 0, variance=T)
    } else if(j=="full_component_DNMT3A_IDH1_2"){
        ptA_full_component_DNMT3A_IDH1_2 <- probtrans(msfA_comp, predt = 0, variance=T)
        ptA_full_component_DNMT3A_IDH1_2_ITD <- probtrans(msfA_comp_ITD, predt = 0, variance=T)
    } else if(j=="full_component_WT1"){
        ptA_full_component_WT1 <- probtrans(msfA_comp, predt = 0, variance=T)
        ptA_full_component_WT1_ITD <- probtrans(msfA_comp_ITD, predt = 0, variance=T)
    } else if(j=="full_component_sAML2"){
        ptA_full_component_sAML2 <- probtrans(msfA_comp, predt = 0, variance=T)
        ptA_full_component_sAML2_ITD <- probtrans(msfA_comp_ITD, predt = 0, variance=T)
    } else if(j=="full_component_sAML1"){
        ptA_full_component_sAML1 <- probtrans(msfA_comp, predt = 0, variance=T)
    }
}

```

```

nce=T)
    ptA_full_component_sAML1_ITD <- probtrans(msfA_comp_ITD, predt =
0,variance=T)
} else if(j=="full_component_mNOS"){
    ptA_full_component_mNOS <- probtrans(msfA_comp, predt = 0,varian-
ce=T)
    ptA_full_component_mNOS_ITD <- probtrans(msfA_comp_ITD, predt =
0,variance=T)
} else if(j=="full_component_no_events"){
    ptA_full_component_no_events <- probtrans(msfA_comp, predt = 0,v-
ariance=T)
    ptA_full_component_no_events_ITD <- probtrans(msfA_comp, predt =
0,variance=T) # can not have ITD
}

ptA_comp <- probtrans(msfA_comp, predt = 0,variance=T)
ptA_comp_ITD <- probtrans(msfA_comp_ITD, predt = 0,variance=T)

comp_patient <- str_remove(j,"full_component_")
comp_patient_ITD <- paste(str_remove(j,"full_component_"),"+ ITD")
for (k in 1:2){
    if(k==1){
        patient <- ptA_comp
        name <- comp_patient
        pdf_name <- paste("Output_Patient/temp_plot_",j,".pdf",sep=""
)
    } else{
        if(j=="full_component_no_events"){ ## special case because
no ITD anyway if component is no events
            patient <- ptA_comp
            name <- comp_patient
            pdf_name <- paste("Output_Patient/temp_plot_",j,".pdf",se-
p="")
        }else{
            patient <- ptA_comp_ITD
            name <- comp_patient_ITD
            pdf_name <- paste("Output_Patient/temp_plot_",j,"_ITD.pdf",
sep="")
        }
    }
    # pdf(pdf_name,width=30,height=7)
    probtrans_plot_list <- list()
    for (i in 1:2){
        if(i==1){
            colors <- statecols[1:6]
        } else{
            colors <- statecols[2:5]
        }
        probtrans_plot_list[[i]] <- plot_probtrans_Conf_Interval(pat-
ient,colors = colors, from=i, ord = ord,xlim=c(0,5.1),
main = paste("Average Patient i-
n Component : ",name),
"Time from CR (years)"))
    }
}

```

```

grid.arrange(probtrans_plot_list[[1]],probtrans_plot_list[[2]],n
row=1)

  if(k==1){
    ptrans1_comp <- data.frame(States=c("Alive","Alive in CR","A
live with relapse","Death in CR","Death with Relapse","Death without CR"
),
    proba=unlist(patient[[1]][patient[[1]]$time>=1 ,c("pst
ate1","pstate2","pstate3","pstate4","pstate5","pstate6")][1,]*100,use.na
mes = F),
    se=unlist(patient[[1]][patient[[1]]$time>=1 ,c("se1",
"se2","se3","se4","se5","se6")][1,]*100,use.names = F) )
    diagnosis_1_comp <- ggplot(ptrans1_comp,aes(x=States,y=proba
)) + geom_bar(stat="identity", fill=statecols)+
      geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba),
ymax=exp(log(proba) + crit * se / proba)),
width=.2, # Width of th
e error bars
      position=position_dodge(.9))+
      geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, co
lor="white", size=8.5,face="bold")+ coord_flip()+ylab("Probability (%)")
+
      theme_minimal() + theme(axis.text=element_text(size=17),ax
is.title=element_text(size=20),plot.title=element_text(size=20,face="bol
d")) +
      ggttitle ("Average Outcome 1 year from Diagnosis")

    ptrans3_comp <- data.frame(States=c("Alive","Alive in CR","A
live with relapse","Death in CR","Death with Relapse","Death without CR"
),
    proba=unlist(patient[[1]][patient[[1]]$time>=3 ,c(
"pstate1","pstate2","pstate3","pstate4","pstate5","pstate6")][1,]*100
,use.names = F),
    se=unlist(patient[[1]][patient[[1]]$time>=3 ,c(
"se1","se2","se3","se4","se5","se6")][1,]*100,use.names = F)
)
    diagnosis_3_comp <- ggplot(ptrans3_comp,aes(x=States,y=proba
)) + geom_bar(stat="identity", fill=statecols)+
      geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba),
ymax=exp(log(proba) + crit * se / proba)),
width=.2, # Width of th
e error bars
      position=position_dodge(.9))+
      geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, co
lor="white", size=8.5,face="bold")+ coord_flip()+ylab("Probability (%)")
+
      theme_minimal() + theme(axis.text=element_text(size=17),ax
is.title=element_text(size=20),plot.title=element_text(size=20,face="bol
d")) +
      ggttitle ("Average Outcome 3 years from Diagnosis")

    ptrans1_CR_comp <- data.frame(States=c("Alive in CR","Alive
with relapse","Death in CR","Death with Relapse"),
    proba=unlist(patient[[2]][patient[[2]]$time>=1
,c("pstate2","pstate3","pstate4","pstate5")][1,]*100,use.names = F),
    se=unlist(patient[[2]][patient[[2]]$time>=1 ,c

```

```

("se2", "se3", "se4", "se5")][1,]*100,use.names = F) )
CR_1_comp <- ggplot(ptrans1_CR_comp,aes(x=States,y=proba)) +
geom_bar(stat="identity", fill=statecols[2:5])+
  geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba),
  ymax=exp(log(proba) + crit * se / proba)),
  width=.2,                                     # Width of the
e error bars
  position=position_dodge(.9))++
  geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, co
lor="white", size=8.5,face="bold")+ coord_flip() +ylab("Probability (%)")
+
  theme_minimal() + theme(axis.text=element_text(size=17),ax
is.title=element_text(size=20),plot.title=element_text(size=20,face="bol
d")) +
  ggtitle ("Average Outcome 1 year from CR")

ptrans3_CR_comp <- data.frame(States=c("Alive in CR","Alive
with relapse", "Death in CR", "Death with Relapse"),
  proba=unlist(patient[[2]][patient[[2]]$time>=3
,c("pstate2", "pstate3", "pstate4", "pstate5")][1,]*100,use.names = F),
  se=unlist(patient[[2]][patient[[2]]$time>=3 ,c
("se2", "se3", "se4", "se5")][1,]*100,use.names = F))
CR_3_comp <- ggplot(ptrans3_CR_comp,aes(x=States,y=proba)) +
geom_bar(stat="identity", fill=statecols[2:5])+
  geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba),
  ymax=exp(log(proba) + crit * se / proba)),
  width=.2,                                     # Width of the
e error bars
  position=position_dodge(.9))++
  geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, co
lor="white", size=8.5,face="bold")+ coord_flip() +ylab("Probability (%)")
+
  theme_minimal() + theme(axis.text=element_text(size=17),ax
is.title=element_text(size=20),plot.title=element_text(size=20,face="bol
d")) +
  ggtitle ("Average Outcome 3 years from CR")

grid.arrange(diagnosis_1_comp,diagnosis_3_comp,CR_1_comp,CR_
3_comp,nrow=1)
}else{
  ptrans1_comp_ITD <- data.frame(States=c("Alive", "Alive in C
R", "Alive with relapse", "Death in CR", "Death with Relapse", "Death withou
t CR"),
  proba=unlist(patient[[1]][patient[[1]]$time>=1 ,c("pst
ate1", "pstate2", "pstate3", "pstate4", "pstate5", "pstate6")][1,]*100,use.na
mes = F),
  se=unlist(patient[[1]][patient[[1]]$time>=1 ,c("se1",
"se2", "se3", "se4", "se5", "se6")][1,]*100,use.names = F))
  diagnosis_1_comp_ITD <- ggplot(ptrans1_comp_ITD,aes(x=States
,y=proba)) + geom_bar(stat="identity", fill=statecols)++
  geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba),
  ymax=exp(log(proba) + crit * se / proba)),
  width=.2,                                     # Width of the
e error bars
  position=position_dodge(.9))++

```

```

geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white", size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +
  theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
  ggtitle ("Average Outcome 1 year from Diagnosis")

ptrans3_comp_ITD <- data.frame(States=c("Alive","Alive in CR","Alive with relapse","Death in CR","Death with Relapse","Death without CR"),
  proba=unlist(patient[[1]][patient[[1]]$time>=3,c("pstate1","pstate2","pstate3","pstate4","pstate5","pstate6")][1,]*100,use.names = F),
  se=unlist(patient[[1]][patient[[1]]$time>=3,c("se1","se2","se3","se4","se5","se6")][1,]*100,use.names = F)
)
diagnosis_3_comp_ITD <- ggplot(ptrans3_comp_ITD,aes(x=States,y=proba)) + geom_bar(stat="identity", fill=statecols) +
  geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(proba) + crit * se / proba)),
  width=.2, # Width of the error bars
  position=position_dodge(.9)) +
  geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white", size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +
  theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
  ggtitle ("Average Outcome 3 years from Diagnosis")

ptrans1_CR_comp_ITD <- data.frame(States=c("Alive in CR","Alive with relapse","Death in CR","Death with Relapse"),
  proba=unlist(patient[[2]][patient[[2]]$time>=1,c("pstate2","pstate3","pstate4","pstate5")][1,]*100,use.names = F),
  se=unlist(patient[[2]][patient[[2]]$time>=1,c("se2","se3","se4","se5")][1,]*100,use.names = F))
CR_1_comp_ITD <- ggplot(ptrans1_CR_comp_ITD,aes(x=States,y=proba)) + geom_bar(stat="identity", fill=statecols[2:5]) +
  geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(proba) + crit * se / proba)),
  width=.2, # Width of the error bars
  position=position_dodge(.9)) +
  geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white", size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +
  theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
  ggtitle ("Average Outcome 1 year from CR")

ptrans3_CR_comp_ITD <- data.frame(States=c("Alive in CR","Alive with relapse","Death in CR","Death with Relapse"),
  proba=unlist(patient[[2]][patient[[2]]$time>=3

```

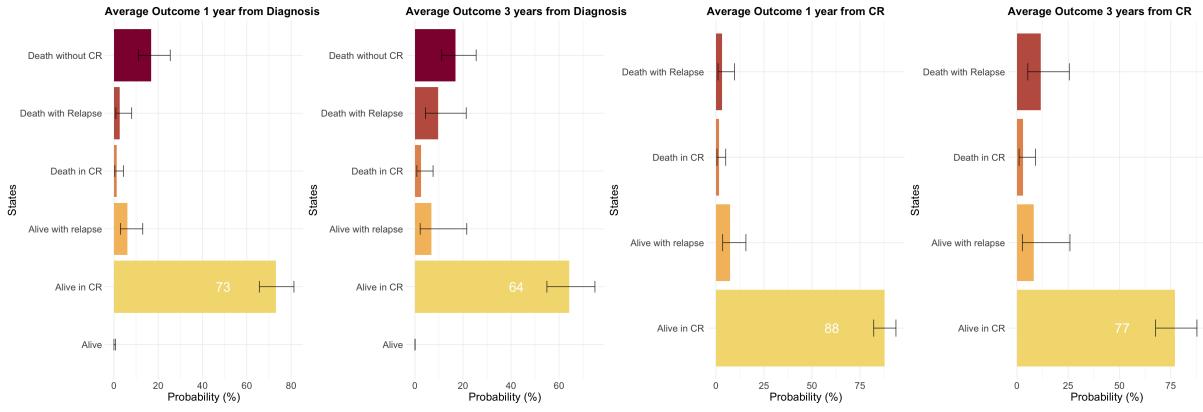
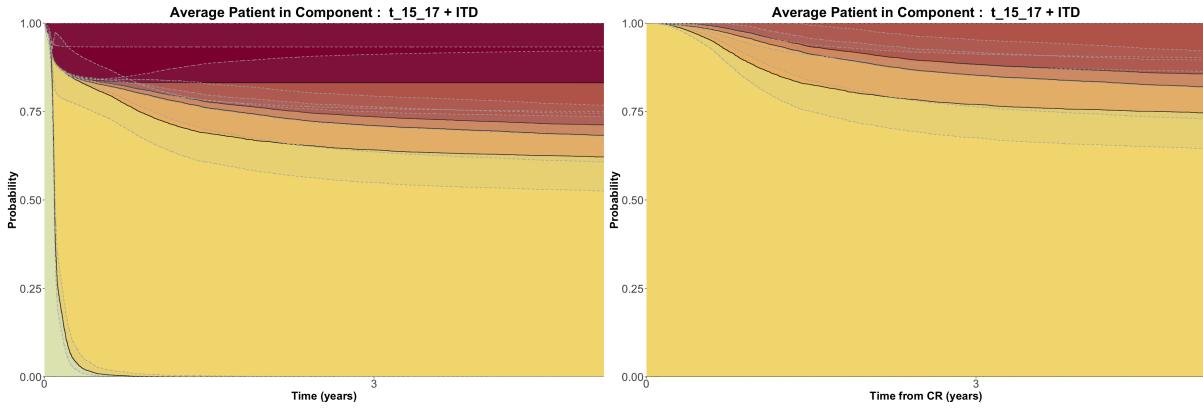
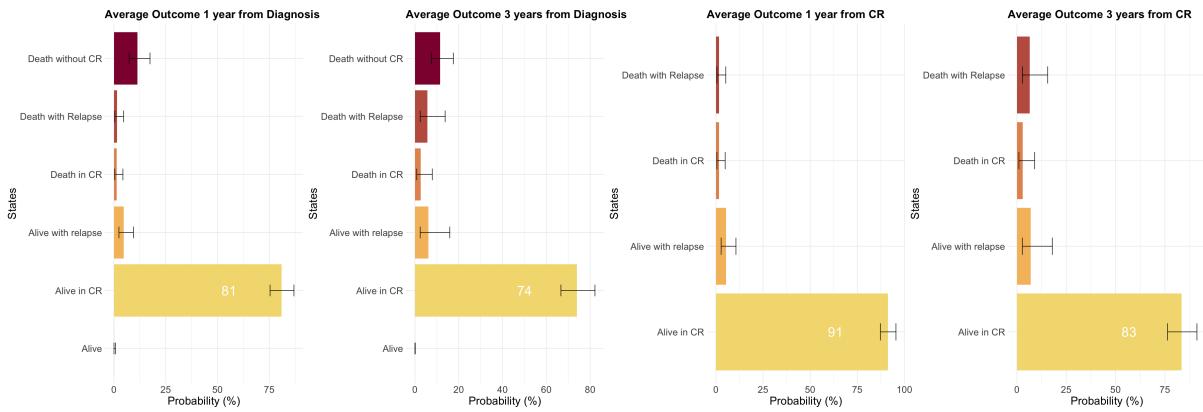
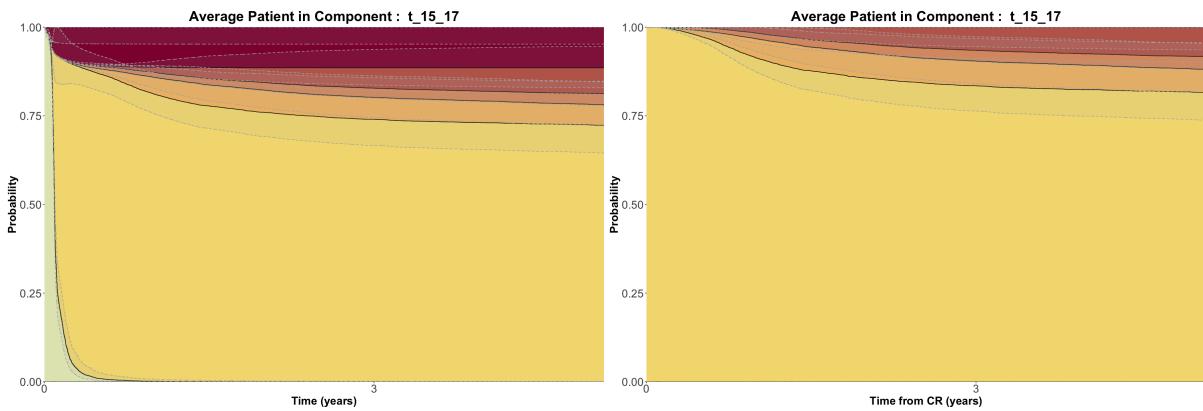
```

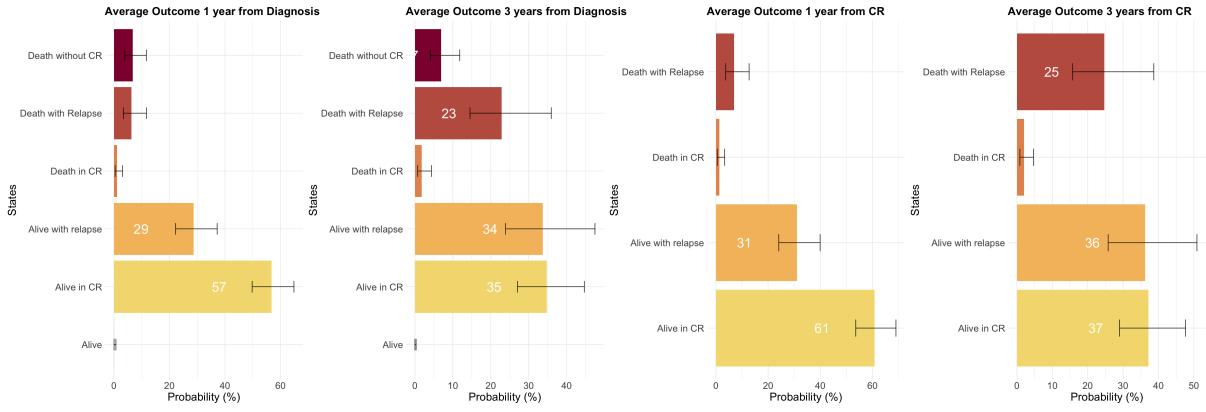
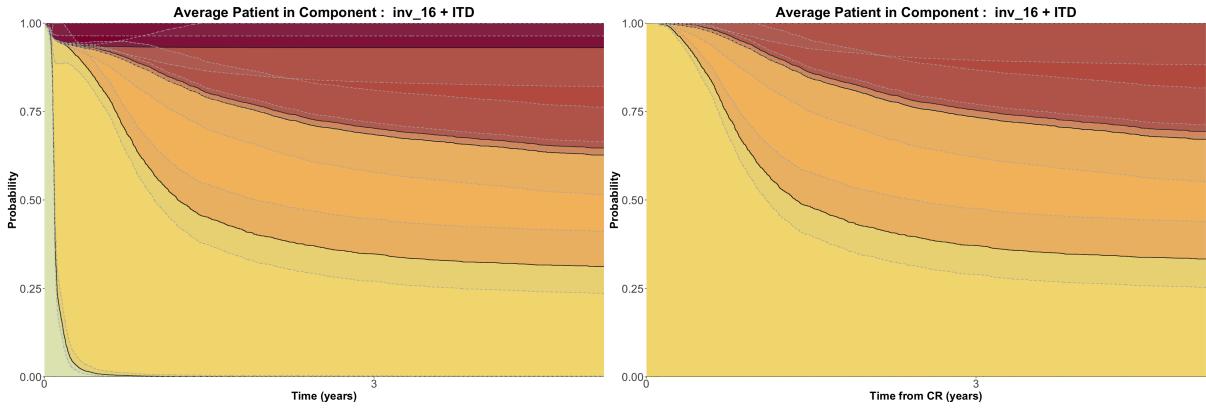
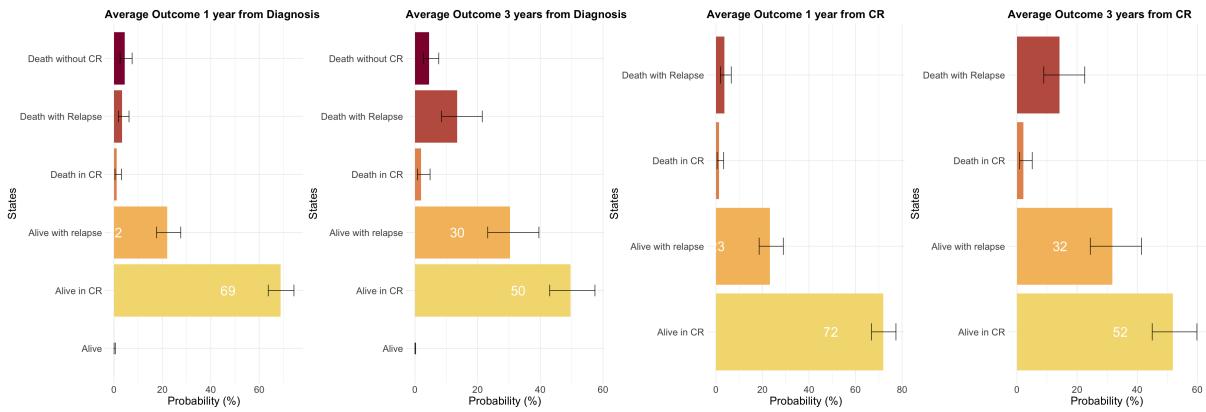
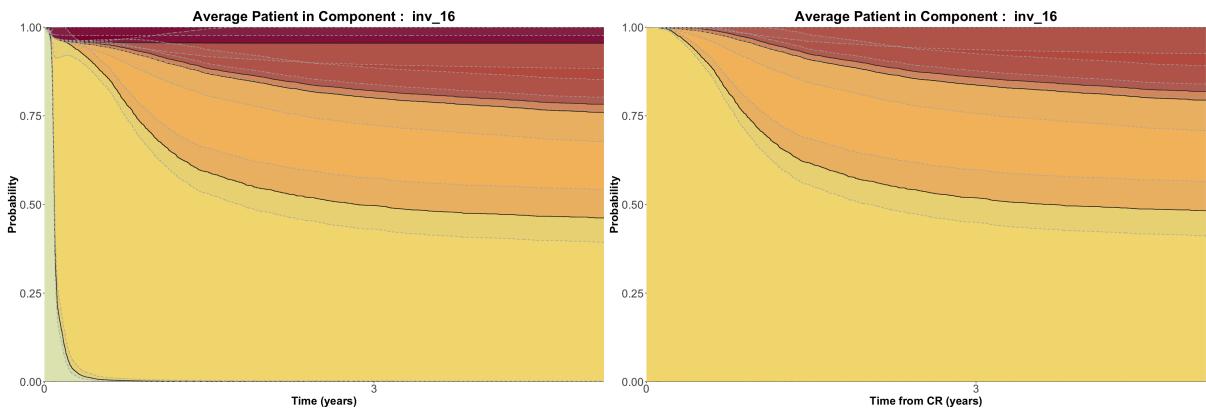
, c("pstate2", "pstate3", "pstate4", "pstate5"))[1,]*100,use.names = F),
                     se=unlist(patient[[2]][patient[[2]]$time>=3 ,c
("se2", "se3", "se4", "se5"))[1,]*100,use.names = F) )
CR_3_comp_ITD <- ggplot(ptrans3_CR_comp_ITD,aes(x=States,y=proba)) + geom_bar(stat="identity", fill=statecols[2:5])+
  geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba),
  ymax=exp(log(proba) + crit * se / proba)),
  width=.2, # Width of the error bars
  position=position_dodge(.9))+geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white", size=8.5,face="bold") + coord_flip() + ylab("Probability (%)")+
  theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
  ggtitle ("Average Outcome 3 years from CR")

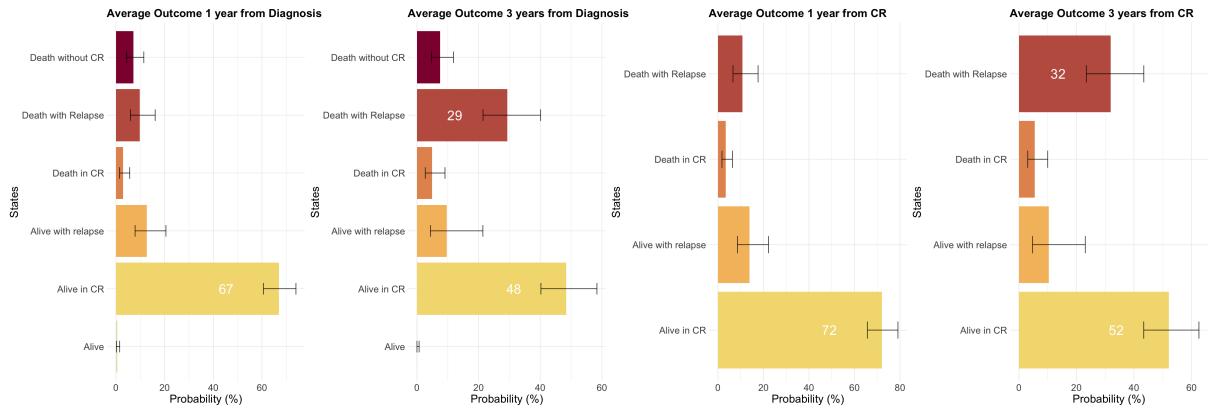
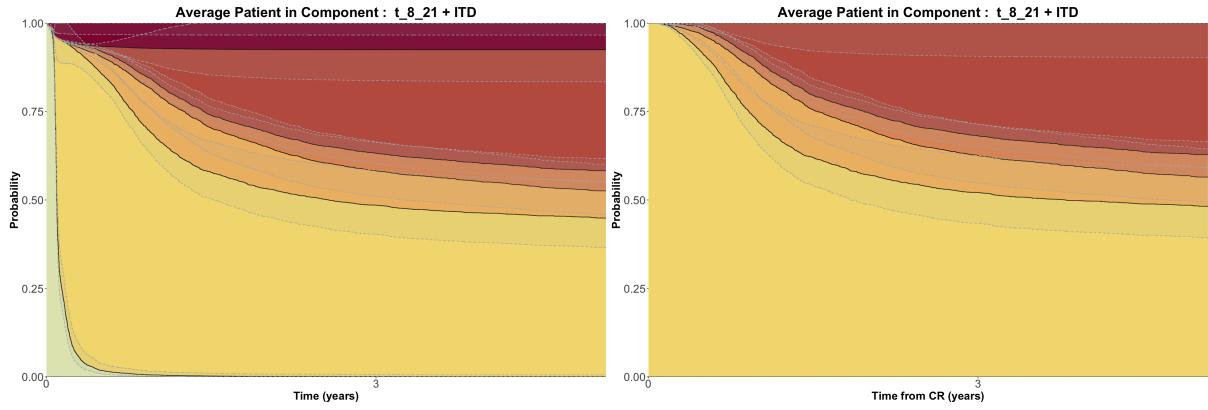
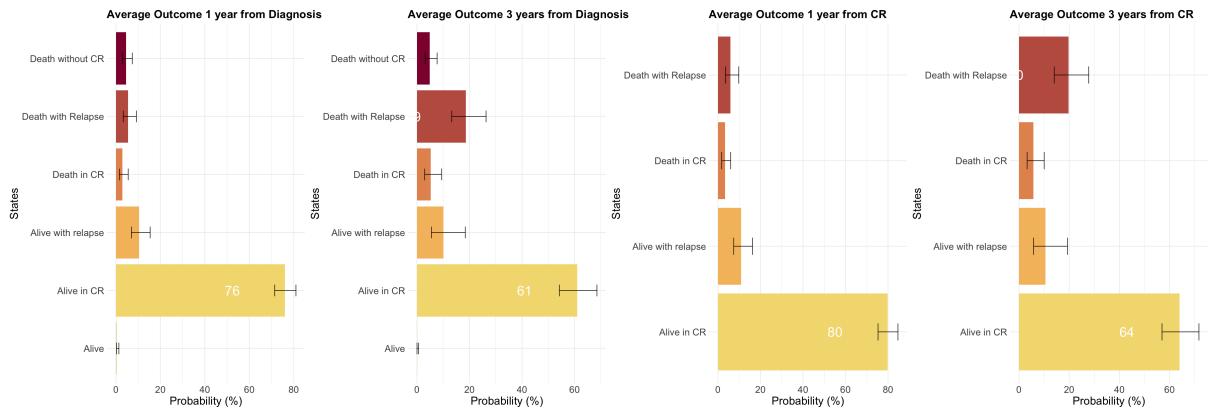
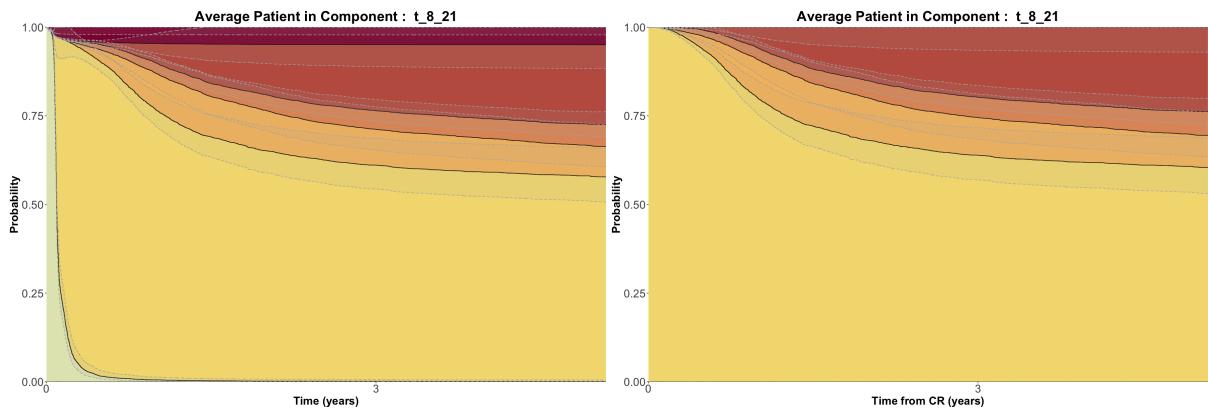
grid.arrange(diagnosis_1_comp_ITD,diagnosis_3_comp_ITD,CR_1_comp_ITD,CR_3_comp_ITD,nrow=1)
}
# dev.off()
}

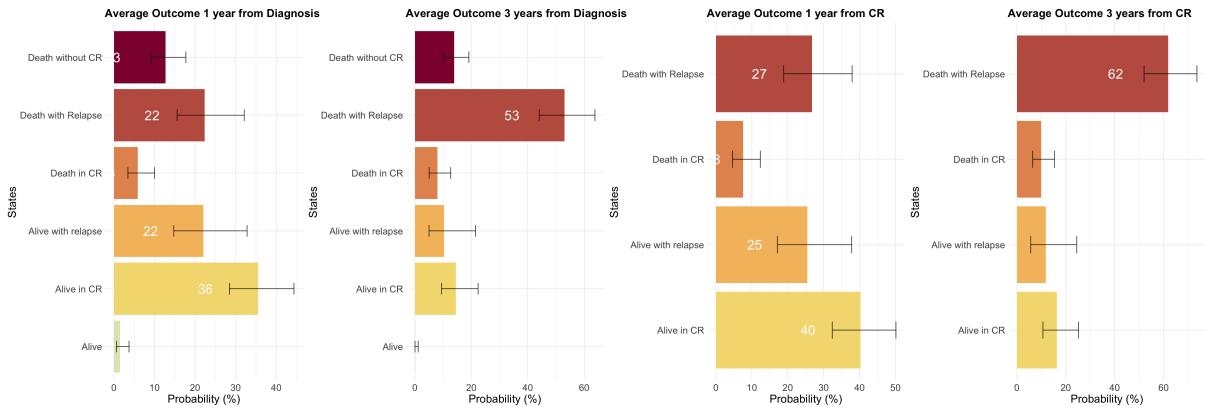
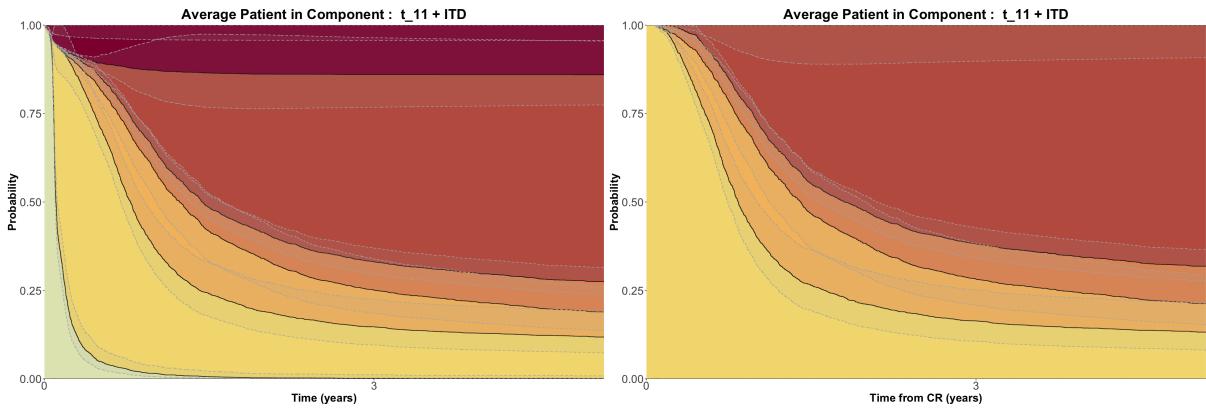
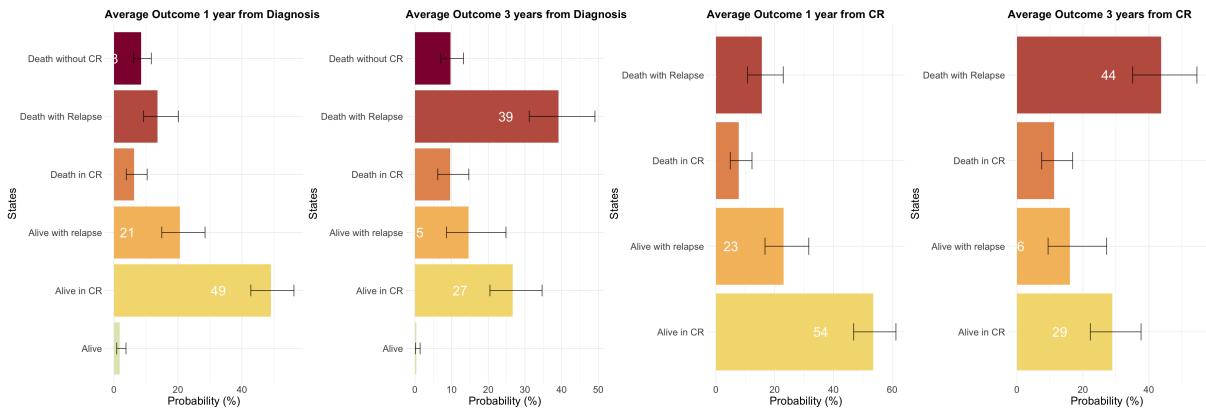
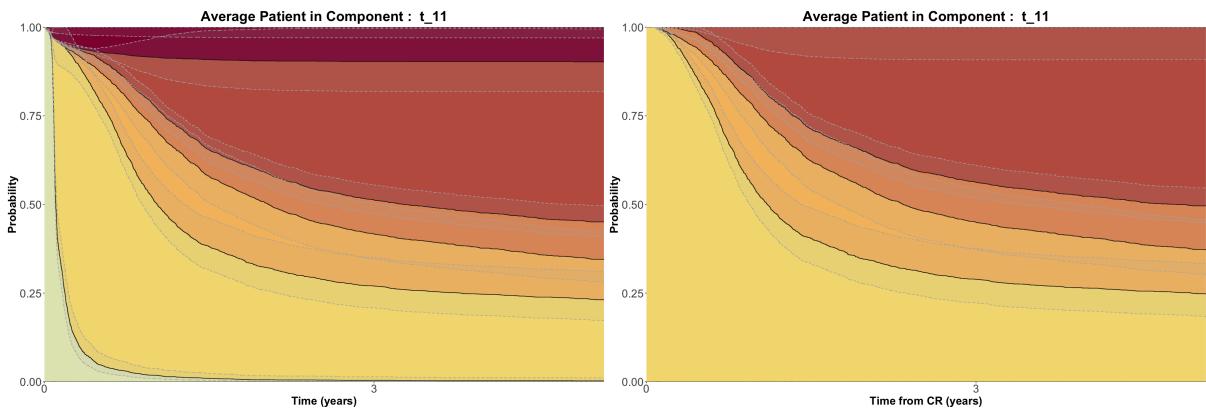
}

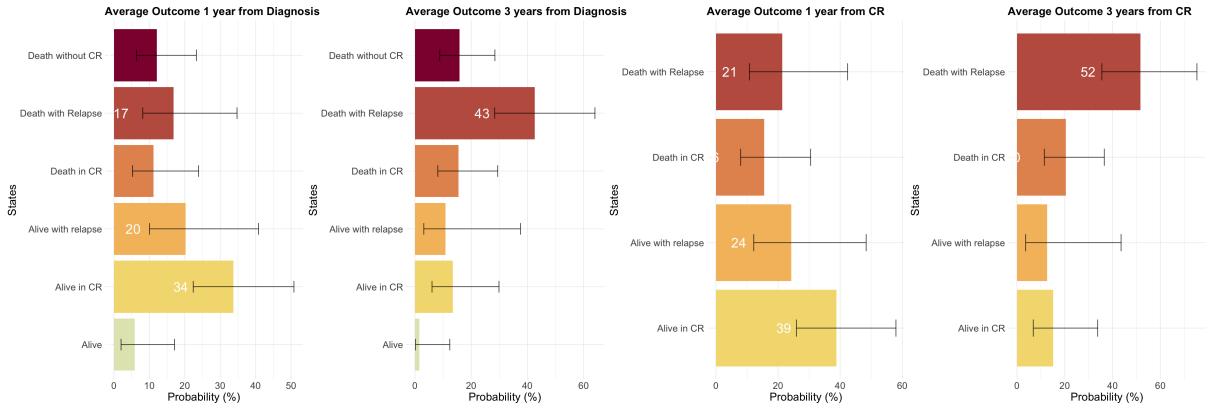
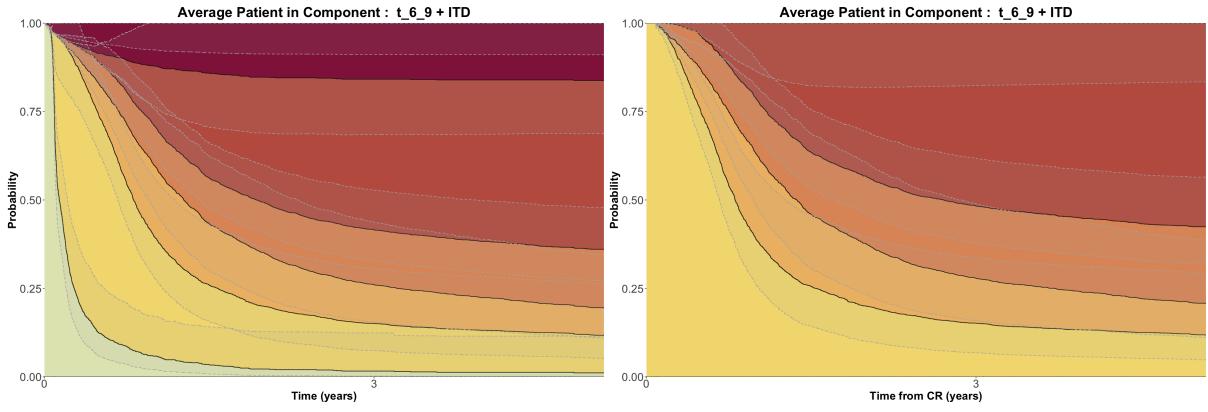
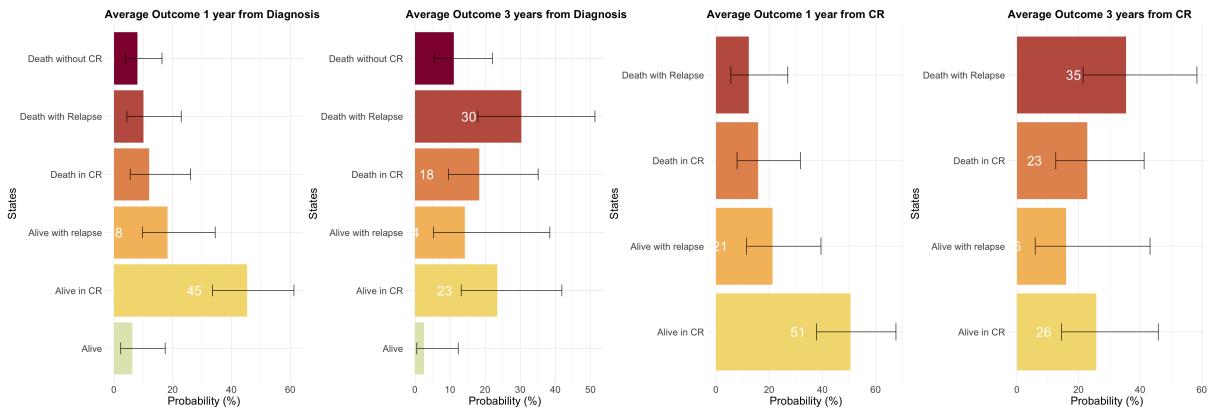
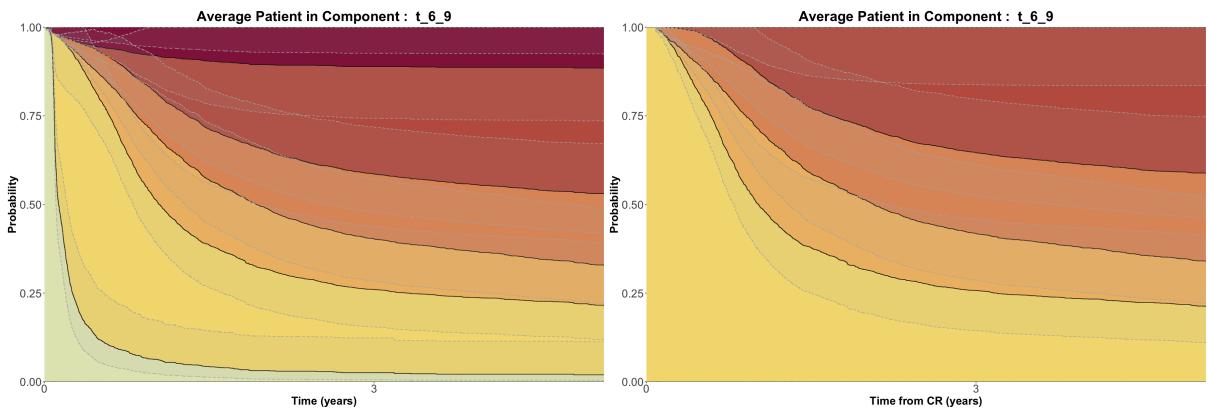
```

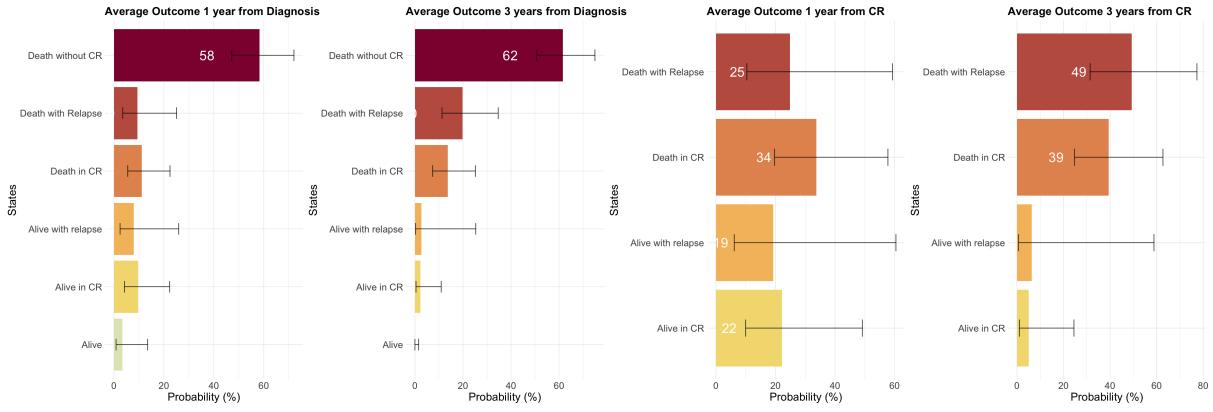
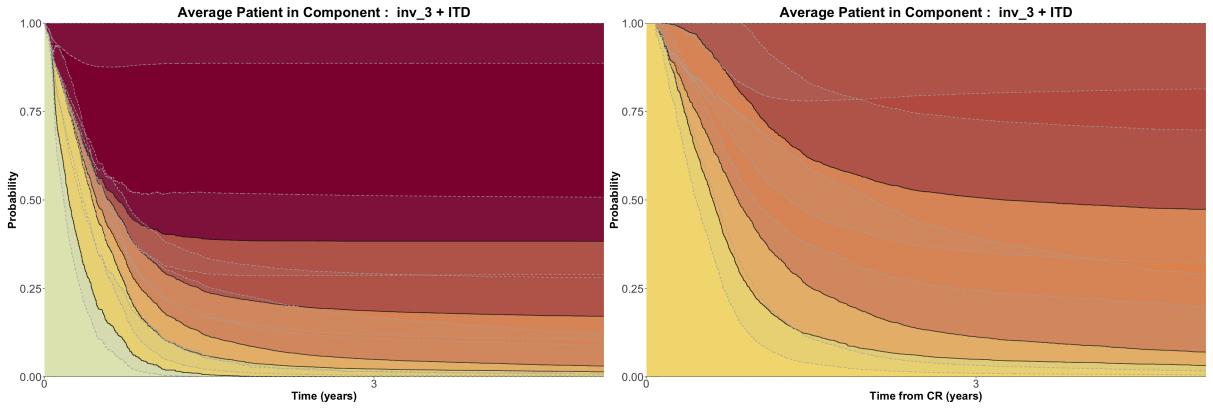
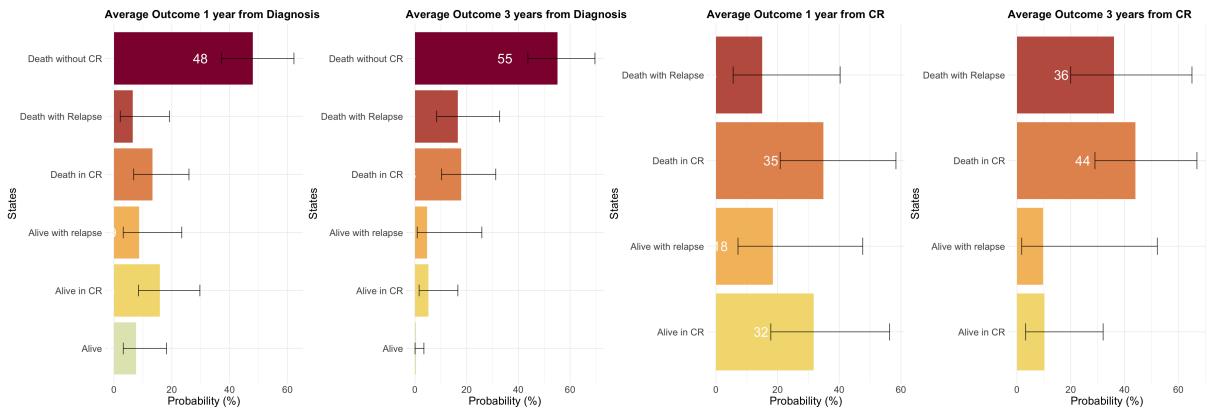
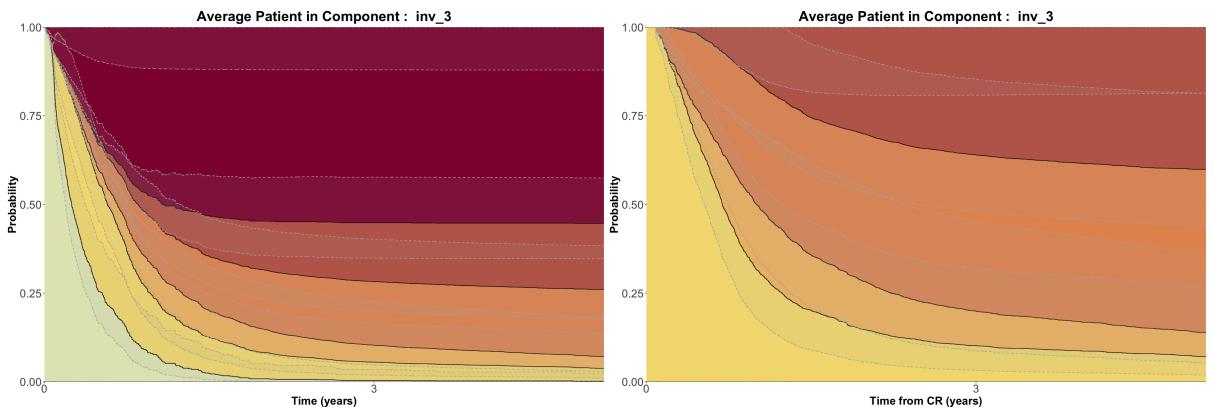


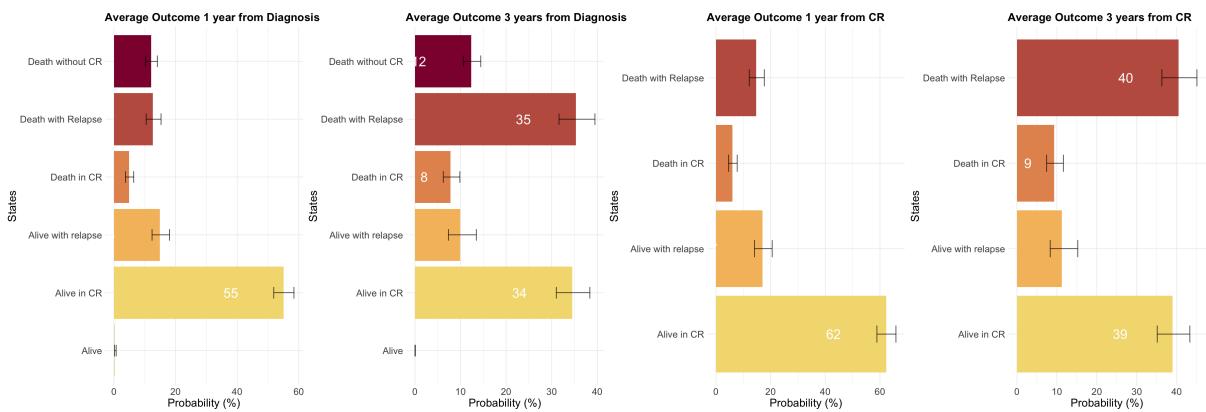
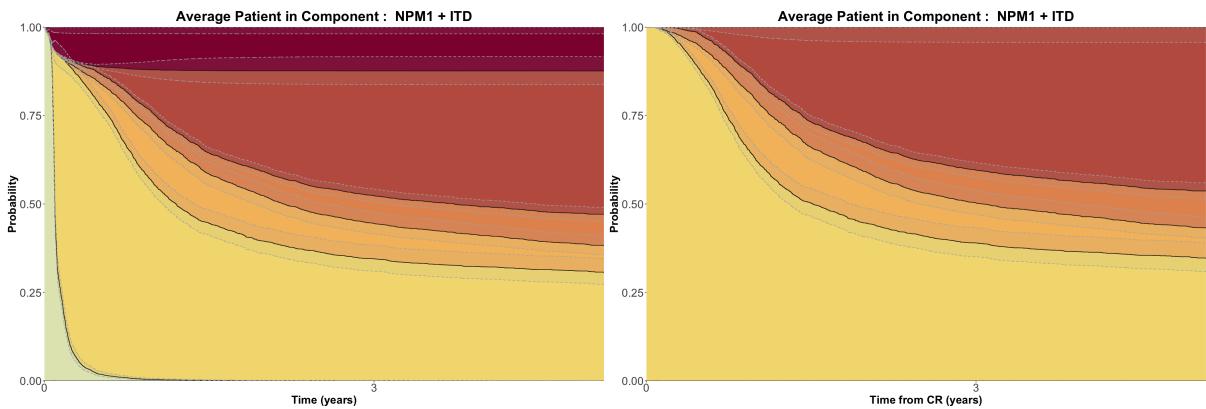
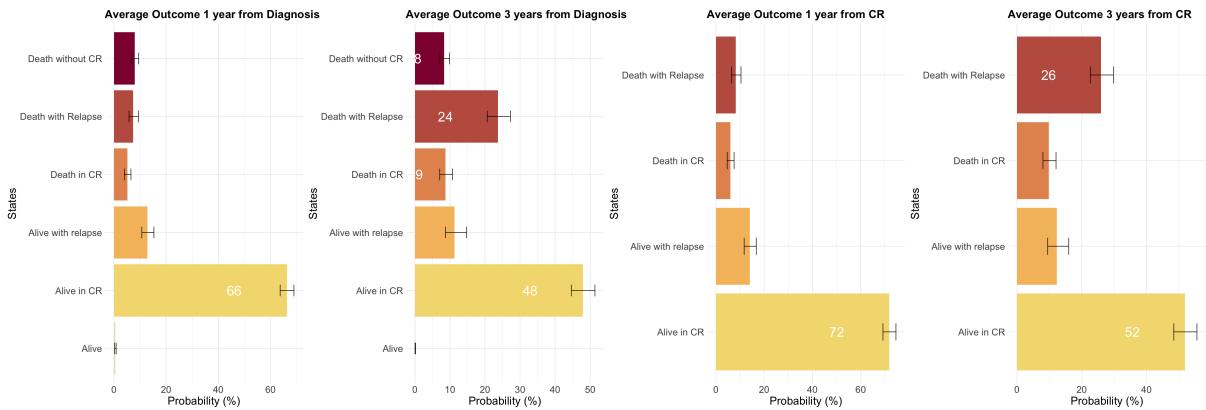
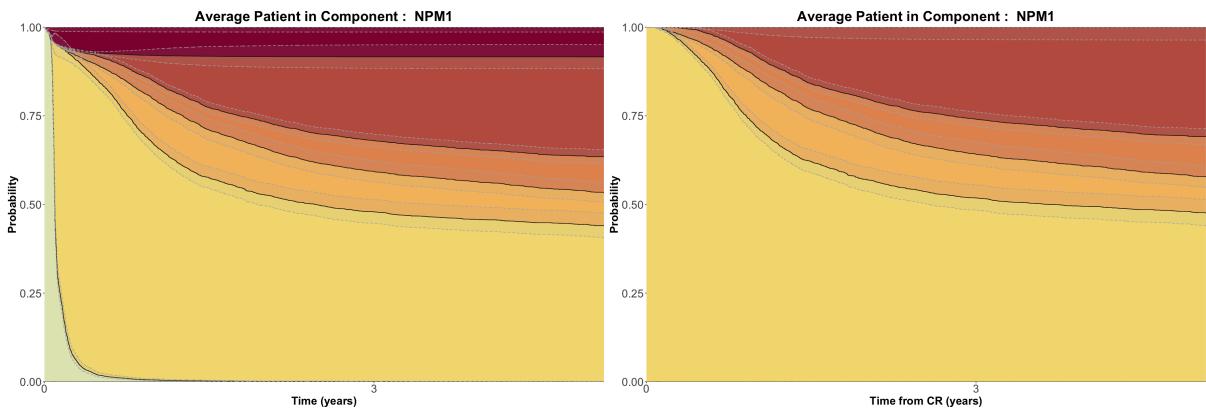


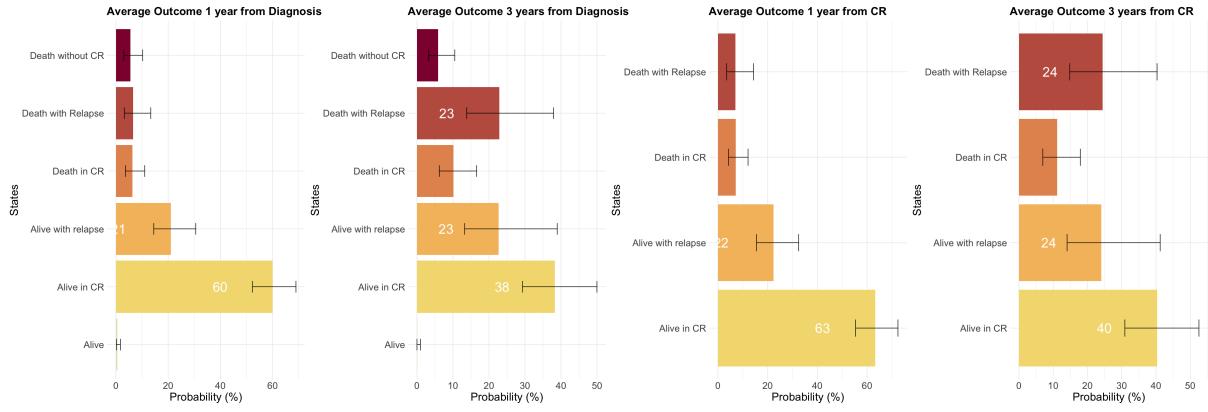
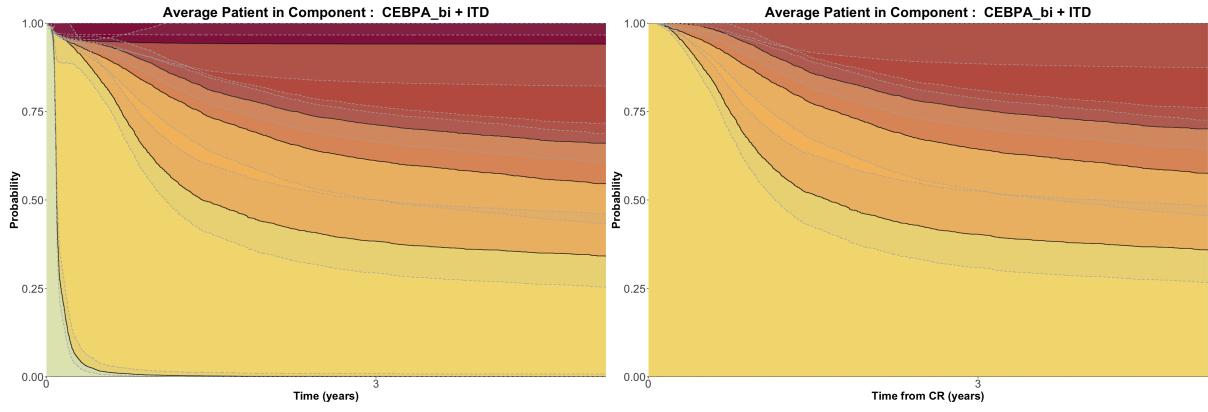
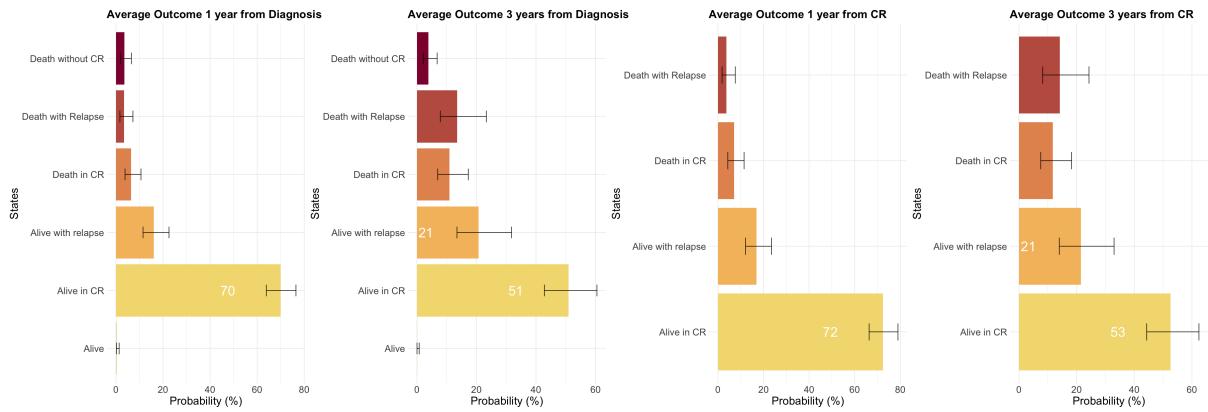
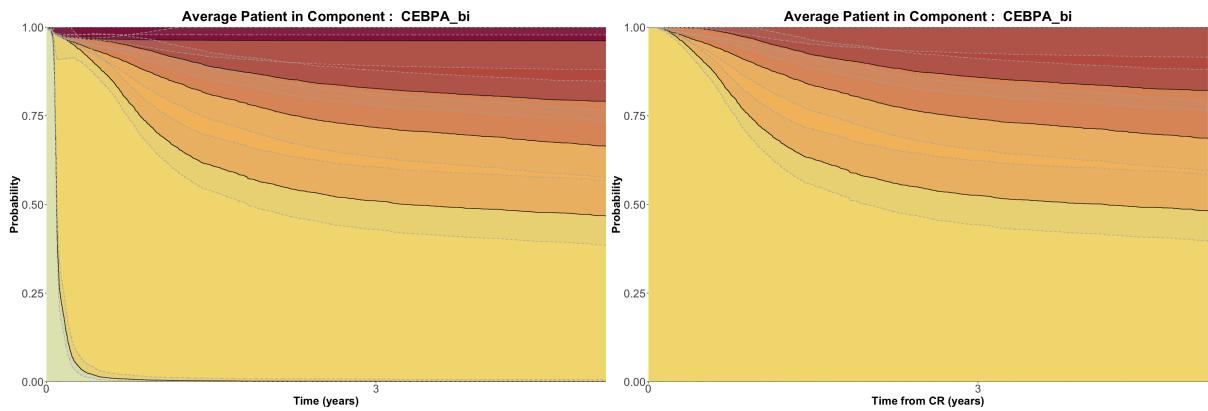


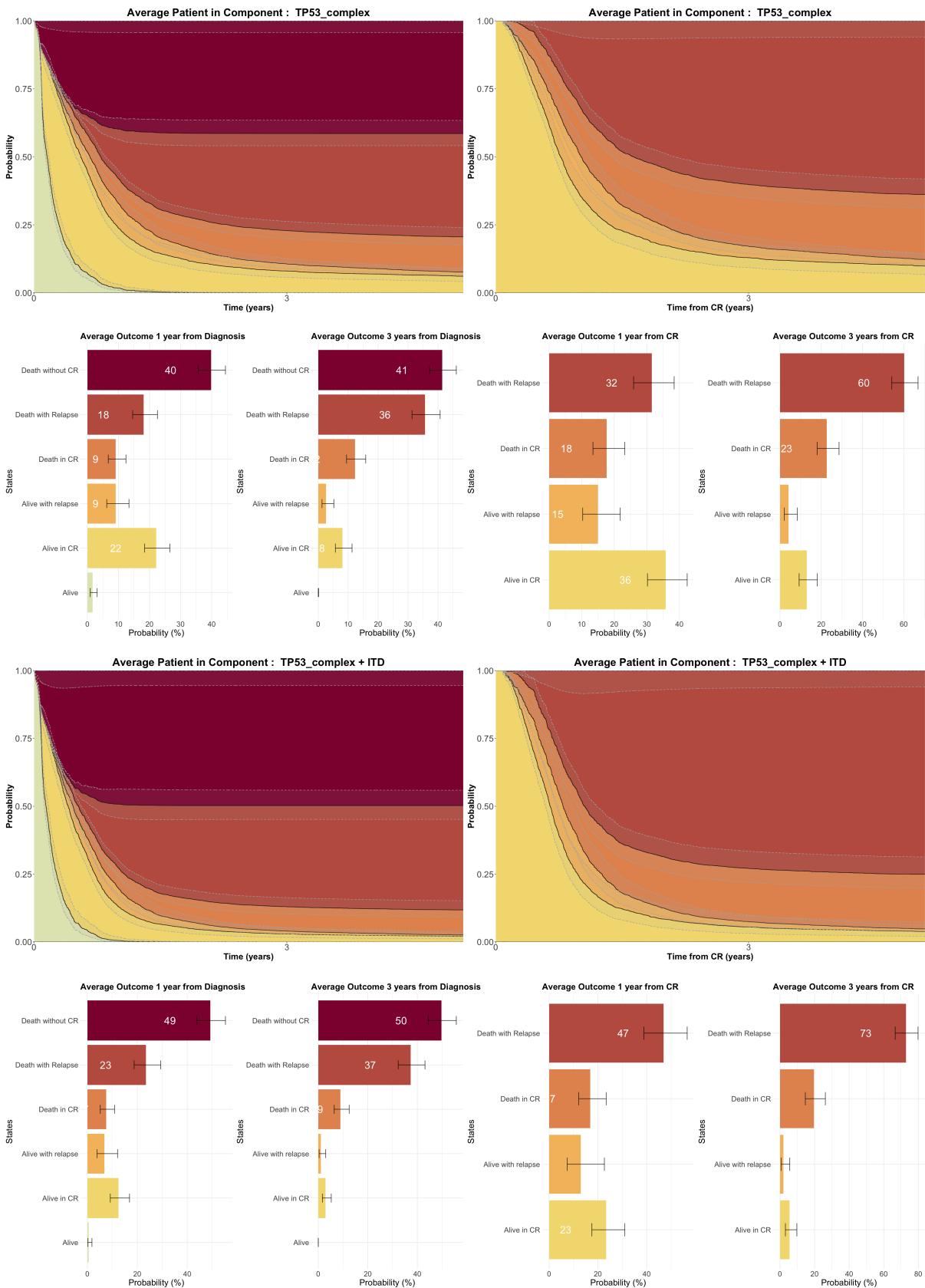


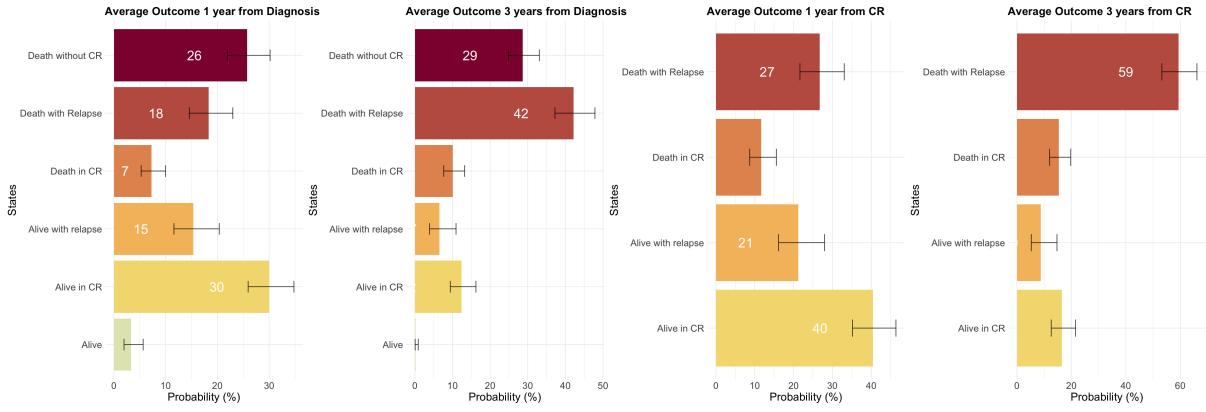
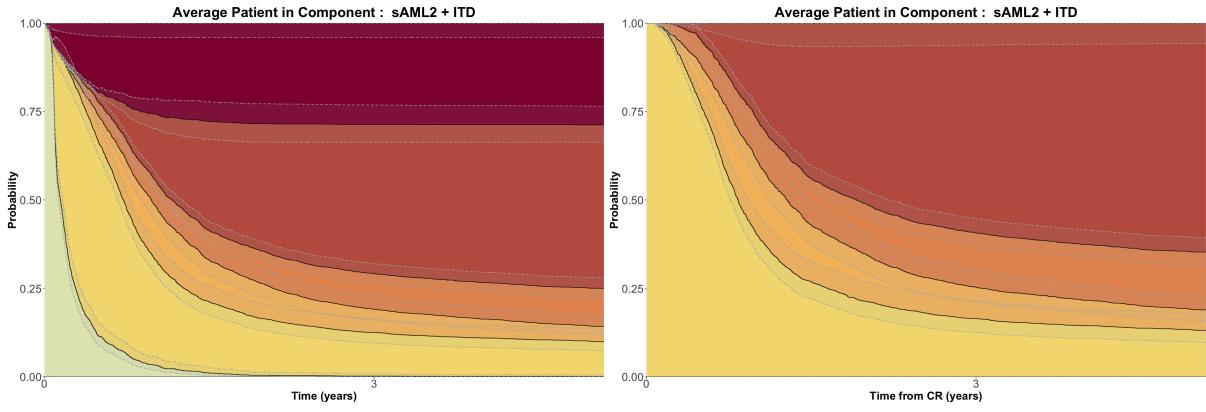
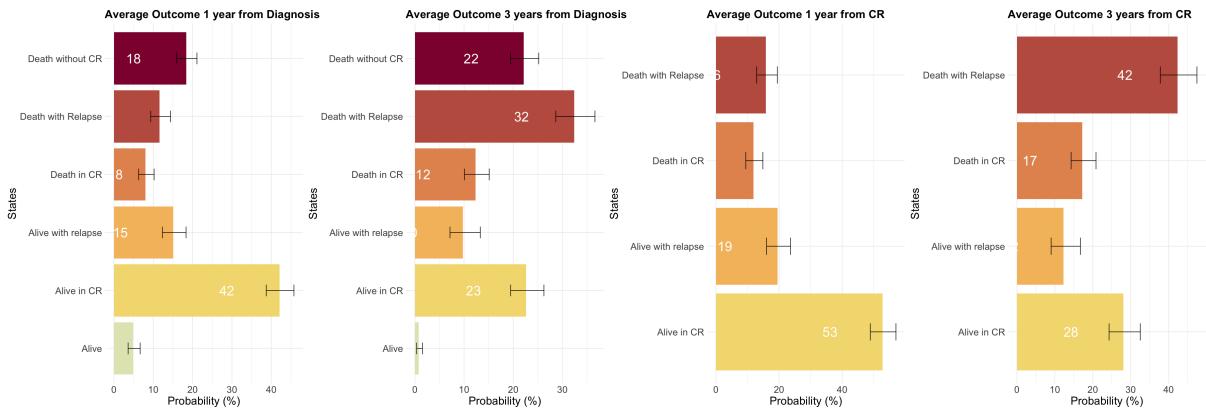
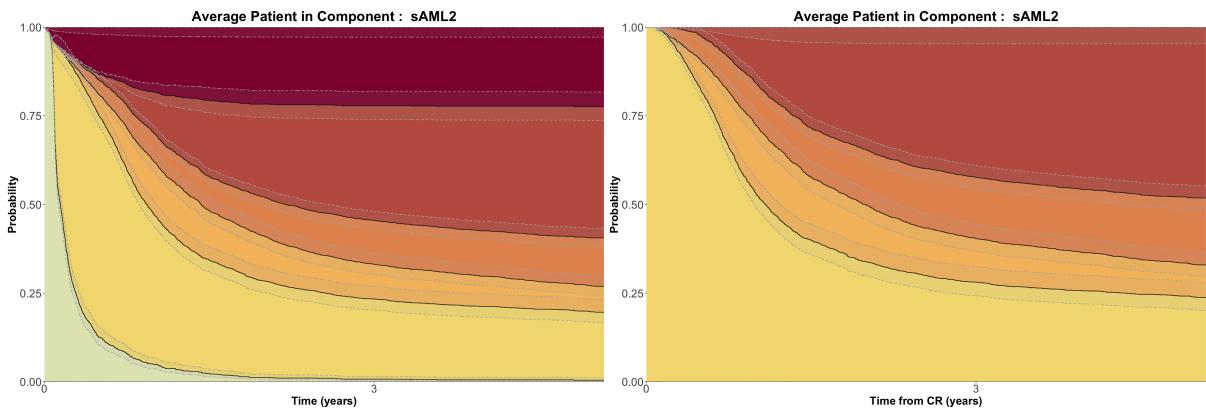


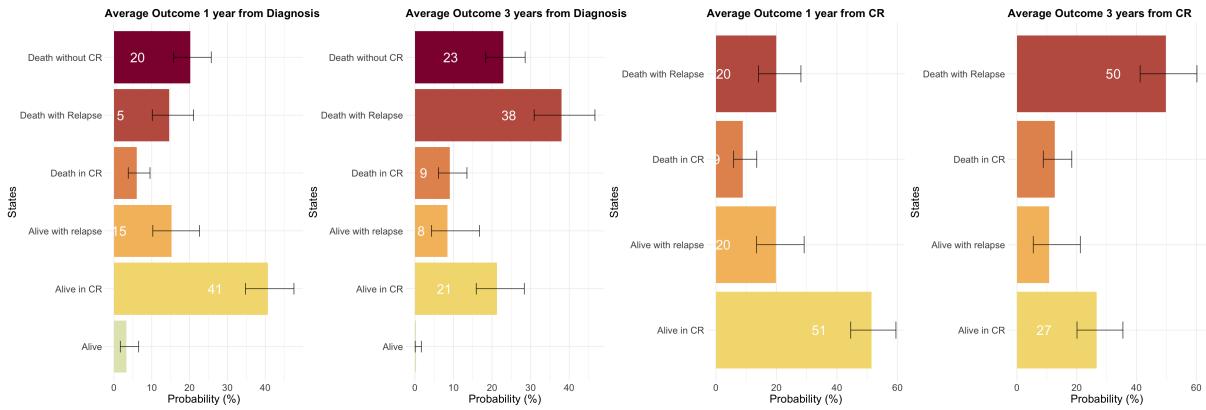
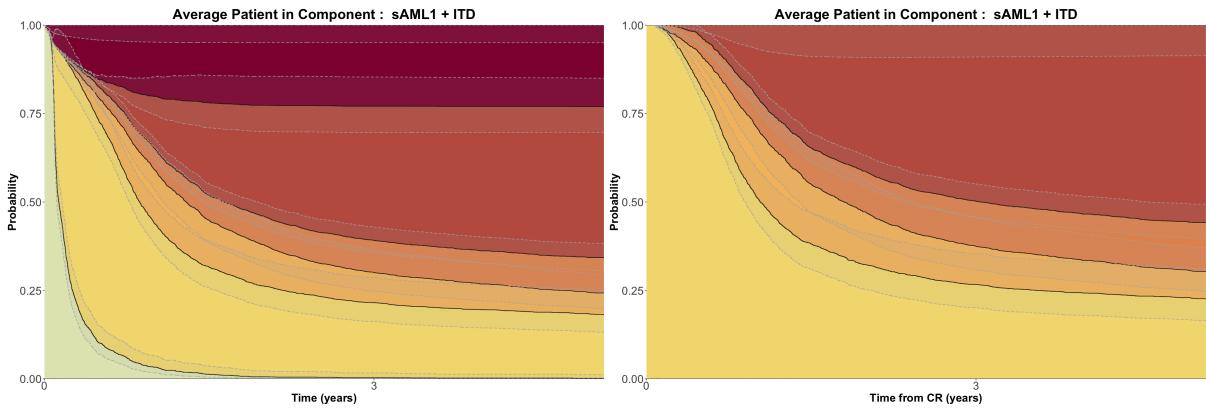
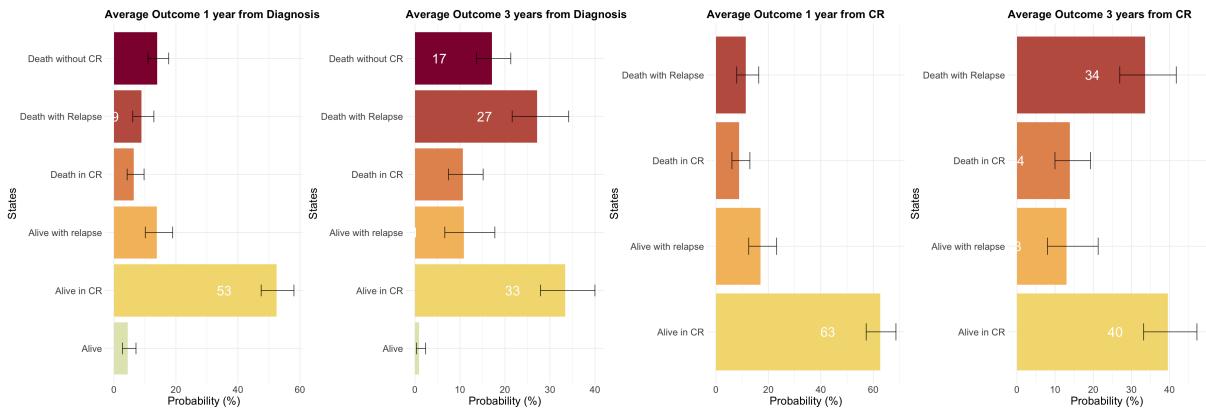
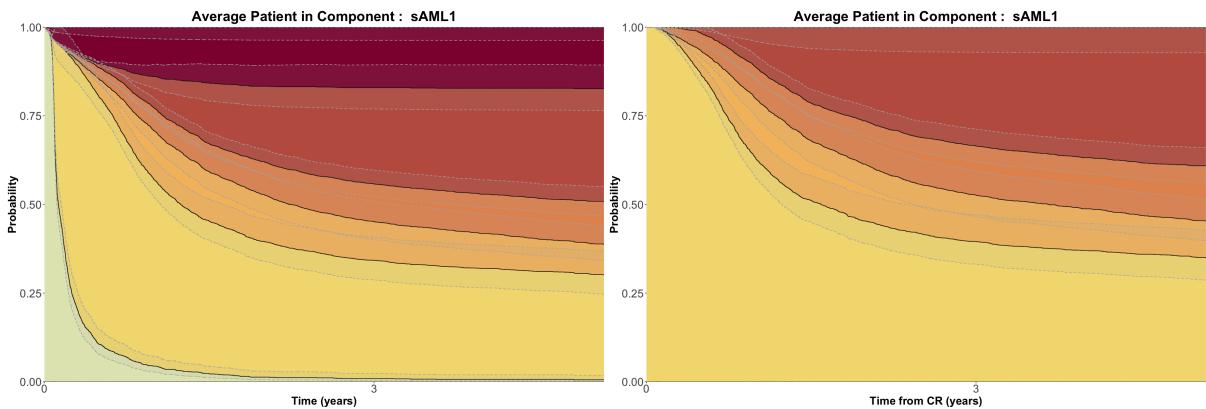


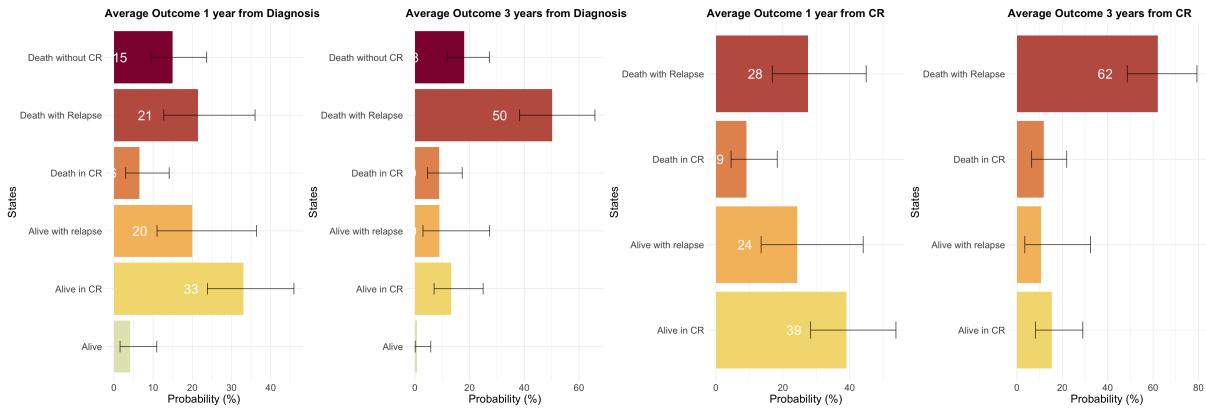
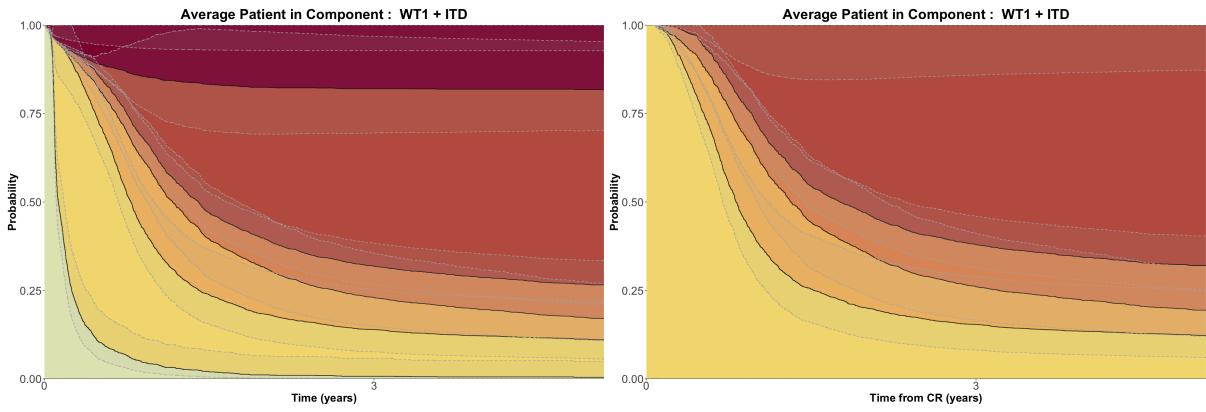
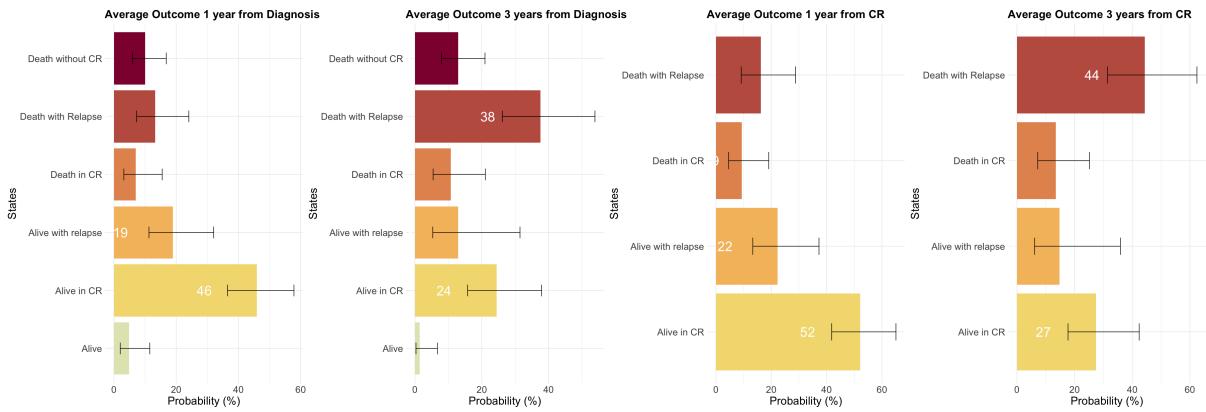
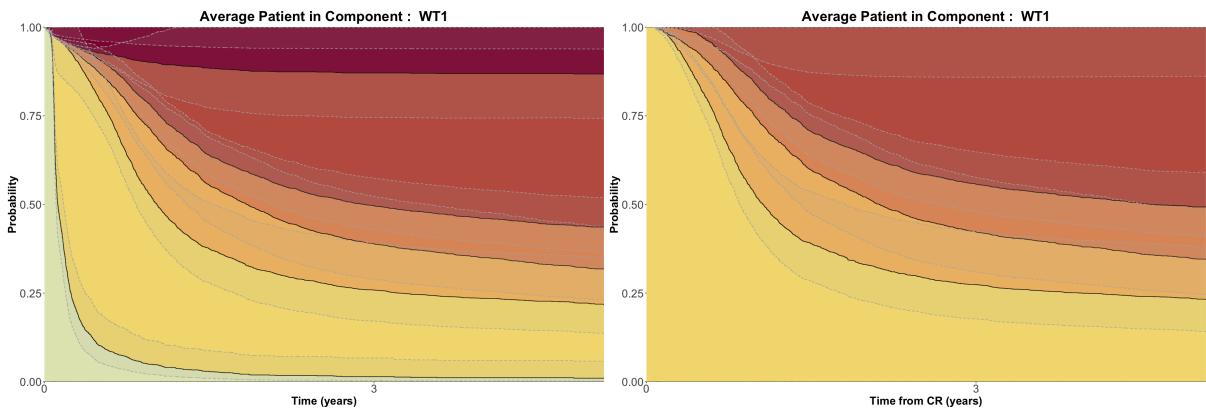


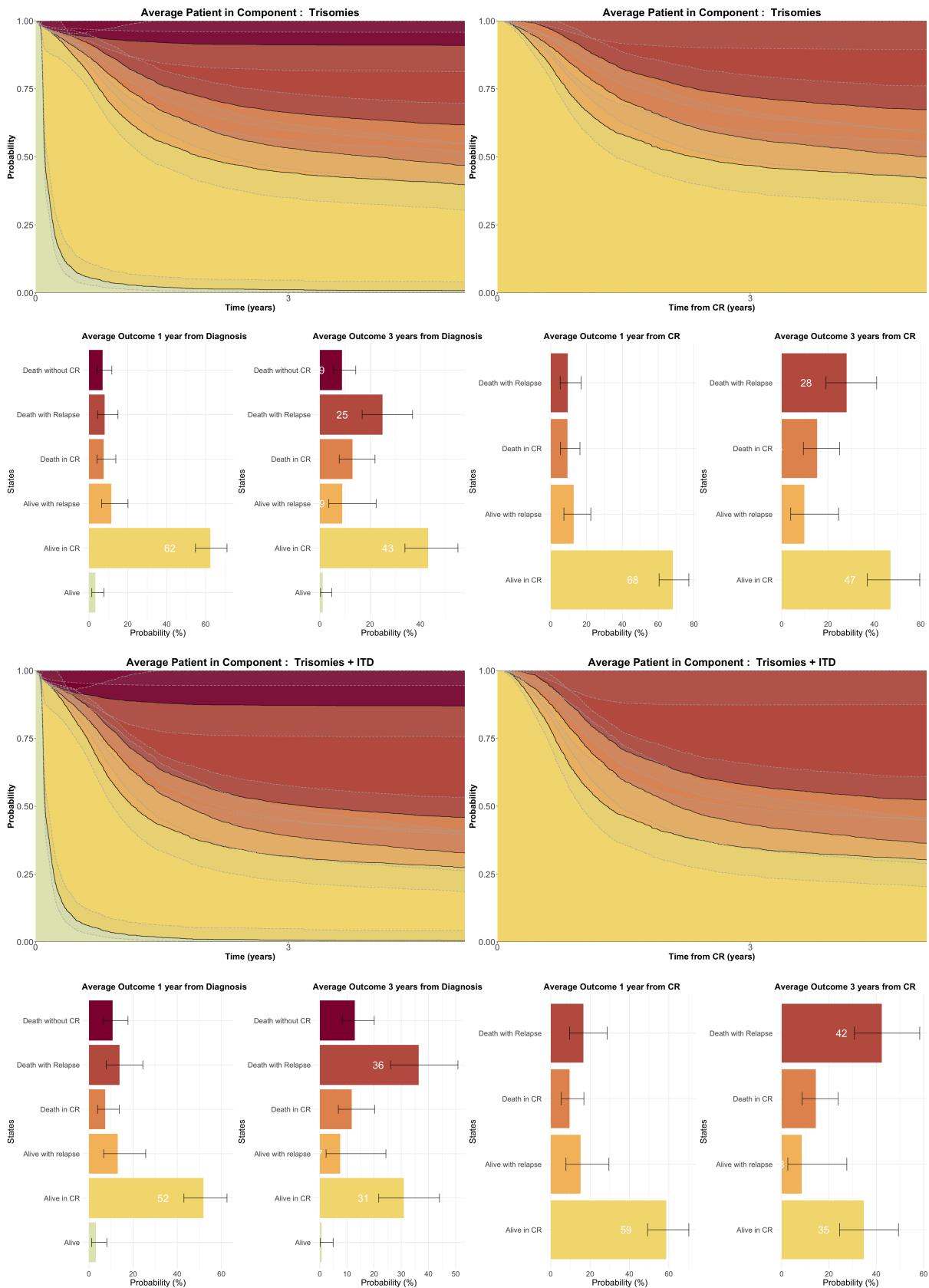


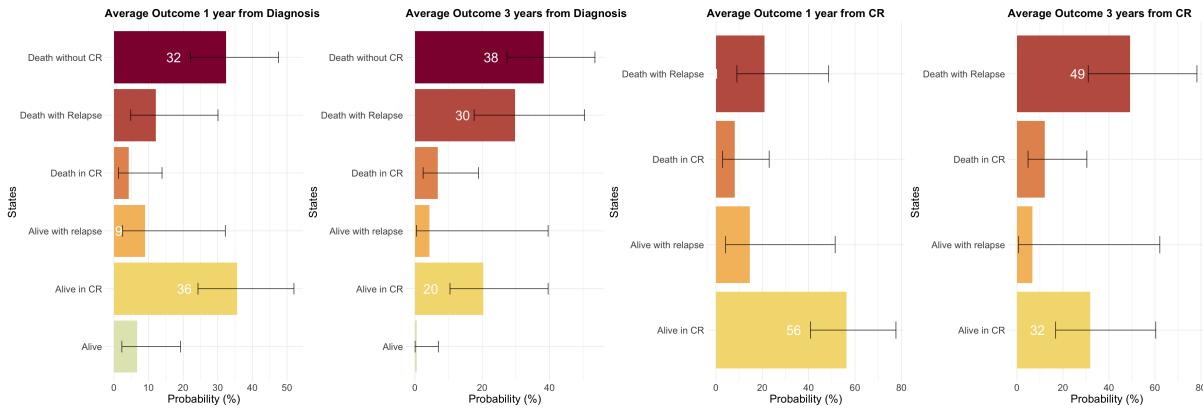
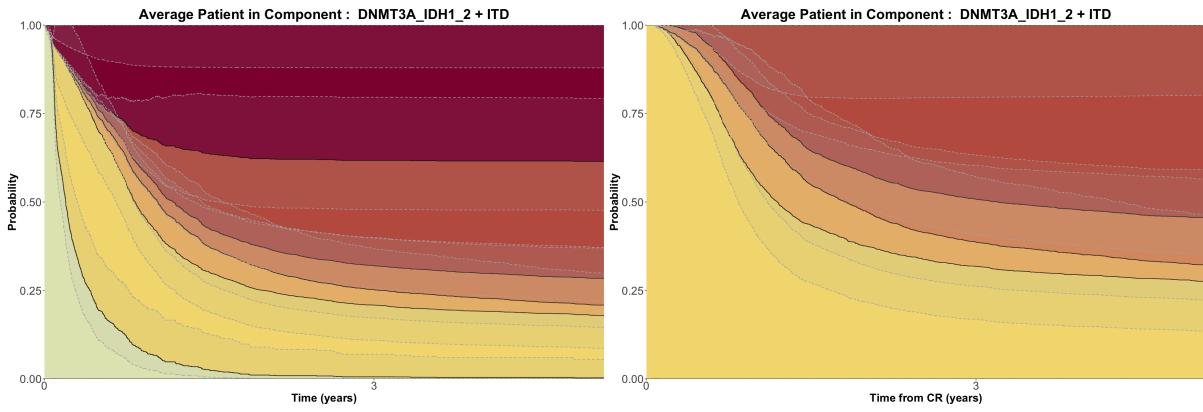
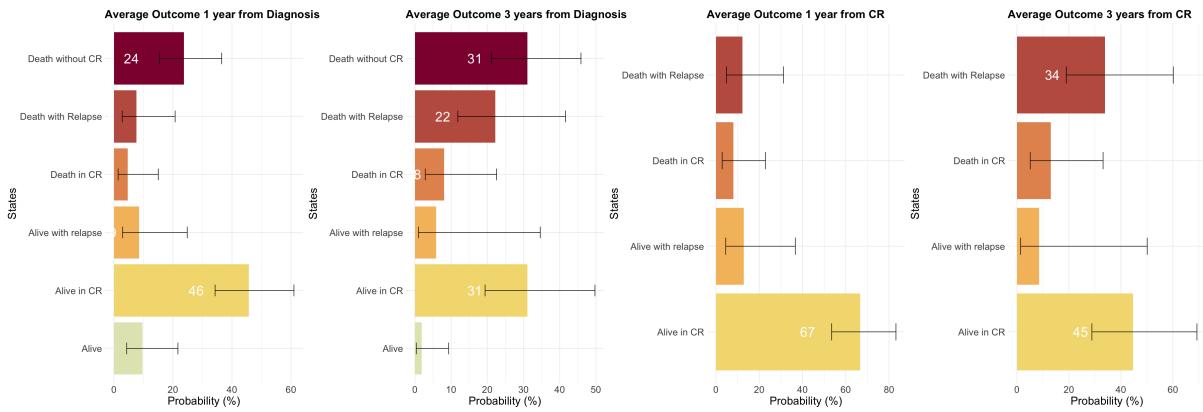
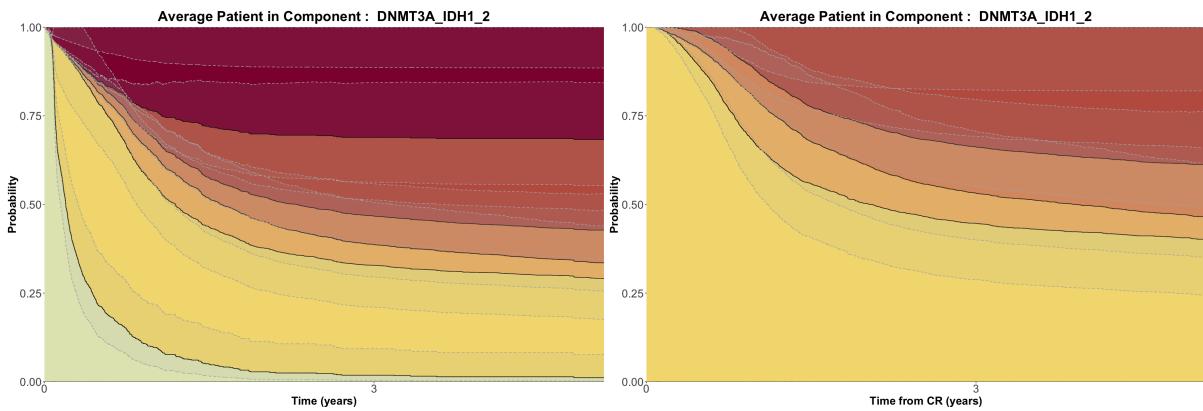


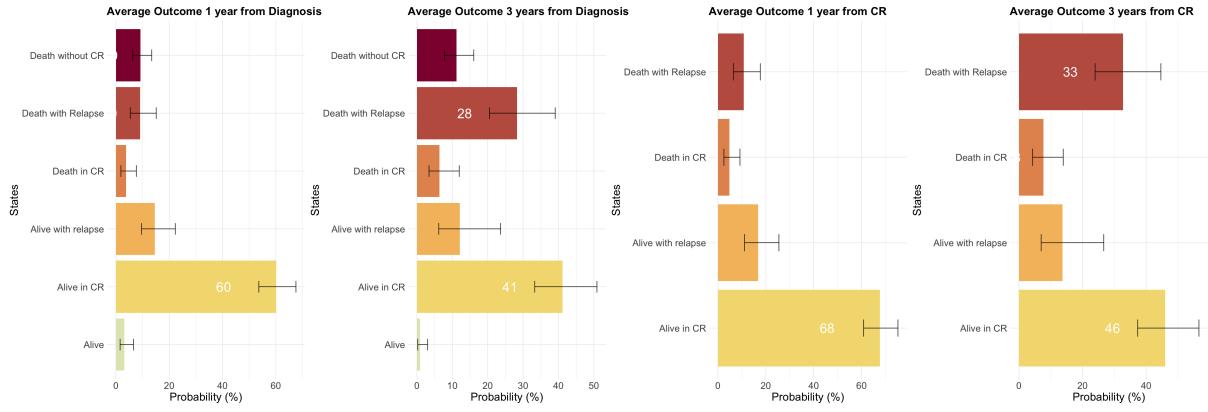
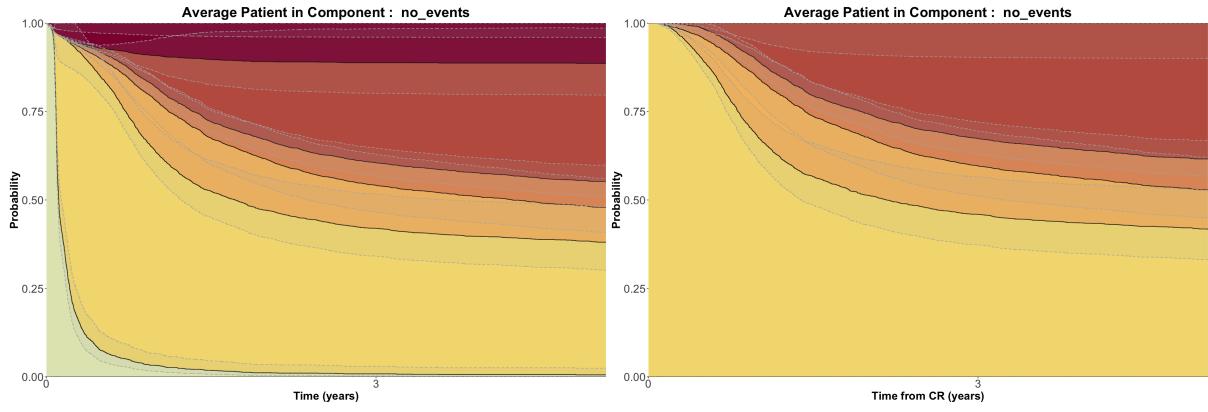
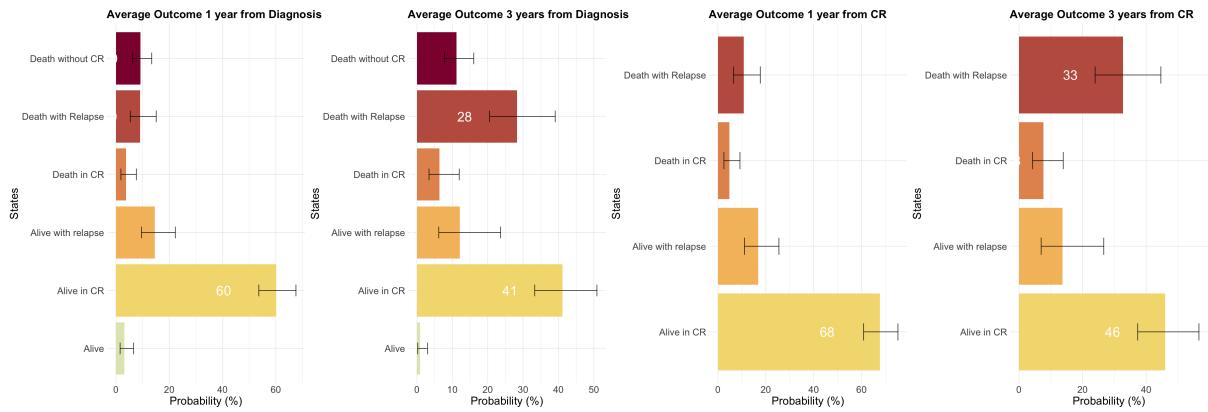
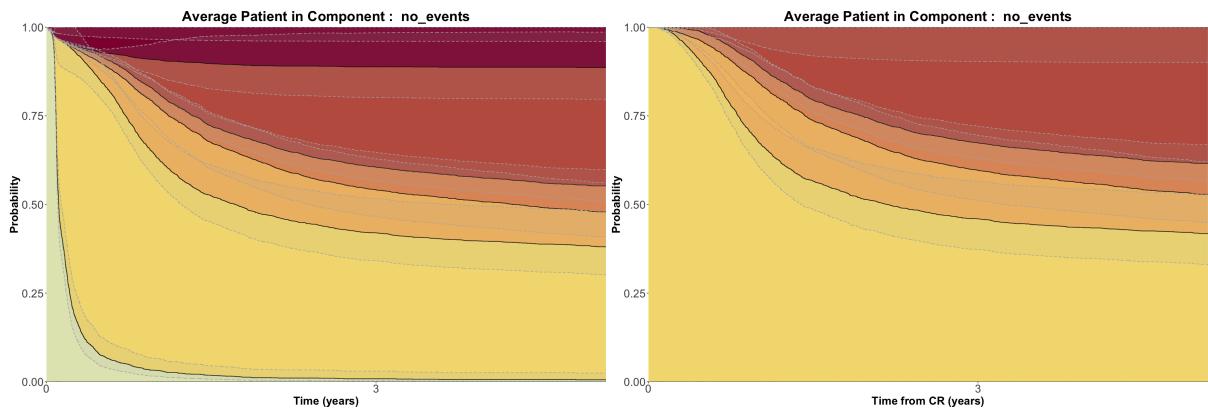


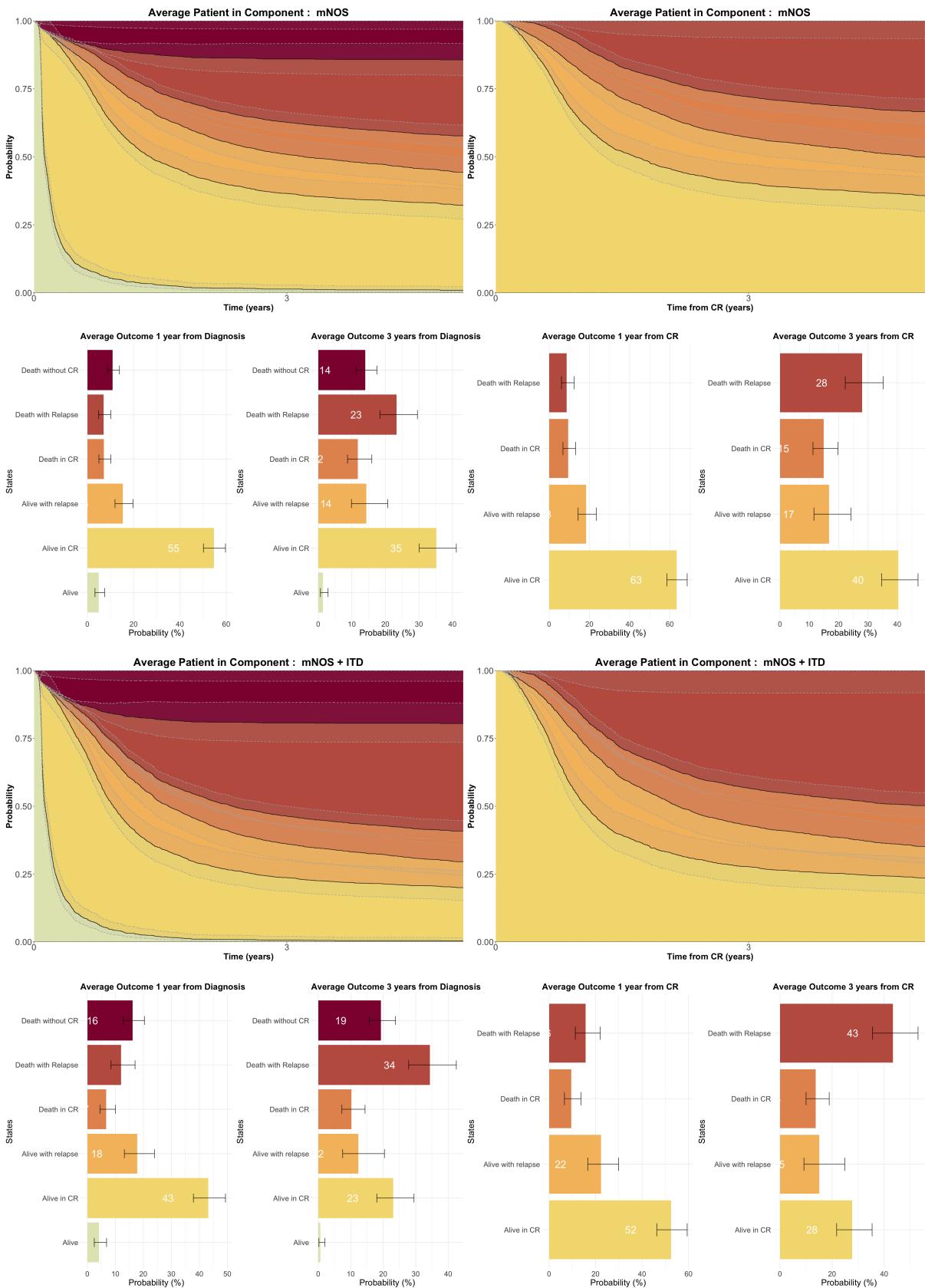












III) Final Tutorial for personalized outcome predictions

1. Empty dataframe with clinical,genes and cytos informations.

```
In [20]: user_data <- data.frame(covariates = c(demo,clin,all_gen,all_cyto))
user_data$value <- "Not specified"
rownames(user_data) <- user_data$covariates
user_data$covariates <- NULL
head(user_data,20)
```

A data.frame: 20 × 1

	value
	<chr>
gender	Not specified
age	Not specified
ahd	Not specified
perf_status	Not specified
bm_blasts	Not specified
secondary	Not specified
wbc	Not specified
hb	Not specified
plt	Not specified
ASXL1	Not specified
ASXL2	Not specified
ASXL3	Not specified
ATRX	Not specified
BAGE3	Not specified
BCOR	Not specified
BRAF	Not specified
CBFB	Not specified
CBL	Not specified
CDKN2A	Not specified
CEBPA_bi	Not specified

2. User specifies some covariates (imputation for the rest).

```
In [21]: change_covariates <- list(gender=1,age=63,bm_blasts=98,hb=9.4,plt=170,wbc=17.3,BCOR=1,SF3B1=1)
user_data[names(change_covariates),"value"] <- as.vector(unlist(change_covariates))
head(user_data,20)
```

A data.frame: 20 × 1

	value
	<chr>
gender	1
age	63
ahd	Not specified
perf_status	Not specified
bm_blasts	98
secondary	Not specified
wbc	17.3
hb	9.4
plt	170
ASXL1	Not specified
ASXL2	Not specified
ASXL3	Not specified
ATRX	Not specified
BAGE3	Not specified
BCOR	1
BRAF	Not specified
CBFB	Not specified
CBL	Not specified
CDKN2A	Not specified
CEBPA.bi	Not specified

3. Handle not specified values using stored clinical variables from above.

```
In [22]: user_data[user_data$value=="Not specified" & rownames(user_data) %in% c(
  all_gen,all_cyto) , "value"] <- 0
user_data[user_data$value=="Not specified" & rownames(user_data) %in% c(
  clin,demo) , "value"] <- covariates_median[rownames(user_data)[user_data$ 
value=="Not specified"]]
user_data$value <- as.numeric(user_data$value)  ### make sure values are 
numeric!
head(user_data,20)
```

A data.frame: 20 × 1

	value
	<code><dbl></code>
gender	1.0
age	63.0
ahd	0.0
perf_status	1.0
bm_blasts	98.0
secondary	1.0
wbc	17.3
hb	9.4
plt	170.0
ASXL1	0.0
ASXL2	0.0
ASXL3	0.0
ATRX	0.0
BAGE3	0.0
BCOR	1.0
BRAF	0.0
CBFB	0.0
CBL	0.0
CDKN2A	0.0
CEBPA_bi	0.0

4. Convert data into a class+clinical dataset.

```
In [23]: classify_components(user_data,proposal=F)
```

A data.frame: 27 × 1

	value
	<dbl>
gender	1.0
age	63.0
ahd	0.0
perf_status	1.0
bm_blasts	98.0
secondary	1.0
wbc	17.3
hb	9.4
plt	170.0
full_component_t_15_17	0.0
full_component_inv_16	0.0
full_component_t_8_21	0.0
full_component_t_11	0.0
full_component_t_6_9	0.0
full_component_inv_3	0.0
full_component_NPM1	0.0
full_component_CEBPA_bi	0.0
full_component_TP53_complex	0.0
full_component_sAML2	1.0
full_component_sAML1	0.0
full_component_WT1	0.0
full_component_Trisomies	0.0
full_component_DNMT3A_IDH1_2	0.0
full_component_no_events	0.0
full_component_mNOS	0.0
overlap	0.0
ITD	0.0

5. Prepare multistate plots and transition probabilities for patient centric and use corresponding saved risk ELN , proposal and component plots to display as well.

```
In [24]: # Find principal component to display and Proposal Risk
user_data_simplified <- classify_components(user_data,proposal=T)
princ_comp_patient <- find_principal_component(user_data_simplified)
# Find ELN risk
eln_patient <- eln_classification_webapp_simplified(data.frame(t(user_da
ta)))$eln_2017
# pdf( "Output_patient/Main/Main_examples.pdf",width=30,height=7)

#####
##### PART 1 : needs
##### computation (patient centric) #####
#####

user_data_simplified <- classify_components(user_data,proposal=T)
covariates <- c(demo,clin,comp,"ITD","NEW_favorable","NEW_intermediate",
"NEW_adverse")
characteristics <- rownames(user_data_simplified[c(comp,"ITD","age"),,F]
[which(user_data_simplified[c(comp,"ITD","age"),"value"]>0),,F])
set_notebook_plot_size(30,8)
par(mfrow=c(1,2))

tmp_data <- rbind(msdata,rep( NA, ncol( msdata ) ))
tmp_data[nrow(tmp_data),covariates] <- as.numeric(user_data_simplified[c
ovariates,])
patA <- tmp_data[rep(nrow(tmp_data),each=5),covariates]
patA$trans <- 1:5
attr(patA, "trans") <- tmat_data

patA <- expand.covs(patA, covariates, longnames = FALSE)
patA$strata <- patA$trans
msfa <- msfit(cfull, patA, trans = tmat_data)

pta <- probtrans(msfa, predt = 0,variance=T)
ord <- c(1,2,3,4,5,6)

title_str1<- ifelse("age" %in% characteristics,paste("Patient ","-year-o
ld and with",sep=as.character(round(user_data_simplified["age","value"
])), "Patient with")
title_str2 <- toString(str_remove(characteristics[characteristics %in% c
(comp,"ITD")],"full_component_"))
title_str <- paste(title_str1 , title_str2)

probtrans_plot_list <- list()
for (i in 1:2){
  if(i==1){
    colors <- statecols[1:6]
  } else{
    colors <- statecols[2:5]
  }
  probtrans_plot_list[[i]] <- plot_probtrans_Conf_Interval(pta,colors
= colors, from=i, ord = ord,xlim=c(0,5.1),
                           main = title_str,
                           xlab=ifelse(i==1,"Time (years)","Time fr
om CR (years)"))
}
```

```

}

grid.arrange(probtrans_plot_list[[1]],probtrans_plot_list[[2]],nrow=1)

ptrans1 <- data.frame(States=c("Alive","Alive in CR","Alive with relapse",
"Death in CR","Death with Relapse","Death without CR"),
proba=unlist(ptA[[1]][ptA[[1]]$time>=1 ,c("pstate1","pstate2","pstate3",
"pstate4","pstate5","pstate6")][1,]*100,use.names = F),
se=unlist(ptA[[1]][ptA[[1]]$time>=1 ,c("se1","se2","se3","se4","se5",
"se6")][1,]*100,use.names = F) )
diagnosis_1 <- ggplot(ptrans1,aes(x=States,y=proba)) + geom_bar(stat="identity",
fill=statecols)+geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(
proba) + crit * se / proba)),
width=.2, # Width of the error bars
position=position_dodge(.9))+geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",
size=8.5,face="bold")+ coord_flip() +ylab("Probability (%)")+
theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
ggttitle ("Patient Outcome 1 year from Diagnosis")

ptrans3 <- data.frame(States=c("Alive","Alive in CR","Alive with relapse",
"Death in CR","Death with Relapse","Death without CR"),
proba=unlist(ptA[[1]][ptA[[1]]$time>=3 ,c("pstate1","pstate2",
"pstate3","pstate4","pstate5","pstate6")][1,]*100,use.names = F),
se=unlist(ptA[[1]][ptA[[1]]$time>=3 ,c("se1","se2","se3",
"se4","se5","se6")][1,]*100,use.names = F) )
diagnosis_3 <- ggplot(ptrans3,aes(x=States,y=proba)) + geom_bar(stat="identity",
fill=statecols)+geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(
proba) + crit * se / proba)),
width=.2, # Width of the error bars
position=position_dodge(.9))+geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",
size=8.5,face="bold")+ coord_flip() +ylab("Probability (%)")+
theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
ggttitle ("Patient Outcome 3 years from Diagnosis")

ptrans1_CR <- data.frame(States=c("Alive in CR","Alive with relapse","Death in CR",
"Death with Relapse"),
proba=unlist(ptA[[2]][ptA[[2]]$time>=1 ,c("pstate2","pstate3",
"pstate4","pstate5")][1,]*100,use.names = F),
se=unlist(ptA[[2]][ptA[[2]]$time>=1 ,c("se2","se3","se4",
"se5")][1,]*100,use.names = F) )
CR_1 <- ggplot(ptrans1_CR,aes(x=States,y=proba)) + geom_bar(stat="identity",
fill=statecols[2:5])+geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(
proba) + crit * se / proba)),
width=.2, # Width of the error bars
position=position_dodge(.9))+geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",
size=8.5,face="bold")+ coord_flip() +ylab("Probability (%)")+
theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
ggttitle ("Patient Outcome 1 year from CR")

```

```

ptrans3_CR <- data.frame(States=c("Alive in CR","Alive with relapse","Death in CR","Death with Relapse"),
                         proba=unlist(ptA[[2]][ptA[[2]]$time>=3 ,c("pstate2","pstate3","pstate4","pstate5")][1,]*100,use.names = F),
                         se=unlist(ptA[[2]][ptA[[2]]$time>=3 ,c("se2","se3","se4",
                         "se5")][1,]*100,use.names = F) )
CR_3 <- ggplot(ptrans3_CR,aes(x=States,y=proba)) + geom_bar(stat="identity",
fill=statecols[2:5])+
geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(
proba) + crit * se / proba)),
width=.2, # Width of the error bars
position=position_dodge(.9))+geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",
size=8.5,face="bold")+ coord_flip() +ylab("Probability (%)")+
theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
ggttitle ("Patient Outcome 3 years from CR")

```

```
grid.arrange(diagnosis_1,diagnosis_3,CR_1,CR_3,nrow=1)
```

```
#####
##### PART 2 : Principal Component (with or without ITD (precomputed)) #####
#####

# Average Patient in Princ Comp / Princ Comp with ITD : we have already
presaved the computations in the cell before and we just need to plot them
```

```
### where the patient belongs ?
```

```

patient_princ_comp_prob_trans <- eval(parse(text = paste("ptA",princ_com
p_patient,sep=" ")))
patient_comp_class <- str_replace(str_remove(princ_comp_patient,"full_co
mponent_"),"_ITD"," with ITD")
```

```
### now we know which probtrans we need and we display the figures (it i
s instantaneous because no computation here)
```

```

probtrans_plot_list <- list()
for (i in 1:2){
  if(i==1){
    colors <- statecols[1:6]
  } else{
    colors <- statecols[2:5]
  }
  probtrans_plot_list[[i]] <- plot_probtrans_Conf_Interval(patient_pri
nc_comp_prob_trans,colors = colors, from=i, ord = ord,xlim=c(0,5.1),
main = paste("Average Patient Component
:",patient_comp_class),
xlab=ifelse(i==1,"Time (years)","Time fr
om CR (years)"))
}
```

```

}

grid.arrange(probtrans_plot_list[[1]],probtrans_plot_list[[2]],nrow=1)

ptrans1_princ_comp <- data.frame(States=c("Alive","Alive in CR","Alive with relapse","Death in CR","Death with Relapse","Death without CR"),
                                   proba=unlist(patient_princ_comp_prob_trans[[1]][patient_princ_comp_prob_trans[[1]]$time>=1 ,c("pstate1","pstate2","pstate3","pstate4","pstate5","pstate6")][1,]*100,use.names = F),
                                   se=unlist(patient_princ_comp_prob_trans[[1]][patient_princ_comp_prob_trans[[1]]$time>=1 ,c("se1","se2","se3","se4","se5","se6")][1,]*100,use.names = F))
diagnosis_1_princ_comp <- ggplot(ptrans1_princ_comp,aes(x=States,y=proba)) + geom_bar(stat="identity", fill=statecols) +
  geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(proba) + crit * se / proba)),
                width=.2, # Width of the error bars
                position=position_dodge(.9)) +
  geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white", size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +
  theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
  ggtitle ("Patient Outcome 1 year from Diagnosis")

ptrans3_princ_comp <- data.frame(States=c("Alive","Alive in CR","Alive with relapse","Death in CR","Death with Relapse","Death without CR"),
                                   proba=unlist(patient_princ_comp_prob_trans[[1]][patient_princ_comp_prob_trans[[1]]$time>=3 ,c("pstate1","pstate2","pstate3","pstate4","pstate5","pstate6")][1,]*100,use.names = F),
                                   se=unlist(patient_princ_comp_prob_trans[[1]][patient_princ_comp_prob_trans[[1]]$time>=3 ,c("se1","se2","se3","se4","se5","se6")][1,]*100,use.names = F))
diagnosis_3_princ_comp <- ggplot(ptrans3_princ_comp,aes(x=States,y=proba)) + geom_bar(stat="identity", fill=statecols) +
  geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(proba) + crit * se / proba)),
                width=.2, # Width of the error bars
                position=position_dodge(.9)) +
  geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white", size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +
  theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
  ggtitle ("Patient Outcome 3 years from Diagnosis")

ptrans1_CR_princ_comp <- data.frame(States=c("Alive in CR","Alive with relapse","Death in CR","Death with Relapse"),
                                       proba=unlist(patient_princ_comp_prob_trans[[2]][patient_princ_comp_prob_trans[[2]]$time>=1 ,c("pstate2","pstate3","pstate4","pstate5")][1,]*100,use.names = F),
                                       se=unlist(patient_princ_comp_prob_trans[[2]][patient_princ_comp_prob_trans[[2]]$time>=1 ,c("se2","se3","se4","se5")][1,]*100,use.names = F))
CR_1_princ_comp <- ggplot(ptrans1_CR_princ_comp,aes(x=States,y=proba)) + geom_bar(stat="identity", fill=statecols[2:5]) +
  geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(proba) + crit * se / proba)),
                width=.2, # Width of the error bars
                position=position_dodge(.9)) +
  geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white", size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +
  theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
  ggtitle ("Patient Outcome 1 year from CR")

```

```

    position=position_dodge(.9))+  

    geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",  

size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +  

theme_minimal() + theme(axis.text=element_text(size=17),axis.title=ele  

ment_text(size=20),plot.title=element_text(size=20,face="bold")) +  

ggttitle ("Patient Outcome 1 year from CR")

ptrans3_CR_princ_comp <- data.frame(States=c("Alive in CR","Alive with r  

elapse","Death in CR","Death with Relapse"),
proba=unlist(patient_princ_comp_prob_trans[[2]][patient_pr  

inc_comp_prob_trans[[2]]$time>=3 ,c("pstate2","pstate3","pstate4","pstat  

e5")][1,]*100,use.names = F),
se=unlist(patient_princ_comp_prob_trans[[2]][patient_princ  

_comp_prob_trans[[2]]$time>=3 ,c("se2","se3","se4","se5")][1,]*100,use.n  

ames = F)
CR_3_princ_comp <- ggplot(ptrans3_CR_princ_comp,aes(x=States,y=proba)) +  

geom_bar(stat="identity", fill=statecols[2:5]) +  

geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log  

(proba) + crit * se / proba)),
width=.2, # Width of the error bars
position=position_dodge(.9))+  

geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",  

size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +  

theme_minimal() + theme(axis.text=element_text(size=17),axis.title=ele  

ment_text(size=20),plot.title=element_text(size=20,face="bold")) +  

ggttitle ("Patient Outcome 3 years from CR")

```

```
grid.arrange(diagnosis_1_princ_comp,diagnosis_3_princ_comp,CR_1_princ_co  

mp,CR_3_princ_comp,nrow=1)
```

```
#####
##### PART 3 : ELN (p  

recomputed) #####
if(eln_patient=="favorable"){
  patient_eln_prob_trans <- ptA_eln_favorable
  patient_eln_class <- "Favorable"
} else if (eln_patient=="intermediate"){
  patient_eln_prob_trans <- ptA_eln_intermediate
  patient_eln_class <- "Intermediate"
} else{
  patient_eln_prob_trans <- ptA_eln_adverse
  patient_eln_class <- "Adverse"
}
print(patient_eln_class)
```

now we know which probtrans we need and we display the figures (it is instantaneous because no computation here)

```
probtrans_plot_list <- list()
for (i in 1:2){
  if(i==1){
    colors <- statecols[1:6]
  } else{
    colors <- statecols[2:5]
```

```

    }
    probtrans_plot_list[[i]] <- plot_probtrans_Conf_Interval(patient_eln
    _prob_trans,colors = colors, from=i, ord = ord,xlim=c(0,5.1),
                                              main = paste("Average Patient in ELN :
    ",patient_eln_class),
                                              xlab=ifelse(i==1,"Time (years)","Time fr
    om CR (years)"))
}
grid.arrange(probtrans_plot_list[[1]],probtrans_plot_list[[2]],nrow=1)

ptrans1_eln <- data.frame(States=c("Alive","Alive in CR","Alive with rel
apse","Death in CR","Death with Relapse","Death without CR"),
                           proba=unlist(patient_eln_prob_trans[[1]][patient_eln_prob_trans[[1
   ]]$time>=1 ,c("pstate1","pstate2","pstate3","pstate4","pstate5","pstate
   6")][1,]*100,use.names = F),
                           se=unlist(patient_eln_prob_trans[[1]][patient_eln_prob_trans[[1]]$t
   ime>=1 ,c("se1","se2","se3","se4","se5","se6")][1,]*100,use.names = F)
)
diagnosis_1_eln <- ggplot(ptrans1_eln,aes(x=States,y=proba)) + geom_bar(
stat="identity", fill=statecols)+geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log
(proba) + crit * se / proba)),
                                             width=.2,                                     # Width of the error bars
                                             position=position_dodge(.9))+geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",
size=8.5,face="bold")+ coord_flip() +ylab("Probability (%)")+
theme_minimal() + theme(axis.text=element_text(size=17),axis.title=ele
ment_text(size=20),plot.title=element_text(size=20,face="bold")) +
ggttitle ("Patient Outcome 1 year from Diagnosis")

ptrans3_eln <- data.frame(States=c("Alive","Alive in CR","Alive with rel
apse","Death in CR","Death with Relapse","Death without CR"),
                           proba=unlist(patient_eln_prob_trans[[1]][patient_eln_prob_
   trans[[1]]$time>=3 ,c("pstate1","pstate2","pstate3","pstate4","pstate5",
   "pstate6")][1,]*100,use.names = F),
                           se=unlist(patient_eln_prob_trans[[1]][patient_eln_prob_trans[[1]]$t
   ime>=3 ,c("se1","se2","se3","se4","se5","se6")][1,]*100,use.names = F))
diagnosis_3_eln <- ggplot(ptrans3_eln,aes(x=States,y=proba)) + geom_bar(
stat="identity", fill=statecols)+geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log
(proba) + crit * se / proba)),
                                             width=.2,                                     # Width of the error bars
                                             position=position_dodge(.9))+geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",
size=8.5,face="bold")+ coord_flip() +ylab("Probability (%)")+
theme_minimal() + theme(axis.text=element_text(size=17),axis.title=ele
ment_text(size=20),plot.title=element_text(size=20,face="bold")) +
ggttitle ("Patient Outcome 3 years from Diagnosis")

ptrans1_CR_eln <- data.frame(States=c("Alive in CR","Alive with relapse",
   "Death in CR","Death with Relapse"),
                           proba=unlist(patient_eln_prob_trans[[2]][patient_eln_prob_
   trans[[2]]$time>=1 ,c("pstate2","pstate3","pstate4","pstate5")][1,]*100,
   use.names = F),
                           se=unlist(patient_eln_prob_trans[[2]][patient_eln_prob_trans[[2]]$time>=1 ,c("pstate2","pstate3","pstate4","pstate5")][1,]*100,
   use.names = F))

```

```

ns[[2]]$time>=1 ,c("se2","se3","se4","se5"))[1,]*100,use.names = F)
)
CR_1_eln <- ggplot(ptrans1_CR_eln,aes(x=States,y=proba)) + geom_bar(stat
="identity", fill=statecols[2:5])+geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log
(proba) + crit * se / proba)),
width=.2, # Width of the error bars
position=position_dodge(.9))+geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",
size=8.5,face="bold")+ coord_flip() +ylab("Probability (%)")+
theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
ggttitle ("Patient Outcome 1 year from CR")

ptrans3_CR_eln <- data.frame(States=c("Alive in CR","Alive with relapse"
,"Death in CR","Death with Relapse"),
proba=unlist(patient_eln_prob_trans[[2]][patient_eln_prob_
trans[[2]]$time>=3 ,c("pstate2","pstate3","pstate4","pstate5"))[1,]*100,
use.names = F),
se=unlist(patient_eln_prob_trans[[2]][patient_eln_prob_tra
ns[[2]]$time>=3 ,c("se2","se3","se4","se5"))[1,]*100,use.names = F)
)
CR_3_eln <- ggplot(ptrans3_CR_eln,aes(x=States,y=proba)) + geom_bar(stat
="identity", fill=statecols[2:5])+geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log
(proba) + crit * se / proba)),
width=.2, # Width of the error bars
position=position_dodge(.9))+geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",
size=8.5,face="bold")+ coord_flip() +ylab("Probability (%)")+
theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
ggttitle ("Patient Outcome 3 years from CR")

grid.arrange(diagnosis_1_eln,diagnosis_3_eln,CR_1_eln,CR_3_eln,nrow=1)

```

```

#####
##### PART 4 #####
#####

# Average Patient in risk Proposal : we have already presaved the computations in the cell before and we just need to plot them
```

where the patient belongs ?

```

if(user_data_simplified[ "NEW_favorable",]==1){
  patient_proposal_prob_trans <- ptA_proposal_favorable
  patient_risk_class <- "Favorable"
} else if (user_data_simplified[ "NEW_intermediate",]==1){
  patient_proposal_prob_trans <- ptA_proposal_intermediate
  patient_risk_class <- "Intermediate"
} else{
  patient_proposal_prob_trans <- ptA_proposal_adverse
  patient_risk_class <- "Adverse"
}
```

```

}

print(patient_risk_class)

### now we know which probtrans we need and we display the figures (it is
# instantaneous because no computation here)

probtrans_plot_list <- list()
for (i in 1:2){
  if(i==1){
    colors <- statecols[1:6]
  } else{
    colors <- statecols[2:5]
  }
  probtrans_plot_list[[i]] <- plot_probtrans_Conf_Interval(patient_proposal_prob_trans,colors = colors, from=i, ord = ord,xlim=c(0,5.1),
                                                          main = paste("Average Patient in the Proposal Molecular Classification : ",patient_risk_class),
                                                          xlab=ifelse(i==1,"Time (years)","Time from CR (years)"))
}
grid.arrange(probtrans_plot_list[[1]],probtrans_plot_list[[2]],nrow=1)

ptrans1_proposal <- data.frame(States=c("Alive","Alive in CR","Alive with relapse","Death in CR","Death with Relapse","Death without CR"),
                                proba=unlist(patient_proposal_prob_trans[[1]][patient_proposal_prob_trans[[1]]$time>=1 ,c("pstate1","pstate2","pstate3","pstate4","pstate5","pstate6")][1,]*100,use.names = F),
                                se=unlist(patient_proposal_prob_trans[[1]][patient_proposal_prob_trans[[1]]$time>=1 ,c("se1","se2","se3","se4","se5","se6")][1,]*100,use.names = F))
diagnosis_1_proposal <- ggplot(ptrans1_proposal,aes(x=States,y=proba)) +
  geom_bar(stat="identity", fill=statecols) +
  geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(proba) + crit * se / proba)),
                width=.2, # Width of the error bars
                position=position_dodge(.9)) +
  geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",
            size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +
  theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
  ggtitle ("Patient Outcome 1 year from Diagnosis")

ptrans3_proposal <- data.frame(States=c("Alive","Alive in CR","Alive with relapse","Death in CR","Death with Relapse","Death without CR"),
                                proba=unlist(patient_proposal_prob_trans[[1]][patient_proposal_prob_trans[[1]]$time>=3 ,c("pstate1","pstate2","pstate3","pstate4","pstate5","pstate6")][1,]*100,use.names = F),
                                se=unlist(patient_proposal_prob_trans[[1]][patient_proposal_prob_trans[[1]]$time>=3 ,c("se1","se2","se3","se4","se5","se6")][1,]*100,use.names = F))
diagnosis_3_proposal <- ggplot(ptrans3_proposal,aes(x=States,y=proba)) +
  geom_bar(stat="identity", fill=statecols) +
  geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log(proba) + crit * se / proba)),
                width=.2, # Width of the error bars
                position=position_dodge(.9)) +
  geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",
            size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +
  theme_minimal() + theme(axis.text=element_text(size=17),axis.title=element_text(size=20),plot.title=element_text(size=20,face="bold")) +
  ggtitle ("Patient Outcome 3 years from Diagnosis")

```

```

    position=position_dodge(.9))+  

      geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",  

      size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +  

      theme_minimal() + theme(axis.text=element_text(size=17),axis.title=ele  

      ment_text(size=20),plot.title=element_text(size=20,face="bold")) +  

      ggtitle ("Patient Outcome 3 years from Diagnosis")

ptrans1_CR_proposal <- data.frame(States=c("Alive in CR","Alive with rel  

apse","Death in CR","Death with Relapse"),  

      proba=unlist(patient_proposal_prob_trans[[2]][patient_prop  

osal_prob_trans[[2]]$time>=1 ,c("pstate2","pstate3","pstate4","pstate5"  

)][1,]*100,use.names = F),  

      se=unlist(patient_proposal_prob_trans[[2]][patient_proposa  

l_prob_trans[[2]]$time>=1 ,c("se2","se3","se4","se5")][1,]*100,use.names  

= F)
CR_1_proposal <- ggplot(ptrans1_CR_proposal,aes(x=States,y=proba)) + geo  

m_bar(stat="identity", fill=statecols[2:5]) +  

geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log  

(proba) + crit * se / proba)),  

      width=.2, # Width of the error bars  

      position=position_dodge(.9))+  

      geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",  

      size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +  

      theme_minimal() + theme(axis.text=element_text(size=17),axis.title=ele  

      ment_text(size=20),plot.title=element_text(size=20,face="bold")) +  

      ggtitle ("Patient Outcome 1 year from CR")

ptrans3_CR_proposal <- data.frame(States=c("Alive in CR","Alive with rel  

apse","Death in CR","Death with Relapse"),  

      proba=unlist(patient_proposal_prob_trans[[2]][patient_prop  

osal_prob_trans[[2]]$time>=3 ,c("pstate2","pstate3","pstate4","pstate5"  

)][1,]*100,use.names = F),  

      se=unlist(patient_proposal_prob_trans[[2]][patient_proposa  

l_prob_trans[[2]]$time>=3 ,c("se2","se3","se4","se5")][1,]*100,use.names  

= F)
CR_3_proposal <- ggplot(ptrans3_CR_proposal,aes(x=States,y=proba)) + geo  

m_bar(stat="identity", fill=statecols[2:5]) +  

geom_errorbar(aes(ymin=exp(log(proba) - crit * se / proba), ymax=exp(log  

(proba) + crit * se / proba)),  

      width=.2, # Width of the error bars  

      position=position_dodge(.9))+  

      geom_text(aes(label=round(proba,0)), vjust=0.5,hjust=4, color="white",  

      size=8.5,face="bold") + coord_flip() + ylab("Probability (%)") +  

      theme_minimal() + theme(axis.text=element_text(size=17),axis.title=ele  

      ment_text(size=20),plot.title=element_text(size=20,face="bold")) +  

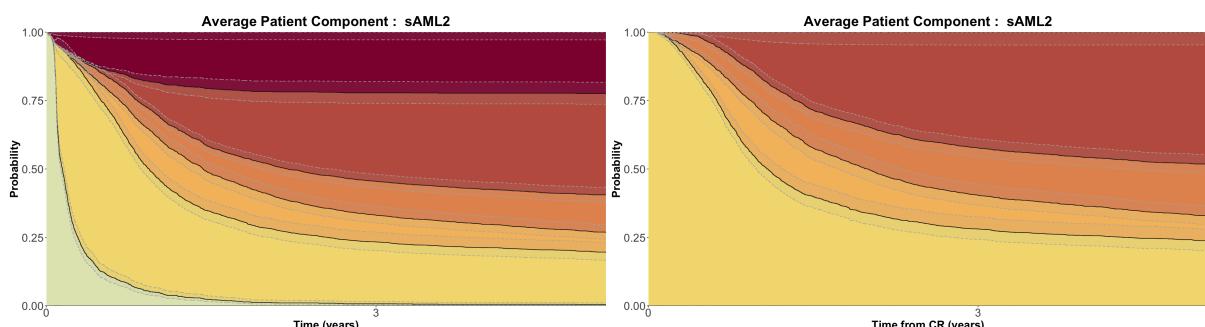
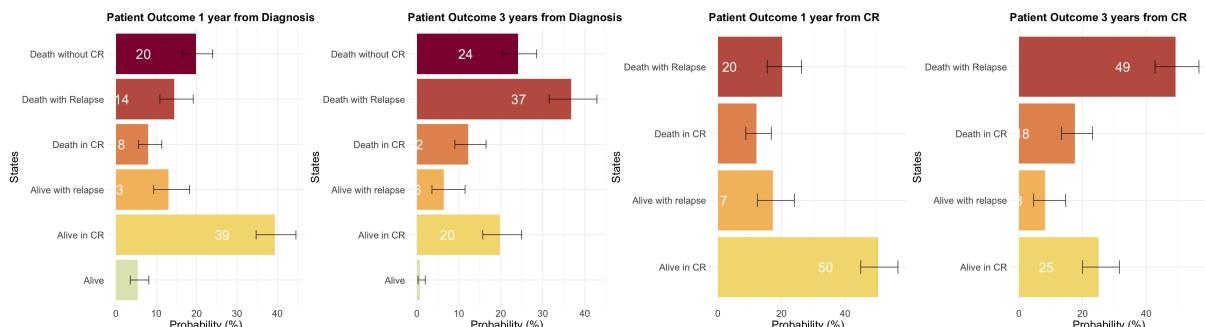
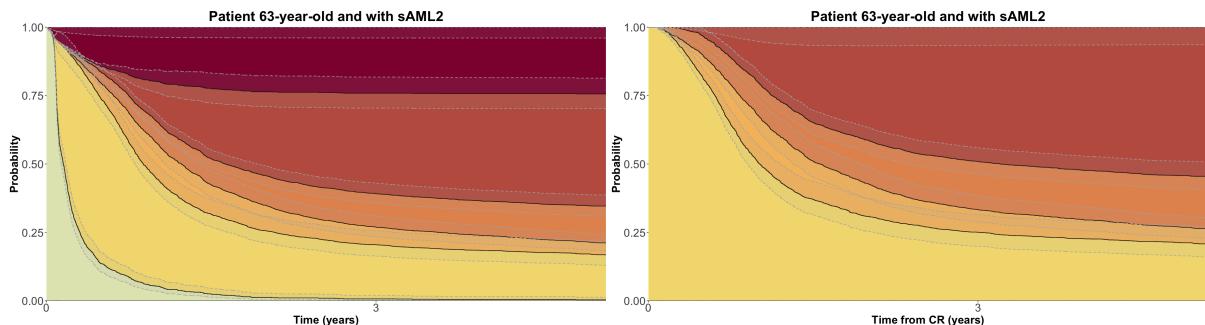
      ggtitle ("Patient Outcome 3 years from CR")

grid.arrange(diagnosis_1_proposal,diagnosis_3_proposal,CR_1_proposal,CR_3  

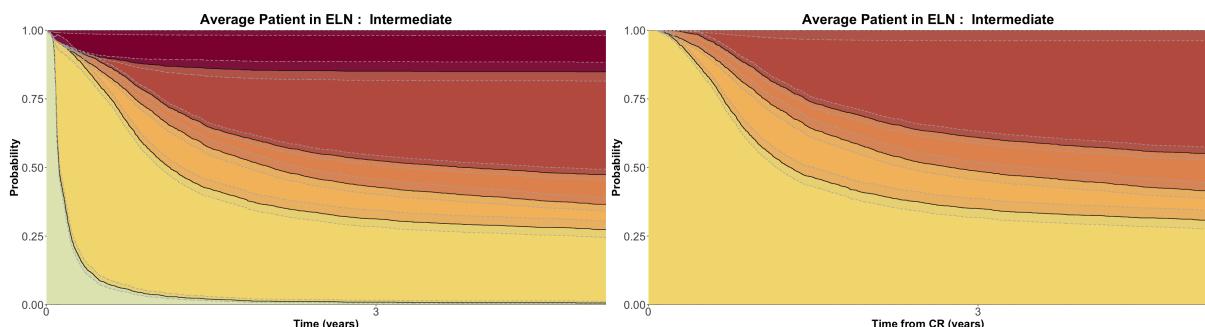
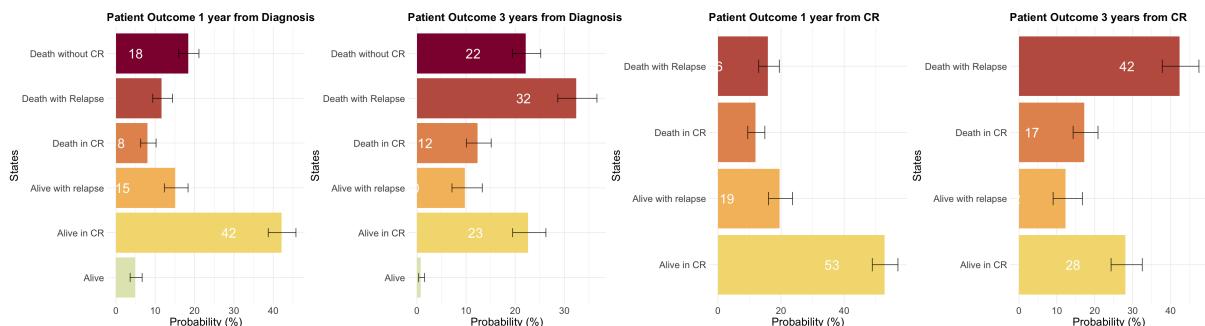
_proposal,nrow=1)

# dev.off()

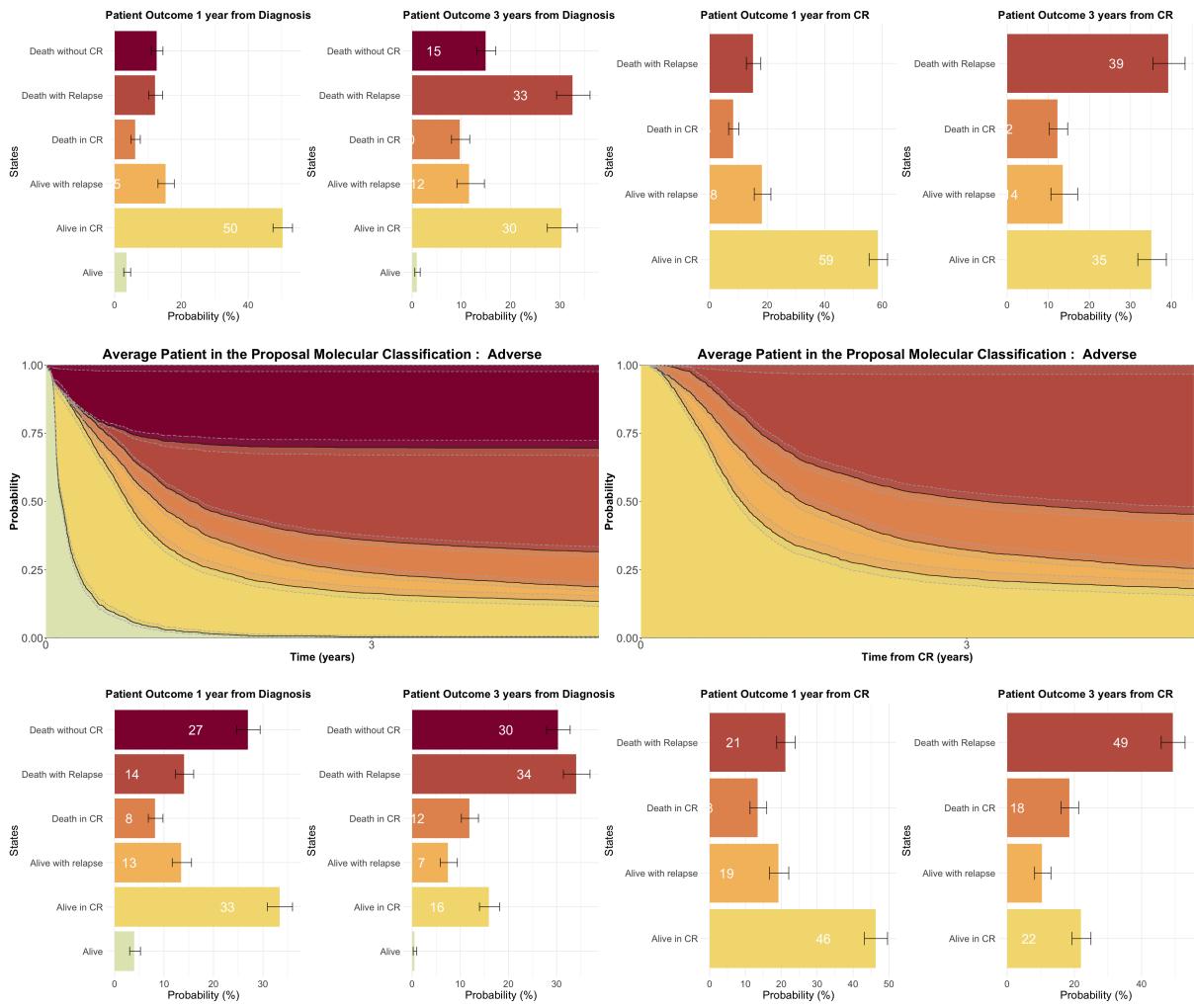
```



[1] "Intermediate"



[1] "Adverse"



6. Patient Centric covariate effects.

```
In [25]: # pdf("Output_patient/Main/Page2_example_with_median_patient.pdf",width=40,height=10)
coef_data <- data.frame(summary(cfull)$coefficients)
set_notebook_plot_size(42,12)
p <- list()
count <- 1
for (trans in c(".1",".2",".3",".4",".5")){
  l <- coef_data[endsWith(rownames(coef_data),trans),]
  rownames(l) <- gsub('.{0,2}$', ' ', rownames(l))

  k <- merge(l,user_data_simplified,by=0)
  k$average_patient <- 0
  k$average_patient <- ifelse(k$Row.names %in% names(covariates_median),covariates_median[k$Row.names],0)
  k["BetaZs"] <- k$coef * (k$value-k$average_patient)
  k["BetaZs"] <- k["BetaZs"] / max(abs(k["BetaZs"]))
  k <- k[(k$BetaZs!=0 & !is.na(k$BetaZs)),]
  #k["BetaZ"] <- k$coef * (k$value)
  #k["BetaZ"] <- k["BetaZ"] / max(abs(k["BetaZ"]))
  k <- k[(k$BetaZs!=0 & !is.na(k$BetaZs)),]

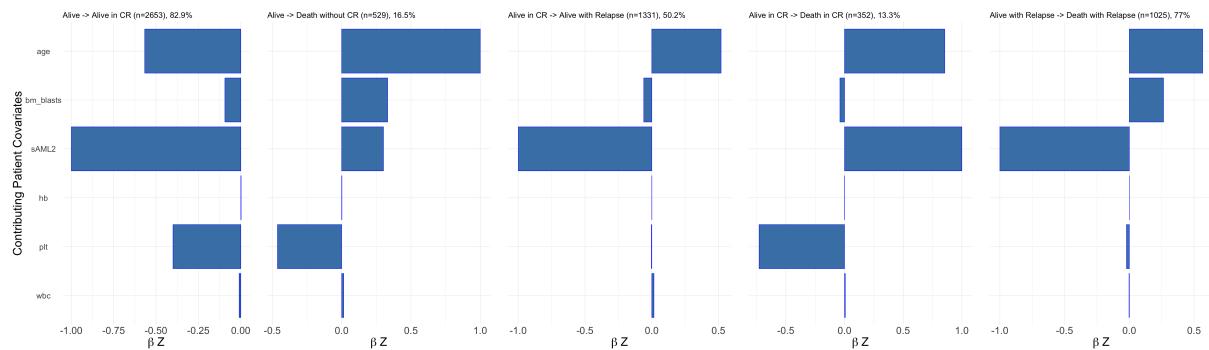
  k$Row.names <- str_remove(k$Row.names,"full_component_")
#  k <- k[order(k["BetaZ"],decreasing=T),]

  title <- ifelse(trans=="1","Alive -> Alive in CR (n=2653), 82.9%",
                  ifelse(trans=="2","Alive -> Death without CR (n=529), 16.5%",
                         ifelse(trans=="3","Alive in CR -> Alive with Relapse (n=1331), 50.2%",
                               ifelse(trans=="4","Alive in CR -> Death in CR (n=352), 13.3%","Alive with Relapse -> Death with Relapse (n=1025), 77%")))
  xlab <- ifelse(trans=="1","Contributing Patient Covariates","");
  p[[trans]] <- ggplot(data = k,aes(x = factor(Row.names,level=rev(Row.names)),y= BetaZs)) + geom_bar(stat="identity",colour="blue",fill="steelblue") + coord_flip() + xlab(xlab) + ylab(expression(beta ~ z)) + ggtitle(title) +
    theme_minimal() + theme(axis.title=element_text(size=30),axis.text.y=element_blank(),plot.title=element_text(size=20),axis.text.x=element_text(size=25))

  if(trans=="1"){
    p[[trans]] <- p[[trans]] + theme(axis.text.y=element_text(size=20))
  }
  #+ ifelse(trans=="1",theme(axis.text.x=element_text(size=20)),theme(axis.text.y=element_blank(),axis.ticks.y=element_blank()))
  #ifelse(trans=="1",theme(axis.text.x=element_blank()),theme(axis.text.x=element_text(size=20)))

}
do.call("grid.arrange",c(p,nrow=1))
```

Patient_Centric_Web_App



**

**

**

**

**

**

**