

# Ayanambakkam et al\_ctDNA in UTUC Clinical Analysis

Code ▾

```
library(swimplot) library(grid) library(gtable) library(readr) library(mosaic) library(dplyr) library(survival) library(survminer) library(ggplot2)
library(scales) library(coxphf) library(ggthemes) library(tidyverse) library(gtsuammary) library(flextable) library(parameters) library(car)
library(ComplexHeatmap) library(tidyverse) library(readxl) library(janitor) library(DT) library(pROC) library(rms)
```

#ctDNA Detection Rates by Window and Stages

Hide

```
#ctDNA at MRD by Stage
rm(list=ls())
setwd("~/Downloads")
circ_data <- read.csv("RWE UTUC_Clinical Data 102025.csv")
circ_data <- circ_data[circ_data$Final.cohort=="TRUE",]
circ_data <- circ_data[circ_data$ctDNA.MRD!="",]
circ_data$ctDNA.MRD <- factor(circ_data$ctDNA.MRD, levels=c("NEGATIVE","POSITIVE"))
circ_data <- subset(circ_data, ctDNA.MRD %in% c("NEGATIVE", "POSITIVE"))
circ_data$Stage <- factor(circ_data$Stage, levels=c("0/I","II","III"))
positive_counts_by_stage <- aggregate(circ_data$ctDNA.MRD == "POSITIVE", by=list(circ_data$Stage), FUN=sum)
total_counts_by_stage <- aggregate(circ_data$ctDNA.MRD, by=list(circ_data$Stage), FUN=length)
combined_data <- data.frame(
  Stage = total_counts_by_stage$Group.1,
  Total_Count = total_counts_by_stage$x,
  Positive_Count = positive_counts_by_stage$x,
  Rate = (positive_counts_by_stage$x / total_counts_by_stage$x) * 100 # Convert to percentage
)
combined_data$Rate <- sprintf("%.2f%%", combined_data$Rate)
overall_total_count <- nrow(circ_data)
overall_positive_count <- nrow(circ_data[circ_data$ctDNA.MRD == "POSITIVE",])
overall_positivity_rate <- (overall_positive_count / overall_total_count) * 100 # Convert to percentage
overall_row <- data.frame(
  Stage = "Overall",
  Total_Count = overall_total_count,
  Positive_Count = overall_positive_count,
  Rate = sprintf("%.2f%%", overall_positivity_rate)
)
combined_data <- rbind(combined_data, overall_row)
print(combined_data)
```

Stage <fctr>	Total_Count <int>	Positive_Count <int>	Rate <chr>
0/I	6	2	33.33%
II	6	4	66.67%
III	16	9	56.25%
Overall	28	15	53.57%
4 rows			

Hide

```
#ctDNA at MRD by NAC status
rm(list=ls())
setwd("~/Downloads")
circ_data <- read.csv("RWE UTUC_Clinical Data 102025.csv")
circ_data <- circ_data[circ_data$Final.cohort=="TRUE",]
circ_data <- circ_data[circ_data$ctDNA.MRD!="",]
circ_data$ctDNA.MRD <- factor(circ_data$ctDNA.MRD, levels=c("NEGATIVE","POSITIVE"))
circ_data <- subset(circ_data, ctDNA.MRD %in% c("NEGATIVE", "POSITIVE"))
circ_data$NAC <- factor(circ_data$NAC, levels=c("FALSE","TRUE"))
positive_counts_by_stage <- aggregate(circ_data$ctDNA.MRD == "POSITIVE", by=list(circ_data$NAC), FUN=sum)
total_counts_by_stage <- aggregate(circ_data$ctDNA.MRD, by=list(circ_data$NAC), FUN=length)
combined_data <- data.frame(
  Stage = total_counts_by_stage$Group.1,
  Total_Count = total_counts_by_stage$x,
  Positive_Count = positive_counts_by_stage$x,
  Rate = (positive_counts_by_stage$x / total_counts_by_stage$x) * 100 # Convert to percentage
)
combined_data$Rate <- sprintf("%.2f%%", combined_data$Rate)
overall_total_count <- nrow(circ_data)
overall_positive_count <- nrow(circ_data[circ_data$ctDNA.MRD == "POSITIVE",])
overall_positivity_rate <- (overall_positive_count / overall_total_count) * 100 # Convert to percentage
overall_row <- data.frame(
  Stage = "Overall",
  Total_Count = overall_total_count,
  Positive_Count = overall_positive_count,
  Rate = sprintf("%.2f%%", overall_positivity_rate)
)
combined_data <- rbind(combined_data, overall_row)
print(combined_data)
```

Stage <fctr>	Total_Count <int>	Positive_Count <int>	Rate <chr>
FALSE	23	12	52.17%
TRUE	5	3	60.00%
Overall	28	15	53.57%

3 rows

[Hide](#)

```
#ctDNA at Surveillance by Stage
rm(list=ls())
setwd("~/Downloads")
circ_data <- read.csv("RWE UTUC_Clinical Data 102025.csv")
circ_data <- circ_data[circ_data$Final.cohort=="TRUE",]
circ_data <- circ_data[circ_data$ctDNA.Surveillance!="",]
circ_data$ctDNA.Surveillance <- factor(circ_data$ctDNA.Surveillance, levels=c("NEGATIVE","POSITIVE"))
circ_data$Stage <- factor(circ_data$Stage, levels=c("0/I","II","III"))
positive_counts_by_stage <- aggregate(circ_data$ctDNA.Surveillance == "POSITIVE", by=list(circ_data$Stage), FUN=sum)
total_counts_by_stage <- aggregate(circ_data$ctDNA.Surveillance, by=list(circ_data$Stage), FUN=length)
combined_data <- data.frame(
  Stage = total_counts_by_stage$Group.1,
  Total_Count = total_counts_by_stage$x,
  Positive_Count = positive_counts_by_stage$x,
  Rate = (positive_counts_by_stage$x / total_counts_by_stage$x) * 100 # Convert to percentage
)
combined_data$Rate <- sprintf("%.2f%%", combined_data$Rate)
overall_total_count <- nrow(circ_data)
overall_positive_count <- nrow(circ_data[circ_data$ctDNA.Surveillance == "POSITIVE",])
overall_positivity_rate <- (overall_positive_count / overall_total_count) * 100 # Convert to percentage
overall_row <- data.frame(
  Stage = "Overall",
  Total_Count = overall_total_count,
  Positive_Count = overall_positive_count,
  Rate = sprintf("%.2f%%", overall_positivity_rate)
)
combined_data <- rbind(combined_data, overall_row)
print(combined_data)
```

Stage <fctr>	Total_Count <int>	Positive_Count <int>	Rate <chr>
O/I	9	3	33.33%
II	9	7	77.78%
III	22	12	54.55%
Overall	40	22	55.00%

4 rows

Hide

```
#ctDNA at anytime by Stage
rm(list=ls())
setwd("~/Downloads")
circ_data <- read.csv("RWE_UTUC_Clinical_Data_102025.csv")
circ_data <- circ_data[circ_data$Final.cohort=="TRUE",]
circ_data <- circ_data[circ_data$ctDNA.anytime!="",]
circ_data$ctDNA.anytime <- factor(circ_data$ctDNA.anytime, levels=c("NEGATIVE","POSITIVE"))
circ_data$Stage <- factor(circ_data$Stage, levels=c("O/I","II","III"))
positive_counts_by_stage <- aggregate(circ_data$ctDNA.anytime == "POSITIVE", by=list(circ_data$Stage), FUN=sum)
total_counts_by_stage <- aggregate(circ_data$ctDNA.anytime, by=list(circ_data$Stage), FUN=length)
combined_data <- data.frame(
  Stage = total_counts_by_stage$Group.1,
  Total_Count = total_counts_by_stage$x,
  Positive_Count = positive_counts_by_stage$x,
  Rate = (positive_counts_by_stage$x / total_counts_by_stage$x) * 100 # Convert to percentage
)
combined_data$Rate <- sprintf("%.2f%%", combined_data$Rate)
overall_total_count <- nrow(circ_data)
overall_positive_count <- nrow(circ_data[circ_data$ctDNA.anytime == "POSITIVE",])
overall_positivity_rate <- (overall_positive_count / overall_total_count) * 100 # Convert to percentage
overall_row <- data.frame(
  Stage = "Overall",
  Total_Count = overall_total_count,
  Positive_Count = overall_positive_count,
  Rate = sprintf("%.2f%%", overall_positivity_rate)
)
combined_data <- rbind(combined_data, overall_row)
print(combined_data)
```

Stage <fctr>	Total_Count <int>	Positive_Count <int>	Rate <chr>
O/I	9	3	33.33%
II	10	8	80.00%
III	28	22	78.57%
Overall	47	33	70.21%

4 rows

#Demographics Table

Hide

```
rm(list=ls())
setwd("~/Downloads")
circ_data <- read.csv("RWE_UTUC_Clinical_Data_102025.csv")
circ_data <- circ_data[circ_data$Final.cohort=="TRUE",]

circ_data_subset <- circ_data %>%
  select(
    Age,
    Sex,
    Stage,
    NAC,
    ACT,
    ACT.type,
    RFS.Event,
    OS.Event,
    OS.months) %>%
  mutate(
    Age = as.numeric(Age),
    Sex = factor(Sex),
    Stage = factor(Stage),
    NAC = factor(NAC),
    ACT = factor(ACT),
    ACT.type = factor(ACT.type),
    RFS.Event = factor(RFS.Event, levels = c("FALSE", "TRUE"), labels = c("No Recurrence", "Recurrence")),
    OS.Event = factor(OS.Event, levels = c("FALSE", "TRUE"), labels = c("Alive", "Deceased")),
    OS.months = as.numeric(OS.months))
table1 <- circ_data_subset %>%
  tbl_summary(
    statistic = list(
      all_continuous() ~ "{median} ({min} - {max})",
      all_categorical() ~ "{n} ({p}%)" ) %>%
    bold_labels()
  )
table1
```

Characteristic	N = 47 <sup>†</sup>
<b>Age</b>	77 (47 - 92)
<b>Sex</b>	
Female	17 (36%)
Male	30 (64%)
<b>Stage</b>	
0/I	9 (19%)
II	10 (21%)
III	28 (60%)
<b>NAC</b>	
FALSE	40 (85%)
TRUE	7 (15%)
<b>ACT</b>	
FALSE	13 (28%)
TRUE	34 (72%)
<b>ACT.type</b>	
	13 (28%)
Chemotherapy	18 (38%)
Immunotherapy	16 (34%)
<b>RFS.Event</b>	
No Recurrence	23 (49%)
Recurrence	24 (51%)
<sup>†</sup> Median (Min - Max); n (%)	

Characteristic	N = 47 <sup>†</sup>
<b>OS.Event</b>	
Alive	41 (87%)
Deceased	6 (13%)
<b>OS.months</b>	21 (3 - 48)
<sup>†</sup> Median (Min - Max); n (%)	

Hide

```
fit1 <- as_flex_table(
  table1,
  include = everything(),
  return_calls = FALSE)
fit1
```

Characteristic	N = 47 <sup>†</sup>
<b>Age</b>	77 (47 - 92)
<b>Sex</b>	
Female	17 (36%)
Male	30 (64%)
<b>Stage</b>	
0/I	9 (19%)
II	10 (21%)
III	28 (60%)
<b>NAC</b>	
FALSE	40 (85%)
TRUE	7 (15%)
<b>ACT</b>	
FALSE	13 (28%)
TRUE	34 (72%)
<b>ACT.type</b>	
	13 (28%)
Chemotherapy	18 (38%)
Immunotherapy	16 (34%)
<b>RFS.Event</b>	
No Recurrence	23 (49%)
Recurrence	24 (51%)
<b>OS.Event</b>	
Alive	41 (87%)
Deceased	6 (13%)
<b>OS.months</b>	21 (3 - 48)

<sup>†</sup>Median (Min - Max); n (%)

Hide

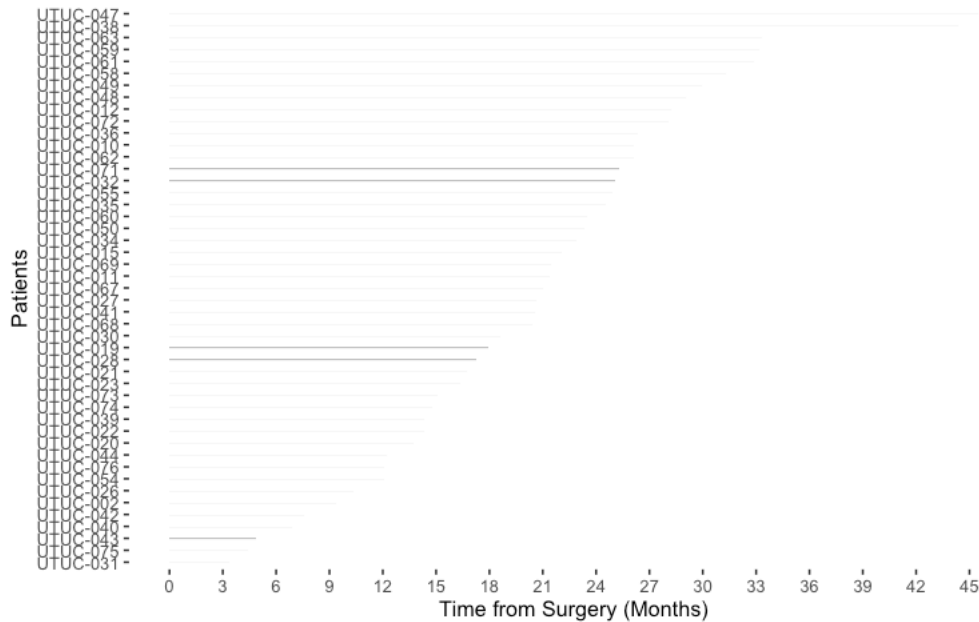
```
save_as_docx(fit1, path= "~/Downloads/1. Demographics Table_RWE UTUC.docx")
```

#Overview plot

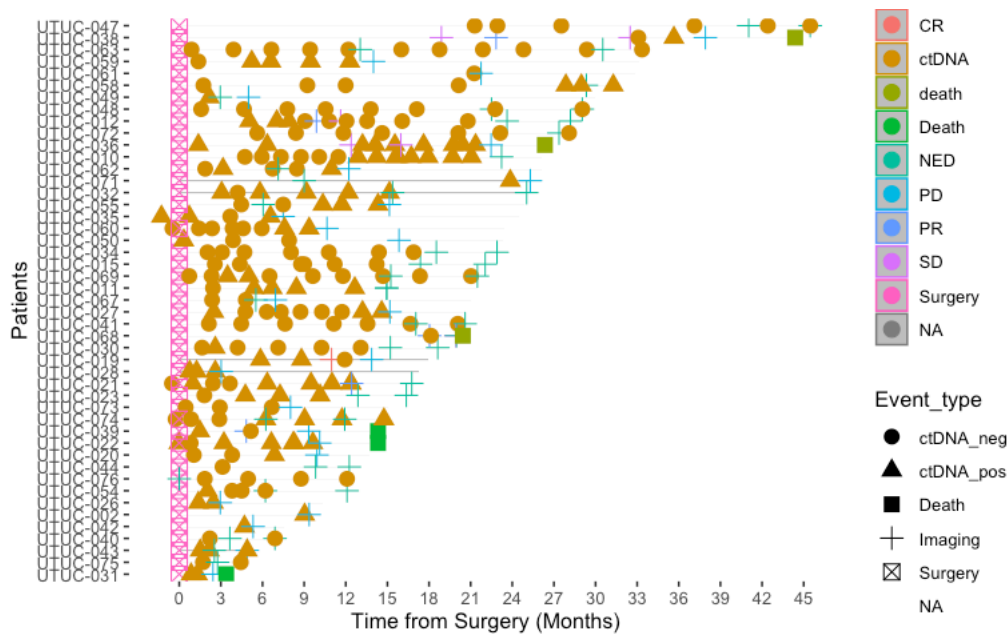
Hide

```
rm(list=ls())
setwd("~/Downloads")
clinstage<- read.csv("RWE UTUC_OP 102025.csv")
clinstage <- clinstage[clinstage$Final.cohort=="TRUE",]
clinstage_df<- as.data.frame(clinstage)

#Display the swimmer plot with the label box
oplot<-swimmer_plot(df=clinstage_df,
                    id='PatientName',
                    end='fu.diff.months',
                    fill='gray',
                    width=.01,)
oplot <- oplot + theme(panel.border = element_blank())
oplot <- oplot + scale_y_continuous(breaks = seq(0, 96, by = 3))
oplot <- oplot + labs(x ="Patients" , y="Time from Surgery (Months)")
oplot
```


[Hide](#)

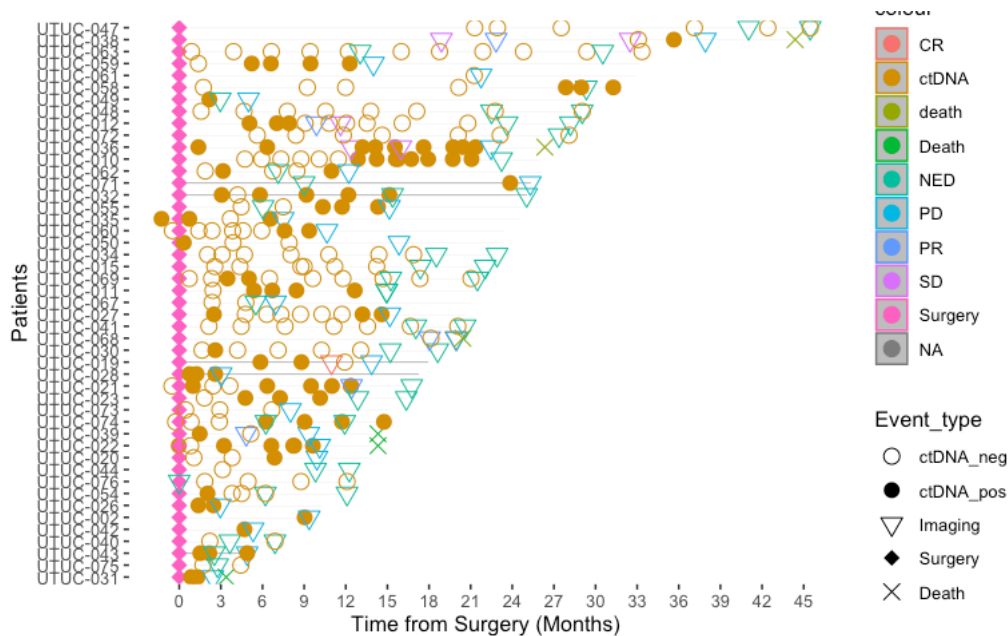
```
##plot events
oplot_ev1 <- oplot + swimmer_points(df_points=clinstage_df,
                                   id='PatientName',
                                   time='date.diff.months',
                                   name_shape = 'Event_type',
                                   name_col = 'Event',
                                   size=3.5,fill='black',
                                   #col='darkgreen'
                                   )
oplot_ev1
```



Hide

```
#Shape customization to Event_type
```

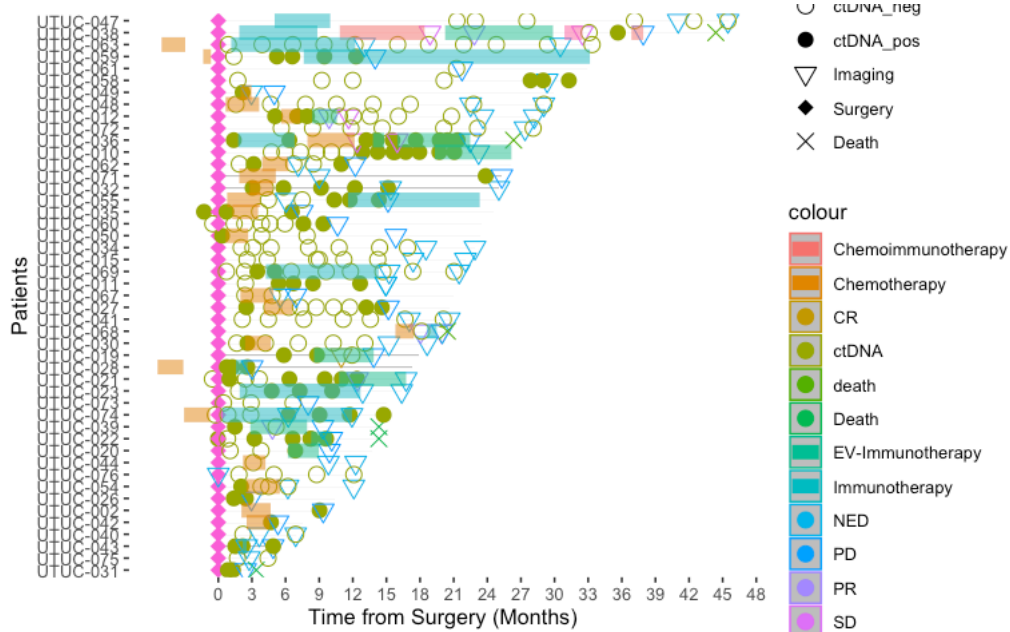
```
oplot_ev1.1 <- oplot_ev1 + ggplot2::scale_shape_manual(name="Event_type", values=c(1,16,6,18,4), breaks=c('ctDNA_neg', 'ctDNA_pos', 'Imaging', 'Surgery', 'Death'))
oplot_ev1.1
```



Hide

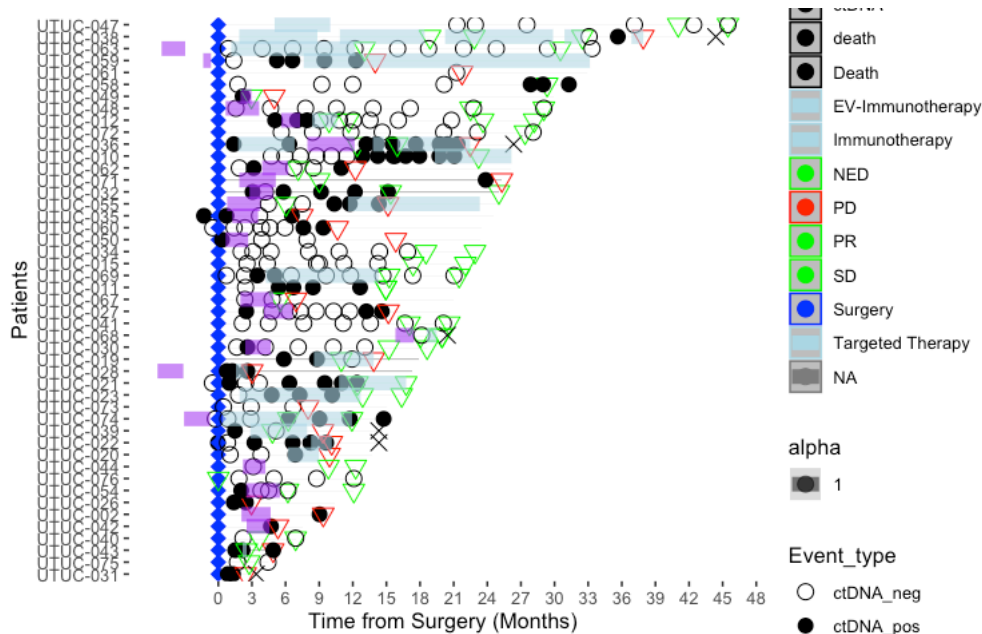
```
#plot treatment
```

```
oplot_ev2 <- oplot_ev1.1 + swimmer_lines(df_lines=clinstage_df,
                                          id='PatientName',
                                          start='Tx_start.months',
                                          end='Tx_end.months',
                                          name_col='Tx_type',
                                          size=3.5,
                                          name_alpha = 1.0)
oplot_ev2 <- oplot_ev2 + guides(linetype = guide_legend(override.aes = list(size = 5, color = "black")))
oplot_ev2
```



Hide

```
#colour customization
oplot_ev2.2 <- oplot_ev2 + ggplot2::scale_color_manual(name="Event", values=c( "lightblue","purple","green", "black", "black", "black", "lightblue","lightblue","green","red","green","green","blue","lightblue", "blue", "blue"))
oplot_ev2.2
```



#RFS by ctDNA status at MRD Window

Hide

```
rm(list=ls())
setwd("~/Downloads")
circ_data <- read.csv("RWE UTUC_Clinical Data 102025.csv")
circ_data <- circ_data[circ_data$Final.cohort=="TRUE",]
circ_data <- circ_data[circ_data$ctDNA.MRD!="",]
circ_data$RFS.MRD <- circ_data$RFS - circ_data$ctDNA.timing
circ_data$RFS.MRD.months <- circ_data$RFS.MRD / 30.437

survfit(Surv(time = circ_data$RFS.MRD.months, event = circ_data$RFS.Event)~ctDNA.MRD, data = circ_data)
```



```
Call: survfit(formula = Surv(time = circ_data$RFS.MRD.months, event = circ_data$RFS.Event) ~
      ctDNA.MRD, data = circ_data)
```

	n	events	median	0.95LCL	0.95UCL
ctDNA.MRD=NEGATIVE	13	3	NA	12.62	NA
ctDNA.MRD=POSITIVE	15	11	9.07	6.77	NA

Hide

```
event_summary <- circ_data %>%
  group_by(ctDNA.MRD) %>%
  summarise(
    Total = n(),
    Events = sum(RFS.Event),
    Fraction = Events / n(),
    Percentage = (Events / n()) * 100
  )
print(event_summary)
```

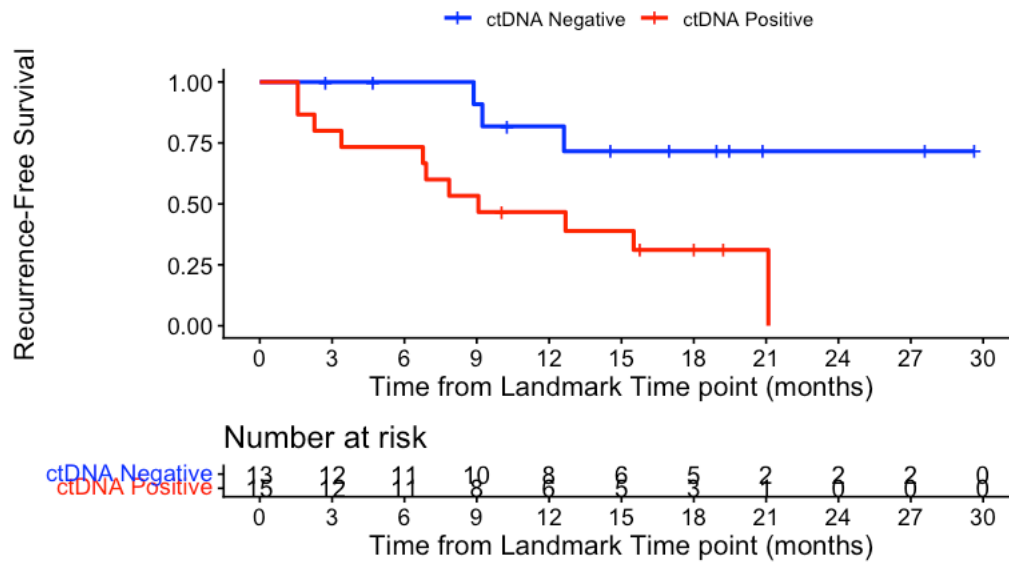
ctDNA.MRD	Total	Events	Fraction	Percentage
<chr>	<int>	<int>	<dbl>	<dbl>
NEGATIVE	13	3	0.2307692	23.07692
POSITIVE	15	11	0.7333333	73.33333

2 rows

Hide

```
surv_object <-Surv(time = circ_data$RFS.MRD.months, event = circ_data$RFS.Event)
KM_curve <- survfit(surv_object ~ ctDNA.MRD, data = circ_data,conf.int=0.95,conf.type="log-log")
ggsurvplot(KM_curve, data = circ_data, pval = FALSE, conf.int = FALSE, risk.table = TRUE, break.time.by=3, palett
e=c("blue","red"), title="RFS - ctDNA at the MRD Window", ylab= "Recurrence-Free Survival", xlab="Time from Landm
ark Time point (months)", legend.labs=c("ctDNA Negative", "ctDNA Positive"), legend.title="")
```

## RFS - ctDNA at the MRD Window



Hide

```
summary(KM_curve, times= c(0, 12, 18))
```

```
Call: survfit(formula = surv_object ~ ctDNA.MRD, data = circ_data,
  conf.int = 0.95, conf.type = "log-log")
```

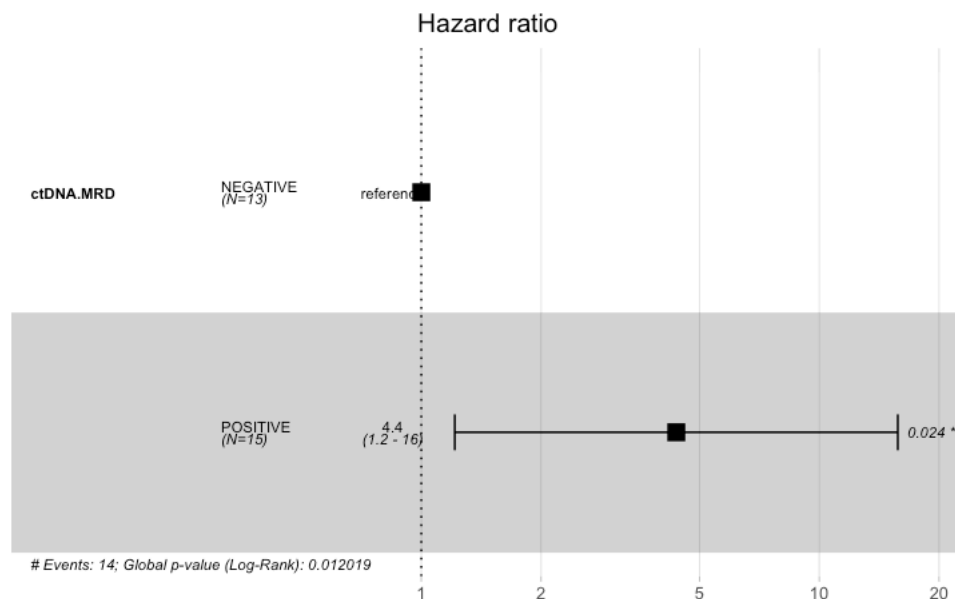
ctDNA.MRD=NEGATIVE								
time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI		
0	13	0	1.000	0.000	1.000	1.000		
12	8	2	0.818	0.116	0.447	0.951		
18	5	1	0.716	0.140	0.350	0.899		

ctDNA.MRD=POSITIVE								
time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI		
0	15	0	1.000	0.000	1.000	1.000		
12	6	8	0.467	0.129	0.212	0.687		
18	3	2	0.311	0.124	0.102	0.550		

Hide

```
circ_data$ctDNA.MRD <- factor(circ_data$ctDNA.MRD, levels=c("NEGATIVE", "POSITIVE"))
cox_fit <- coxph(surv_object ~ ctDNA.MRD, data=circ_data)
ggforest(cox_fit, data = circ_data)
```



Hide

```
summary(cox_fit)
```

Call:

```
coxph(formula = surv_object ~ ctDNA.MRD, data = circ_data)
```

n= 28, number of events= 14

```

              coef exp(coef) se(coef)      z Pr(>|z|)
ctDNA.MRDPOSITIVE 1.4752    4.3720  0.6539 2.256  0.0241 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

              exp(coef) exp(-coef) lower .95 upper .95
ctDNA.MRDPOSITIVE    4.372    0.2287    1.214    15.75
```

Concordance= 0.677 (se = 0.057 )

Likelihood ratio test= 6.31 on 1 df, p=0.01

Wald test = 5.09 on 1 df, p=0.02

Score (logrank) test = 6.05 on 1 df, p=0.01

Hide

```
cox_fit_summary <- summary(cox_fit)
```

#Extract values for HR, 95% CI, and p-value

```
HR <- cox_fit_summary$coefficients[2]
```

```
lower_CI <- cox_fit_summary$conf.int[3]
```

```
upper_CI <- cox_fit_summary$conf.int[4]
```

```
p_value <- cox_fit_summary$coefficients[5]
```

```
label_text <- paste0("HR = ", round(HR, 2), " (", round(lower_CI, 2), "-", round(upper_CI, 2), "); p = ", round(p_value, 3))
```

```
print(label_text)
```

```
[1] "HR = 4.37 (1.21-15.75); p = 0.024"
```

Hide

```
circ_data$ctDNA.MRD <- factor(circ_data$ctDNA.MRD, levels = c("NEGATIVE", "POSITIVE"), labels = c("Negative", "Positive"))
```

```
circ_data$RFS.Event <- factor(circ_data$RFS.Event, levels = c("FALSE", "TRUE"), labels = c("No Recurrence", "Recurrence"))
```

```
contingency_table <- table(circ_data$ctDNA.MRD, circ_data$RFS.Event)
```

```
chi_square_test <- chisq.test(contingency_table)
```

```
print(chi_square_test)
```

Pearson's Chi-squared test with Yates' continuity correction

data: contingency\_table

X-squared = 5.1692, df = 1, p-value = 0.02299

Hide

```
fisher_exact_test <- fisher.test(contingency_table)
```

```
print(fisher_exact_test)
```

Fisher's Exact Test for Count Data

data: contingency\_table

p-value = 0.0213

alternative hypothesis: true odds ratio is not equal to 1

95 percent confidence interval:

1.28780 74.44434

sample estimates:

odds ratio

8.328277

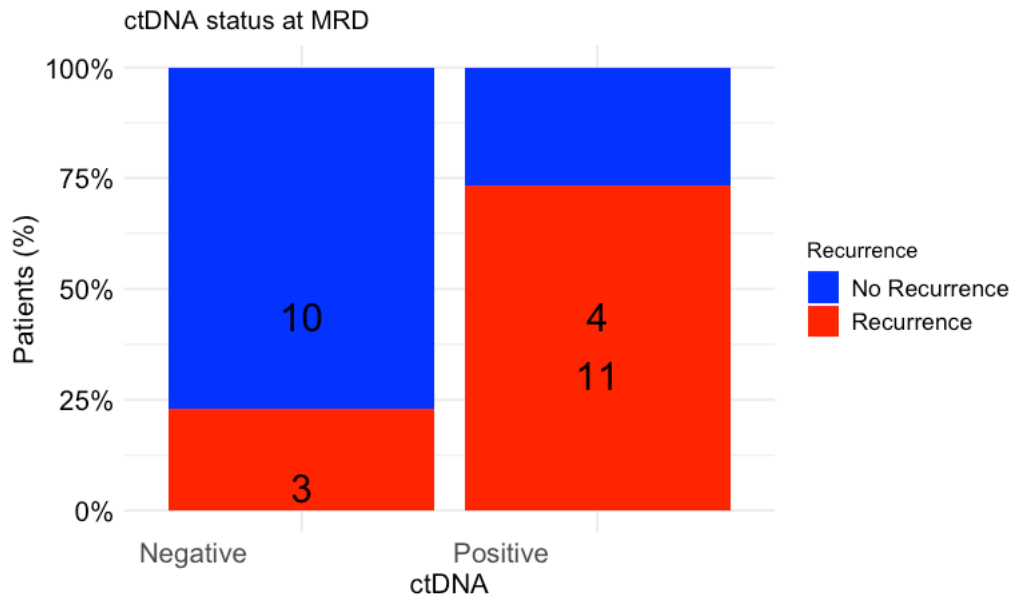
Hide

```
print(contingency_table)
```

	No Recurrence	Recurrence
Negative	10	3
Positive	4	11

Hide

```
table_df <- as.data.frame(contingency_table)
table_df$Total <- ave(table_df$Freq, table_df$Var1, FUN = sum)
table_df$Percentage <- table_df$Freq / table_df$Total
table_df$MiddlePercentage <- table_df$Percentage / 2
ggplot(table_df, aes(x = Var1, y = Percentage, fill = Var2)) +
  geom_bar(stat = "identity") +
  geom_text(aes(y = MiddlePercentage, label = Freq), position = "stack", color = "black", vjust = 1.5, size = 7)
+
  theme_minimal() +
  labs(title = "ctDNA status at MRD",
       x = "ctDNA",
       y = "Patients (%)",
       fill = "Recurrence",
       caption = paste("Fisher's exact test p-value: ", format.pval(fisher_exact_test$p.value))) +
  scale_y_continuous(labels = scales::percent_format()) +
  scale_fill_manual(values = c("No Recurrence" = "blue", "Recurrence" = "red")) + # define custom colors
  theme(axis.text.x = element_text(angle = 0, hjust = 1.5, size = 14), # increase x-axis text size
        axis.text.y = element_text(size = 14, color = "black"), # increase y-axis text size
        axis.title.x = element_text(size = 14, color = "black"), # increase x-axis label size
        axis.title.y = element_text(size = 14, color = "black"), # increase y-axis label size
        legend.text = element_text(size = 12, color = "black")) # increase Recurrence label size
```



Fisher's exact test p-value: 0.0213

#RFS by ctDNA status at Surveillance Window

Hide

```
rm(list=ls())
setwd("~/Downloads")
circ_data <- read.csv("RWE UTUC_Clinical Data 102025.csv")
circ_data <- circ_data[circ_data$Final.cohort=="TRUE",]
circ_data <- circ_data[circ_data$ctDNA.Surveillance!="",]

survfit(Surv(time = circ_data$RFS.months, event = circ_data$RFS.Event)~ctDNA.Surveillance, data = circ_data)
```

Call: survfit(formula = Surv(time = circ\_data\$RFS.months, event = circ\_data\$RFS.Event) ~  
ctDNA.Surveillance, data = circ\_data)

	n	events	median	0.95LCL	0.95UCL
ctDNA.Surveillance=NEGATIVE	18	3	NA	NA	NA
ctDNA.Surveillance=POSITIVE	22	17	13.9	10.1	NA

Hide

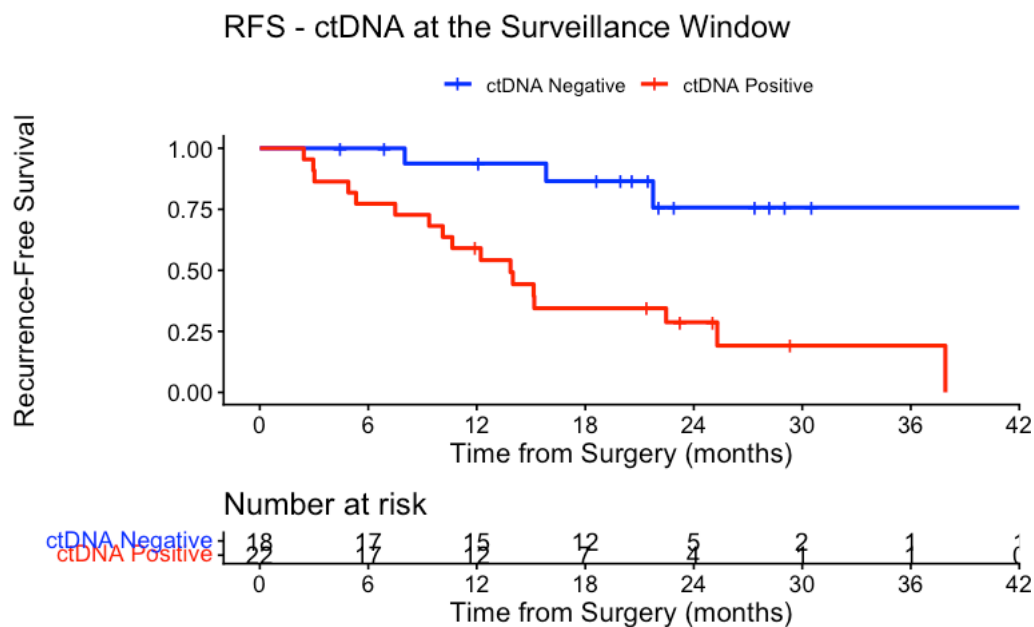
```
event_summary <- circ_data %>%
  group_by(ctDNA.Surveillance) %>%
  summarise(
    Total = n(),
    Events = sum(RFS.Event),
    Fraction = Events / n(),
    Percentage = (Events / n()) * 100
  )
print(event_summary)
```

ctDNA.Surveillance <chr>	Total <int>	Events <int>	Fraction <dbl>	Percentage <dbl>
NEGATIVE	18	3	0.1666667	16.66667
POSITIVE	22	17	0.7727273	77.27273

2 rows

Hide

```
surv_object <- Surv(time = circ_data$RFS.months, event = circ_data$RFS.Event)
KM_curve <- survfit(surv_object ~ ctDNA.Surveillance, data = circ_data, conf.int=0.95, conf.type="log-log")
ggsurvplot(KM_curve, data = circ_data, pval = FALSE, conf.int = FALSE, risk.table = TRUE, break.time.by=6, palette=c("blue","red"), title="RFS - ctDNA at the Surveillance Window", ylab= "Recurrence-Free Survival", xlab="Time from Surgery (months)", legend.labs=c("ctDNA Negative", "ctDNA Positive"), legend.title="")
```



```
summary(KM_curve, times= c(0, 12, 24))
```

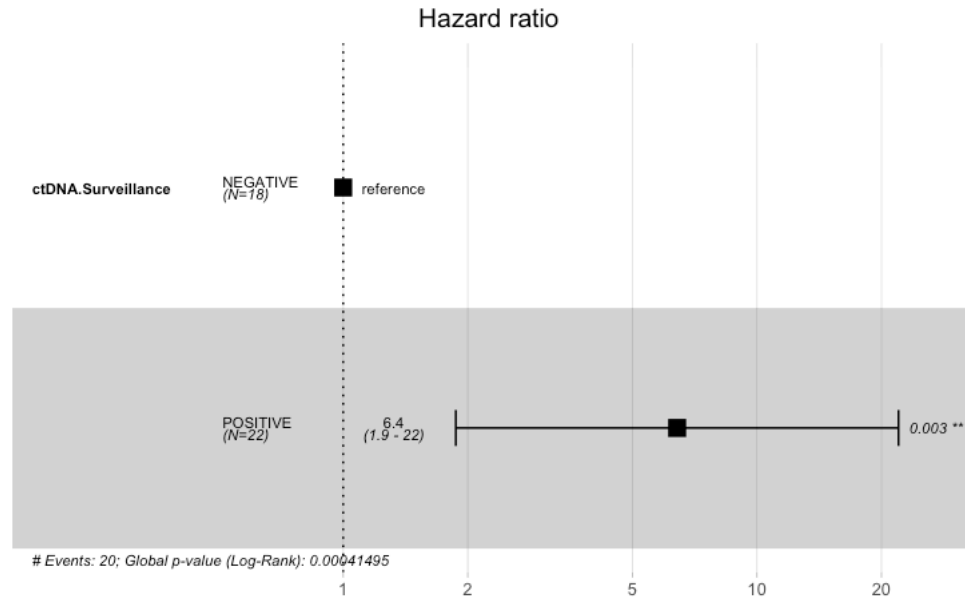
```
Call: survfit(formula = surv_object ~ ctDNA.Surveillance, data = circ_data,
  conf.int = 0.95, conf.type = "log-log")
```

ctDNA.Surveillance=NEGATIVE						
time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
0	18	0	1.000	0.0000	1.000	1.000
12	15	1	0.938	0.0605	0.632	0.991
24	5	2	0.757	0.1277	0.401	0.919

ctDNA.Surveillance=POSITIVE						
time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
0	22	0	1.000	0.000	1.000	1.000
12	12	9	0.591	0.105	0.361	0.762
24	4	6	0.287	0.101	0.114	0.488

```
circ_data$ctDNA.Surveillance <- factor(circ_data$ctDNA.Surveillance, levels=c("NEGATIVE","POSITIVE"))
cox_fit <- coxph(surv_object ~ ctDNA.Surveillance, data=circ_data)
ggforest(cox_fit,data = circ_data)
```



```
summary(cox_fit)
```

```
Call:
coxph(formula = surv_object ~ ctDNA.Surveillance, data = circ_data)

n= 40, number of events= 20

              coef exp(coef) se(coef)      z Pr(>|z|)
ctDNA.SurveillancePOSITIVE 1.8586    6.4151  0.6287  2.956  0.00311 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

              exp(coef) exp(-coef) lower .95 upper .95
ctDNA.SurveillancePOSITIVE    6.415    0.1559    1.871    21.99

Concordance= 0.699 (se = 0.047 )
Likelihood ratio test= 12.46 on 1 df,  p=4e-04
Wald test              = 8.74 on 1 df,  p=0.003
Score (logrank) test = 11.47 on 1 df,  p=7e-04
```

```
cox_fit_summary <- summary(cox_fit)

#Extract values for HR, 95% CI, and p-value
HR <- cox_fit_summary$coefficients[2]
lower_CI <- cox_fit_summary$conf.int[3]
upper_CI <- cox_fit_summary$conf.int[4]
p_value <- cox_fit_summary$coefficients[5]
label_text <- paste0("HR = ", round(HR, 2), " (", round(lower_CI, 2), "-", round(upper_CI, 2), "); p = ", round(p_value, 3))
print(label_text)
```

```
[1] "HR = 6.42 (1.87-21.99); p = 0.003"
```

Hide

```
circ_data$ctDNA.Surveillance <- factor(circ_data$ctDNA.Surveillance, levels = c("NEGATIVE", "POSITIVE"), labels = c("Negative", "Positive"))
circ_data$RFS.Event <- factor(circ_data$RFS.Event, levels = c("FALSE", "TRUE"), labels = c("No Recurrence", "Recurrence"))
contingency_table <- table(circ_data$ctDNA.Surveillance, circ_data$RFS.Event)
chi_square_test <- chisq.test(contingency_table)
print(chi_square_test)
```

Pearson's Chi-squared test with Yates' continuity correction

```
data: contingency_table
X-squared = 12.222, df = 1, p-value = 0.0004722
```

Hide

```
fisher_exact_test <- fisher.test(contingency_table)
print(fisher_exact_test)
```

Fisher's Exact Test for Count Data

```
data: contingency_table
p-value = 0.0003284
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 2.867735 118.989345
sample estimates:
odds ratio
15.45796
```

Hide

```
print(contingency_table)
```

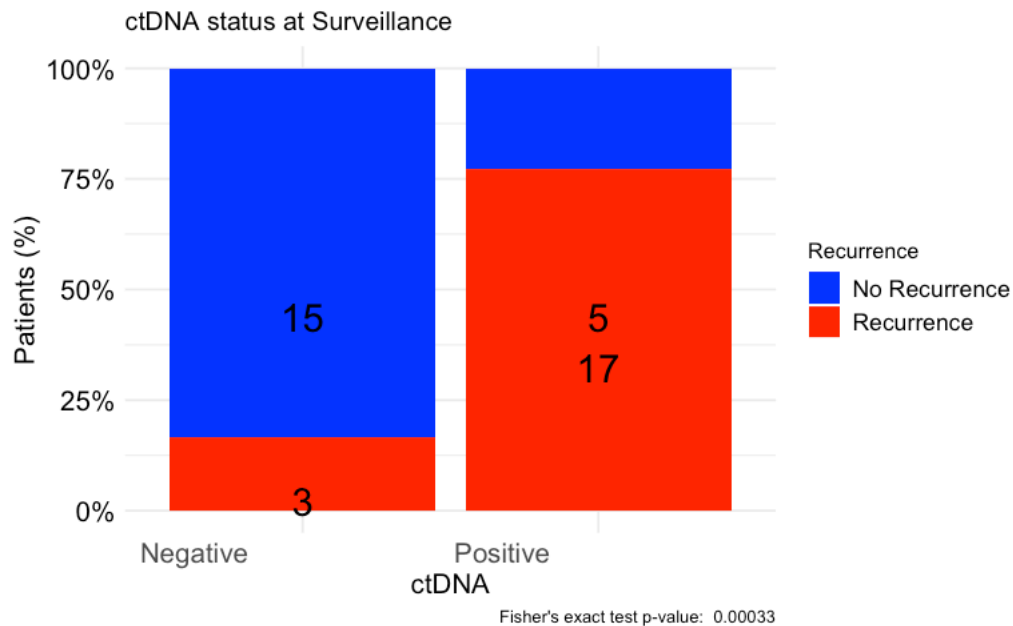
	No Recurrence	Recurrence
Negative	15	3
Positive	5	17

Hide

```

table_df <- as.data.frame(contingency_table)
table_df$Total <- ave(table_df$Freq, table_df$Var1, FUN = sum)
table_df$Percentage <- table_df$Freq / table_df$Total
table_df$MiddlePercentage <- table_df$Percentage / 2
ggplot(table_df, aes(x = Var1, y = Percentage, fill = Var2)) +
  geom_bar(stat = "identity") +
  geom_text(aes(y = MiddlePercentage, label = Freq), position = "stack", color = "black", vjust = 1.5, size = 7)
+
  theme_minimal() +
  labs(title = "ctDNA status at Surveillance",
       x = "ctDNA",
       y = "Patients (%)",
       fill = "Recurrence",
       caption = paste("Fisher's exact test p-value: ", format.pval(fisher_exact_test$p.value))) +
  scale_y_continuous(labels = scales::percent_format()) +
  scale_fill_manual(values = c("No Recurrence" = "blue", "Recurrence" = "red")) + # define custom colors
  theme(axis.text.x = element_text(angle = 0, hjust = 1.5, size = 14), # increase x-axis text size
        axis.text.y = element_text(size = 14, color = "black"), # increase y-axis text size
        axis.title.x = element_text(size = 14, color = "black"), # increase x-axis label size
        axis.title.y = element_text(size = 14, color = "black"), # increase y-axis label size
        legend.text = element_text(size = 12, color = "black")) # increase Recurrence label size

```



#Time-dependent analysis for RFS in longitudinal time points

Hide

```

rm(list=ls())
setwd("~/Downloads")
dt <- read_xlsx("CLIA UTUC Clinical Data_Time dependent.xlsx") |>
  clean_names() |>
  mutate(across(
    .cols = c(window_start_date, dfs_date, surveillance_1_date:surveillance_12_date),
    .fns = ~ as_date(as.Date(x, format = "%Y-%m-%d"))
  )) |>
  filter(final_cohort == TRUE)

dt_biomarker <- dt |>
  select(pts_id, ct_dna_surveillance_available,
         window_start_date,
         surveillance_1_status:surveillance_12_date) |>
  filter(ct_dna_surveillance_available) |>
  pivot_longer(cols = surveillance_1_status:surveillance_12_date,
               names_to = c("visit_number", ".value"),
               names_pattern = "surveillance_(.)(.*)") |>
  mutate(biomarker_time = day(days(date - window_start_date))) |>
  select(pts_id, biomarker_time, biomarker_status = status) |>
  filter(!is.na(biomarker_time))

glimpse(dt_biomarker)

```



Hide

Hide

Hide

```
dt_survival <- dt_survival |>
  filter(dfs_time > 0)

aux <- dt |>
  select(pts_id, ct_dna_surveillance_available,
         window_start_date, dfs_date,
         surveillance_1_date:surveillance_12_date) |>
  mutate(across(.cols = surveillance_1_date:surveillance_12_date,
                .fns = ~ .x - window_start_date)) |>
  mutate(across(.cols = surveillance_1_date:surveillance_12_date,
                .fns = ~ .x < 0)) |>
  rowwise() |>
  mutate(sum_neg =
         sum(c_across(surveillance_1_date:surveillance_12_date),
              na.rm = TRUE)) |>
  select(pts_id, sum_neg)

tab <- left_join(aux, dt) |>
  filter(sum_neg > 0) |>
  select(pts_id, sum_neg, window_start_date,
         surveillance_1_date:surveillance_12_date) |>
  mutate(across(.cols = window_start_date:surveillance_12_date,
                .fns = ~ as_date(.x)))

datatable(tab, filter = "top")
```

Show 10 entries

Search: 

pts_id	sum_neg	window_start_date	surveillance_1_date	surveillance_2_date	surveillance_3_date	surveillance_4_date
<input type="text"/>	All	All	All	All	All	All

Showing 0 to 0 of 0 entries

[Previous](#)[Next](#)[Hide](#)

```
aux <- dt |>
  select(pts_id, ct_dna_surveillance_available,
         window_start_date, dfs_date,
         surveillance_1_date:surveillance_12_date) |>
  mutate(across(.cols = dfs_date:surveillance_12_date,
                .fns = ~ .x - window_start_date)) |>
  mutate(across(.cols = surveillance_2_date:surveillance_12_date,
                .fns = ~ dfs_date < .x)) |>
  rowwise() |>
  mutate(n_biomarker_after_event = sum(c_across(surveillance_2_date:
                                                surveillance_12_date),
                                       na.rm = TRUE)) |>
  mutate(across(.cols = surveillance_1_date:surveillance_12_date,
                .fns = ~ !is.na(.x))) |>
  mutate(total_biomarker = sum(c_across(surveillance_2_date:
                                       surveillance_12_date),
                               na.rm = TRUE)) |>
  select(pts_id, n_biomarker_after_event, total_biomarker)

temp <- aux |>
  select(-pts_id) |>
  group_by(n_biomarker_after_event, total_biomarker) |> # Direct grouping
  summarise(freq = n(), .groups = "drop") # Drop groups after summarization

tab <- left_join(aux, dt) |>
  select(pts_id, n_biomarker_after_event, total_biomarker,
         dfs_date,
         surveillance_2_date:surveillance_12_date) |>
  mutate(across(.cols = dfs_date:surveillance_12_date,
                .fns = ~ as_date(.x))) |>
  filter(n_biomarker_after_event > 0)
datatable(tab, filter = "top")
```

Show 10 entries

Search:

	pts_id	n_biomarker_after_event	total_biomarker	dfs_date	surveillance_2_date	surveillance_3_date	surveillance_4_date
	<input type="text" value=""/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>
1	UTUC-058	1	6	2025-03-24	2023-07-21	2023-10-13	2024-06-17
2	UTUC-063	1	5	2025-02-25	2024-03-04	2024-06-07	2024-09-04
3	UTUC-072	1	6	2025-03-24	2023-08-25	2023-12-07	2024-03-01
4	UTUC-074	1	1	2025-03-19	2025-06-13		

Showing 1 to 4 of 4 entries

Previous1Next

Hide

```
aux <- tmerge(data1 = dt_survival,
              data2 = dt_survival,
              id = pts_id,
              dfs_event = event(dfs_time, dfs_event))
dt_final <- tmerge(data1 = aux,
                  data2 = dt_biomarker,
                  id = pts_id,
                  biomarker_status =
                    tdc(biomarker_time, biomarker_status))

datatable(dt_final, filter = "top")
```

Show10entries

Search:

	pts_id	dfs_time	dfs_event	tstart	tstop	biomarker_status
	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>
1	UTUC-002	143	0	0	133	
2	UTUC-002	143	1	133	143	POSITIVE
3	UTUC-010	707	0	0	144	
4	UTUC-010	707	0	144	181	NEGATIVE
5	UTUC-010	707	0	181	223	NEGATIVE
6	UTUC-010	707	0	223	267	NEGATIVE
7	UTUC-010	707	0	267	307	NEGATIVE
8	UTUC-010	707	0	307	349	NEGATIVE
9	UTUC-010	707	0	349	392	NEGATIVE
10	UTUC-010	707	0	392	433	POSITIVE

Showing 1 to 10 of 163 entries

Previous12345...17Next

Hide

```
# Syntax if there is not time-dependent covariate
# fit <- coxph(Surv(dfs_time, dfs_event) ~ biomarker_status,
#             data = dt_final)
# summary(fit)

fit <- coxph(Surv(tstart, tstop, dfs_event) ~ biomarker_status,
            data = dt_final)
summary(fit)
```

```
Call:
coxph(formula = Surv(tstart, tstop, dfs_event) ~ biomarker_status,
      data = dt_final)

n= 127, number of events= 17
(36 observations deleted due to missingness)

              coef exp(coef) se(coef)      z Pr(>|z|)
biomarker_statusPOSITIVE  2.8209   16.7919   0.6524  4.324 1.53e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

              exp(coef) exp(-coef) lower .95 upper .95
biomarker_statusPOSITIVE    16.79    0.05955     4.675    60.32

Concordance= 0.824 (se = 0.055 )
Likelihood ratio test= 26.23 on 1 df,  p=3e-07
Wald test               = 18.7 on 1 df,  p=2e-05
Score (logrank) test = 31 on 1 df,  p=3e-08
```

Hide

```
cox_fit_summary <- summary(fit)

#Extract values for HR, 95% CI, and p-value
HR <- cox_fit_summary$coefficients[2]
lower_CI <- cox_fit_summary$conf.int[3]
upper_CI <- cox_fit_summary$conf.int[4]
p_value <- cox_fit_summary$coefficients[5]
label_text <- paste0("HR = ", round(HR, 2), " (", round(lower_CI, 2), "-", round(upper_CI, 2), "); p = ", round(p_value, 3))
print(label_text)
```

```
[1] "HR = 16.79 (4.67-60.32); p = 0"
```

#Median numbers of time points in the longitudinal setting

Hide

```
# Load the dataset
rm(list=ls())
setwd("~/Downloads")
circ_data <- read.csv("RWE UTUC_Clinical Data 102025.csv")
circ_data <- circ_data[circ_data$Final.cohort=="TRUE",]
circ_data <- circ_data[circ_data$ctDNA.Surveillance!="",]
circ_datadf <- as.data.frame(circ_data)

median_Nsurvtps <- median(circ_datadf$Nsurvtps, na.rm = TRUE)
min_Nsurvtps <- min(circ_datadf$Nsurvtps, na.rm = TRUE)
max_Nsurvtps <- max(circ_datadf$Nsurvtps, na.rm = TRUE)

cat(sprintf("Median # of surveillance time points: %d (%d-%d)\n",
            median_Nsurvtps, min_Nsurvtps, max_Nsurvtps))
```

```
Median # of surveillance time points: 3 (1-12)
```

#Median of median interval of ctDNA timepoints and radiological imaging assessment

Hide

```
rm(list=ls())
setwd("~/Downloads")
df <- read.csv("RWE UTUC_OP 102025.csv", stringsAsFactors = FALSE)

names(df) <- trimws(names(df))

# ---- Helper: compute median interval per patient ----
median_interval_per_patient <- function(data, filter_col, filter_val) {
  data %>%
    filter(!sym(filter_col) == filter_val) %>%
    arrange(PatientName, date.diff) %>%
    group_by(PatientName) %>%
    mutate(interval = date.diff - lag(date.diff)) %>%
    summarise(median_interval = median(interval, na.rm = TRUE), .groups = "drop") %>%
    mutate(event_group = filter_val)
}

# ---- Compute per-patient medians ----
ctDNA_patient_medians <- median_interval_per_patient(df, "Event", "ctDNA")
imaging_patient_medians <- median_interval_per_patient(df, "Event_type", "Imaging")

# ---- Function to summarize patient-level medians to cohort-level ----
cohort_summary <- function(patient_data, event_label) {
  # Filter out patients with NA median intervals
  valid_data <- patient_data %>% filter(!is.na(median_interval))

  data.frame(
    Event_Type = event_label,
    Cohort_Median_Frequency = median(valid_data$median_interval, na.rm = TRUE),
    Min_Patient_Median = min(valid_data$median_interval, na.rm = TRUE),
    Max_Patient_Median = max(valid_data$median_interval, na.rm = TRUE),
    Range_Patient_Median = max(valid_data$median_interval, na.rm = TRUE) -
      min(valid_data$median_interval, na.rm = TRUE)
  )
}

# ---- Combine all results ----
results <- bind_rows(
  cohort_summary(ctDNA_patient_medians, "ctDNA"),
  cohort_summary(imaging_patient_medians, "Imaging")
)

# ---- Print cohort-level summary ----
cat("==== Median Frequency (Days) per Cohort =====\n")
```

```
==== Median Frequency (Days) per Cohort =====
```

[Hide](#)

```
print(results, row.names = FALSE)
```

Event_Type <chr>	Cohort_Median_Frequency <dbl>	Min_Patient_Median <dbl>	Max_Patient_Median <dbl>	Range_Patient_Median <dbl>
ctDNA	75.75	14	156	142
Imaging	154.25	0	932	932
2 rows				

[Hide](#)

```
# ---- OPTIONAL: Save patient-level medians ----
patient_level_medians <- bind_rows(ctDNA_patient_medians, imaging_patient_medians)
#write.csv(patient_level_medians, "patient_median_intervals.csv", row.names = FALSE)
```

```
#Plot for individual lead-time calculations for each pt
```

[Hide](#)

```

rm(list=ls())
csv_path <- "~/Downloads/CLIA UTUC_Lead Time pts.csv"
df <- read.csv(csv_path, check.names = FALSE)
if ("Final.cohort" %in% names(df)) {
  df <- df[df$Final.cohort == TRUE, ]
} else {
  warning("Column 'Final.cohort' not found in the dataset.")
}
id_candidates <- c("Patient","ID","Pt","Subject","Sample")
id_col <- id_candidates[id_candidates %in% names(df)][1]
if (is.na(id_col)) {
  df <- df %>% mutate(Patient = row_number())
  id_col <- "Patient"
}

num_cols <- intersect(c("MR","CR","LT"), names(df))
df[num_cols] <- lapply(df[num_cols], function(x) suppressWarnings(as.numeric(x)))

# If LT missing, compute in days
if (!("LT" %in% names(df))) {
  df <- df %>% mutate(LT = CR - MR)
}

# ---- convert to months ----
# Approximate: 30.44 days per month (average)
days_to_months <- function(x) x / 30.437

df <- df %>%
  mutate(MR_mo = days_to_months(MR),
         CR_mo = days_to_months(CR),
         LT_mo = days_to_months(LT))

# ---- ordering for y-axis ----
df <- df %>%
  mutate(Earliest = pmin(MR_mo, CR_mo, na.rm = TRUE)) %>%
  arrange(Earliest) %>%
  mutate(Patient_f = factor(.data[[id_col]], levels = .data[[id_col]]))

# ---- long format for points ----
points_long <- df %>%
  select(Patient_f, MR_mo, CR_mo) %>%
  pivot_longer(c(MR_mo, CR_mo), names_to = "Type", values_to = "Months") %>%
  mutate(Type = dplyr::recode(Type,
                             "MR_mo" = "Molecular recurrence",
                             "CR_mo" = "Clinical recurrence"))

# ---- segments between MR and CR ----
segments_df <- df %>%
  transmute(Patient_f,
            x0 = MR_mo, x1 = CR_mo,
            lt_flag = if_else(LT > 120, "gt120", "le120"))

# ---- annotation ----
med_lt <- median(df$LT_mo, na.rm = TRUE)
min_lt <- min(df$LT_mo, na.rm = TRUE)
max_lt <- max(df$LT_mo, na.rm = TRUE)

annot_label <- sprintf("Median lead-time:\n%.1f months (%.1f to %.1f)",
                      med_lt, min_lt, max_lt)

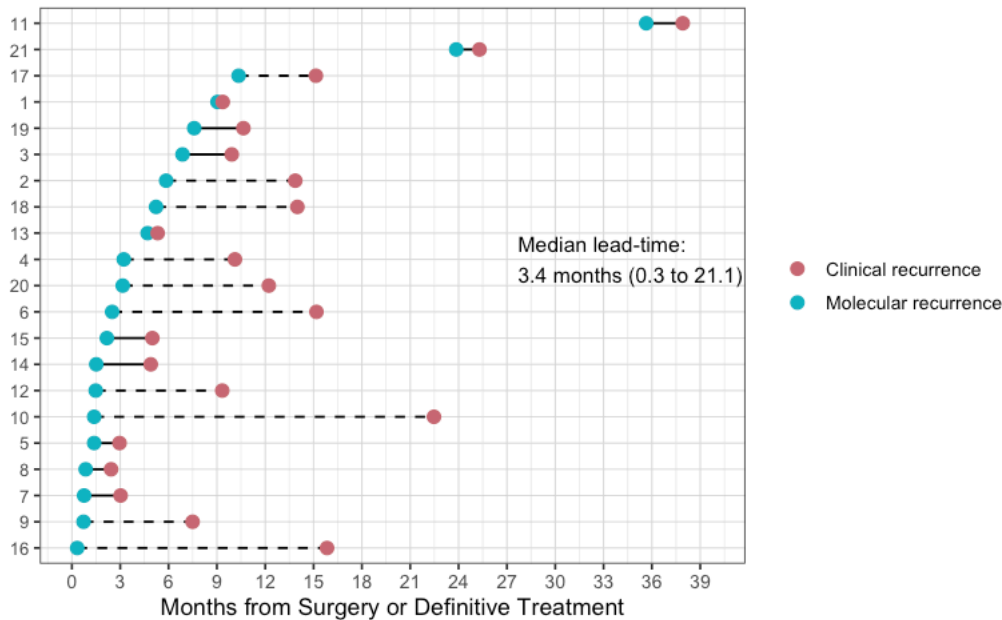
# ---- plot ----
pal <- c("Molecular recurrence" = "#10B4C1",
        "Clinical recurrence" = "#C96A72")

x_max <- max(c(df$MR_mo, df$CR_mo), na.rm = TRUE)
y_mid <- levels(df$Patient_f)[ceiling(nlevels(df$Patient_f) * 0.55)]

p <- ggplot() +
  geom_segment(data = segments_df,
             aes(x = x0, xend = x1, y = Patient_f, yend = Patient_f,
                 linetype = lt_flag),
             linewidth = 0.6, color = "black") +
  scale_linetype_manual(values = c(le120 = "solid", gt120 = "dashed"),
                       guide = "none") +
  geom_point(data = points_long,
            aes(x = Months, y = Patient_f, color = Type),
            size = 2.8) +
  scale_color_manual(values = pal, name = NULL) +

```

```
labs(x = "Months from Surgery or Definitive Treatment", y = NULL) +
annotate("text",
  x = x_max * 0.73,
  y = y_mid,
  label = annot_label,
  hjust = 0, vjust = 0.5, size = 3.8) +
scale_x_continuous(breaks = seq(0, ceiling(x_max/3)*3, by = 3)) +
coord_cartesian(xlim = c(0, x_max * 1.05)) +
theme_bw(base_size = 12) +
theme(
  panel.grid.major = element_line(color = "grey85", linewidth = 0.3),
  panel.grid.minor = element_line(color = "grey92", linewidth = 0.2),
  axis.text.y = element_text(size = 9)
)
print(p)
```



Hide

```
wc_df <- df %>% dplyr::select(MR, CR) %>% tidyr::drop_na()
paired_diff <- wc_df$CR - wc_df$MR

wilx <- wilcox.test(
  x = wc_df$CR, y = wc_df$MR,
  paired = TRUE,
  alternative = "two.sided",
  conf.int = TRUE,      # gives CI for the Hodges-Lehmann estimate of the median difference
  exact = FALSE        # safer for ties/Large N
)

W_stat <- unname(wilx$statistic)      # Wilcoxon V
p_value <- wilx$p.value
HL_est <- unname(wilx$estimate)       # median of (CR - MR)
HL_ci <- wilx$conf.int                # CI for median difference

# Simple p-value formatter for annotation/publication
fmt_p <- function(p) {
  if (is.na(p)) return("NA")
  if (p < 0.001) "< 0.001" else sprintf("= %.3f", round(p, 3))
}
p_text <- paste0("P ", fmt_p(p_value)) # e.g., "P = 0.003" or "P < 0.001"

# Optional: print a compact summary
cat("\nPaired Wilcoxon signed-rank test (CR vs MR)\n",
  "-----\n",
  sprintf("N pairs: %d", nrow(wc_df)), "\n",
  sprintf("W (V): %g", W_stat), "\n",
  sprintf("P-value: %s", ifelse(p_value < 0.001, "< 0.001", sprintf("%.6f", p_value))), "\n",
  sprintf("Hodges-Lehmann median difference (CR - MR): %.1f days", HL_est), "\n",
  sprintf("95% CI: [%.1f, %.1f] days", HL_ci[1], HL_ci[2]), "\n", sep = "")
```

Paired Wilcoxon signed-rank test (CR vs MR)

---

N pairs: 21  
W (V): 231  
P-value: < 0.001  
Hodges-Lehmann median difference (CR - MR): 155.5 days  
95% CI: [86.0, 241.5] days

#Multivariate cox regression at Surveillance Window for RFS

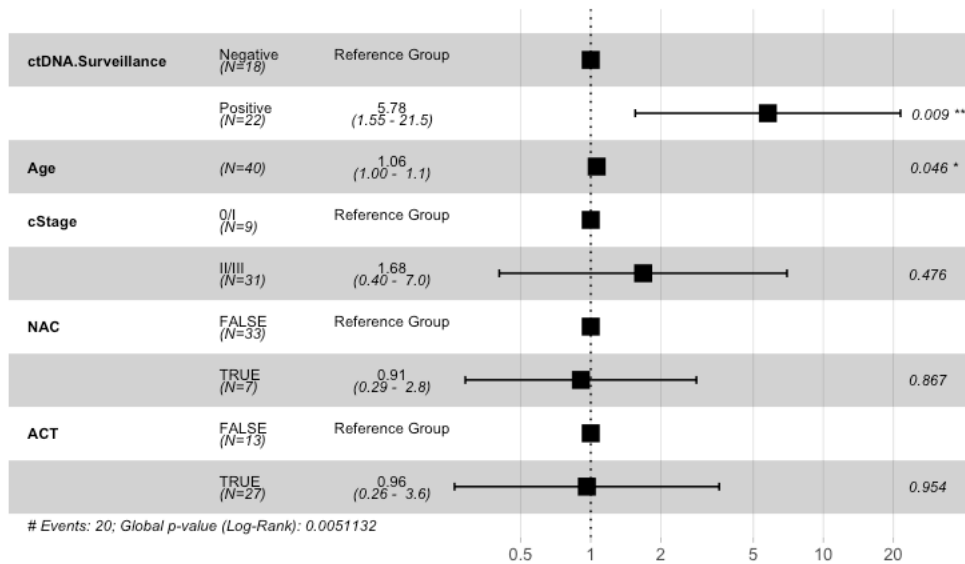
Hide

```
rm(list=ls())
setwd("~/Downloads")
circ_data <- read.csv("RWE UTUC_Clinical Data 102025.csv")
circ_data <- circ_data[circ_data$Final.cohort=="TRUE",]
circ_data <- circ_data[circ_data$ctDNA.Surveillance!="",]

circ_data$ctDNA.Surveillance <- factor(circ_data$ctDNA.Surveillance, levels=c("NEGATIVE","POSITIVE"), labels = c("Negative", "Positive"))
circ_data$cStage <- factor(circ_data$cStage, levels = c("0/I", "II/III"))
circ_data$NAC <- factor(circ_data$NAC, levels = c("FALSE", "TRUE"))
circ_data$ACT <- factor(circ_data$ACT, levels = c("FALSE", "TRUE"))
surv_object <- Surv(time = circ_data$RFS.months, event = circ_data$RFS.Event)
cox_fit <- coxph(surv_object ~ ctDNA.Surveillance + Age + cStage + NAC + ACT, data=circ_data)
ggforest(cox_fit, data = circ_data, main = "Multivariate Regression Model for RFS - All Stages", refLabel = "Reference Group")
```



## Multivariate Regression Model for RFS - All Stages



Hide

```
test.ph <- cox.zph(cox_fit)
```

#ctDNA and MTM/mL Dynamics for pts at surveillance window

Hide

```
#Dynamics and MTM/mL plots for patients with ctDNA negative at surveillance
rm(list=ls())
setwd("~/Downloads")
df <- read.csv("CLIA UTUC ctDNA MTM.csv", stringsAsFactors = FALSE)
df <- df[df$Final.cohort=="TRUE",]
df <- df[df$ctDNA.Surveillance=="NEGATIVE",]

df$RFS.Event <- ifelse(df$RFS.Event %in% c("No", "no", "FALSE", "False", "0"), FALSE,
  ifelse(df$RFS.Event %in% c("Yes", "yes", "TRUE", "True", "1"), TRUE, NA))
df$RFS.Event <- factor(df$RFS.Event, levels = c(FALSE, TRUE))
df <- df %>%
  group_by(PatientName) %>%
  filter(n() >= 2) %>% #keep only pts with at least 2 post-surgery time points
  ungroup()

num_unique <- length(unique(df$PatientName))
cat("Number of unique patients:", num_unique, "\n")
```

Number of unique patients: 17

Hide

```
df_patient_pfs <- df %>%
  group_by(PatientName) %>%
  dplyr::summarize(
    PFS_True = any(RFS.Event == TRUE, na.rm = TRUE),
    PFS_False = all(RFS.Event == FALSE, na.rm = TRUE)
  )

num_true <- sum(df_patient_pfs$PFS_True)
num_false <- sum(df_patient_pfs$PFS_False)

cat("Number of unique patients with Event:", num_true, "\n")
```

Number of unique patients with Event: 3

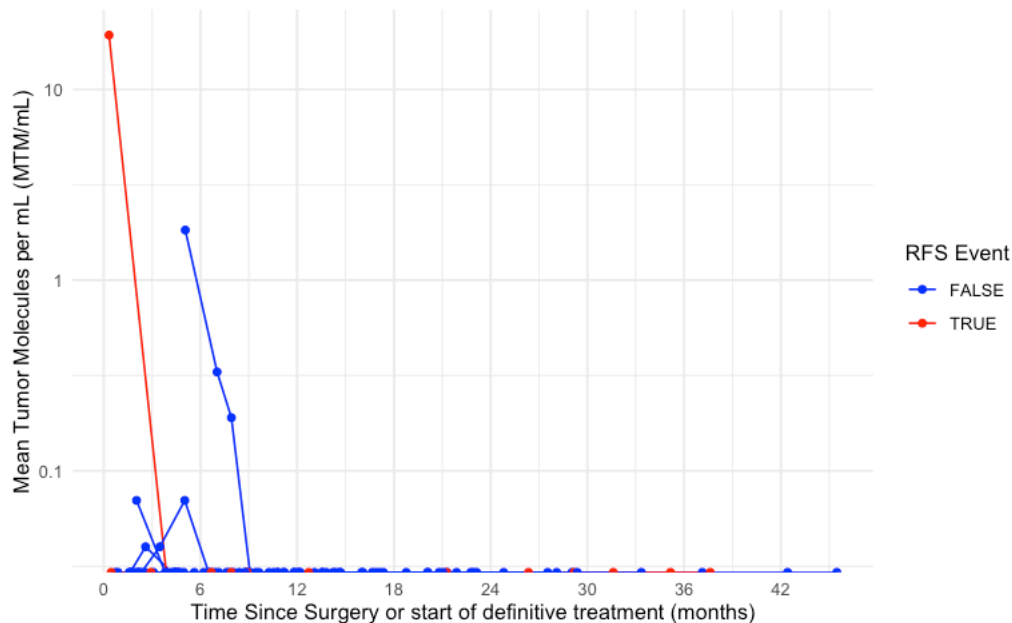
Hide

```
cat("Number of unique patients with No Event:", num_false, "\n")
```

Number of unique patients with No Event: 14

Hide

```
p <- ggplot(df, aes(x = date.diff.months,
                    y = MTM.mL,
                    group = PatientName,
                    color = RFS.Event)) +
  geom_line() +      # Connect timepoints for each patient
  geom_point() +     # Add points for each timepoint
  # Use a log10 scale for the y-axis with specified breaks
  scale_y_log10(breaks = c(0.01, 0.1, 1, 10, 100, 1000),
               labels = c("0.01", "0.1", "1", "10", "100", "1000")) +
  scale_x_continuous(breaks = seq(0, max(df$date.diff.months, na.rm = TRUE), by = 6)) +
  scale_color_manual(values = c("FALSE" = "blue", "TRUE" = "red")) +
  labs(
    x = "Time Since Surgery or start of definitive treatment (months)",
    y = "Mean Tumor Molecules per mL (MTM/mL)",
    color = "RFS Event"
  ) +
  theme_minimal()
print(p)
```



Hide

```
#Dynamics and MTM/mL plots for patients with ctDNA positive at surveillance
rm(list=ls())
setwd("~/Downloads")
df <- read.csv("CLIA UTUC ctDNA MTM.csv", stringsAsFactors = FALSE)
df <- df[df$Final.cohort=="TRUE",]
df <- df[df$ctDNA.Surveillance=="POSITIVE",]

df$RFS.Event <- ifelse(df$RFS.Event %in% c("No", "no", "FALSE", "False", "0"), FALSE,
                      ifelse(df$RFS.Event %in% c("Yes", "yes", "TRUE", "True", "1"), TRUE, NA))
df$RFS.Event <- factor(df$RFS.Event, levels = c(FALSE, TRUE))
df <- df %>%
  group_by(PatientName) %>%
  filter(n() >= 2) %>% #keep only pts with at least 2 post-surgery time points
  ungroup()

num_unique <- length(unique(df$PatientName))
cat("Number of unique patients:", num_unique, "\n")
```

Number of unique patients: 21

Hide

```
df_patient_pfs <- df %>%
  group_by(PatientName) %>%
  dplyr::summarize(
    PFS_True = any(RFS.Event == TRUE, na.rm = TRUE),
    PFS_False = all(RFS.Event == FALSE, na.rm = TRUE)
  )

num_true <- sum(df_patient_pfs$PFS_True)
num_false <- sum(df_patient_pfs$PFS_False)

cat("Number of unique patients with Event:", num_true, "\n")
```

Number of unique patients with Event: 16

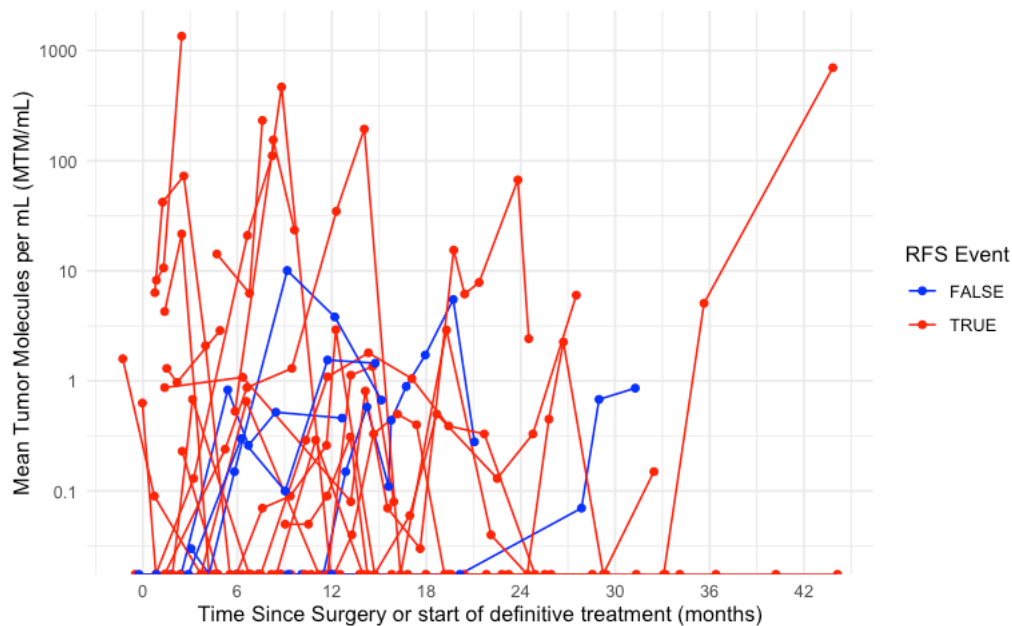
Hide

```
cat("Number of unique patients with No Event:", num_false, "\n")
```

Number of unique patients with No Event: 5

Hide

```
p <- ggplot(df, aes(x = date.diff.months,
  y = MTM.mL,
  group = PatientName,
  color = RFS.Event)) +
  geom_line() + # Connect timepoints for each patient
  geom_point() + # Add points for each timepoint
  # Use a log10 scale for the y-axis with specified breaks
  scale_y_log10(breaks = c(0.01, 0.1, 1, 10, 100, 1000),
    labels = c("0.01", "0.1", "1", "10", "100", "1000")) +
  scale_x_continuous(breaks = seq(0, max(df$date.diff.months, na.rm = TRUE), by = 6)) +
  scale_color_manual(values = c("FALSE" = "blue", "TRUE" = "red")) +
  labs(
    x = "Time Since Surgery or start of definitive treatment (months)",
    y = "Mean Tumor Molecules per mL (MTM/mL)",
    color = "RFS Event"
  ) +
  theme_minimal()
print(p)
```



#ctDNA velocity and lead time liner regression

Hide

```

rm(list=ls())
csv_path <- "~/Downloads/CLIA UTUC_ctDNA velocity.csv" # <- adjust if needed
df_raw <- read.csv(csv_path, check.names = FALSE)
df <- df_raw %>%
  rename(MTM_mL = `MTM.mL`) %>%
  mutate(
    daysCR.months = as.numeric(daysCR.months),
    MTM_mL = as.numeric(MTM_mL)
  ) %>%
  filter(Final.cohort == TRUE, !is.na(PatientName), !is.na(daysCR.months), !is.na(MTM_mL))

preCR <- df %>%
  filter(daysCR.months <= 0, is.finite(MTM_mL), MTM_mL > 0)

eligible <- preCR %>%
  group_by(PatientName) %>%
  filter(n() >= 2, n_distinct(daysCR.months) >= 2) %>%
  ungroup()
if (nrow(eligible) == 0) {
  warning("No patients have ≥2 valid pre-recurrence points with distinct times; regression lines will be omitted.")
}

fits <- eligible %>%
  group_by(PatientName) %>%
  summarise(x_min = min(daysCR.months, na.rm = TRUE), .groups = "drop") %>%
  mutate(grid = map(x_min, ~seq(.x, 0, length.out = 50))) %>%
  select(PatientName, grid)

predict_patient <- function(dat, newx) {
  if (length(newx) == 0) {
    return(tibble(daysCR.months = numeric(0), MTM_mL = numeric(0)))
  }
  dat2 <- dat %>%
    filter(is.finite(MTM_mL), MTM_mL > 0) %>%
    mutate(log_ctdna = log10(MTM_mL))
  if (nrow(dat2) < 2 || n_distinct(dat2$daysCR.months) < 2 || any(!is.finite(dat2$log_ctdna))) {
    return(tibble(daysCR.months = numeric(0), MTM_mL = numeric(0)))
  }
  m <- lm(log_ctdna ~ daysCR.months, data = dat2)
  tibble(
    daysCR.months = newx,
    MTM_mL = 10 ^ predict(m, newdata = tibble(daysCR.months = newx))
  )
}

pred_lines <- eligible %>%
  group_by(PatientName) %>%
  tidyr::nest() %>% # list-column "data" per patient
  left_join(fits, by = "PatientName") %>% # list-column "grid" per patient
  mutate(pred = map2(data, grid, ~predict_patient(.x, .y))) %>%
  select(PatientName, pred) %>%
  tidyr::unnest(pred)

pooled_line <- {
  if (nrow(preCR) >= 2 && n_distinct(preCR$daysCR.months) >= 2) {
    pooled_x <- seq(min(preCR$daysCR.months, na.rm = TRUE), 0, length.out = 100)
    pooled_fit <- lm(log10(MTM_mL) ~ daysCR.months, data = preCR)
    tibble(
      daysCR.months = pooled_x,
      MTM_mL = 10 ^ predict(pooled_fit, newdata = tibble(daysCR.months = pooled_x))
    )
  } else {
    tibble(daysCR.months = numeric(0), MTM_mL = numeric(0))
  }
}

x_min <- floor(min(df$daysCR.months, na.rm = TRUE) / 3) * 3
x_max <- ceiling(max(df$daysCR.months, na.rm = TRUE) / 3) * 3
y_min_pos <- max(min(df$MTM_mL[df$MTM_mL > 0], na.rm = TRUE) / 2, 0.01)
y_max_pos <- 10 ^ ceiling(log10(max(df$MTM_mL, na.rm = TRUE)))
log_breaks <- 10 ^ seq(floor(log10(y_min_pos)), ceiling(log10(y_max_pos)))

p <- ggplot() +
  # per-patient fitted lines (if any)
  geom_line(data = pred_lines,
    aes(x = daysCR.months, y = MTM_mL, color = PatientName),
    linewidth = 1) +

```

```

# pooled dashed trend (if any)
geom_line(data = pooled_line,
          aes(x = daysCR.months, y = MTM_mL),
          linewidth = 0.8, linetype = "dashed", color = "grey35") +
# raw points (all timepoints, before and after)
geom_point(data = df,
           aes(x = daysCR.months, y = MTM_mL, color = PatientName),
           size = 1.8, alpha = 0.85) +
# vertical line at imaging-positive date
geom_vline(xintercept = 0, linetype = "dashed", linewidth = 0.8, color = "black") +
scale_y_log10(breaks = log_breaks,
              labels = label_number(accuracy = 1)) +
scale_x_continuous(breaks = seq(x_min, x_max, by = 3)) + # every 3 months
labs(
  x = "Months before/after clinical recurrence (0 = imaging positive)",
  y = "ctDNA level (MTM/mL)",
  color = "Patient"
) +
theme_bw(base_size = 12) +
theme(
  panel.grid.major = element_line(color = "grey85", linewidth = 0.3),
  panel.grid.minor = element_line(color = "grey92", linewidth = 0.2),
  legend.position = "none" # change to "right" if you want the legend
)

print(p)

```

