Online tools websitie instructions：

1、 http://gliovis.bioinfo.cnio.es/ （Figure1A , Figure 2A-G complete by this website）
CGGA dataset informations： https://pubmed.ncbi.nlm.nih.gov/28291232/； the data can be download from this wbsite （http://gliovis.bioinfo.cnio.es/）
TCGA（LGG-GBM）dataset informations：https://pubmed.ncbi.nlm.nih.gov/26824661/; the data can be download from this wbsite （http://gliovis.bioinfo.cnio.es/）
Rembrandt dataset informations： https://pubmed.ncbi.nlm.nih.gov/19208739/； the data can be download from this wbsite （http://gliovis.bioinfo.cnio.es/）
Gravendeel dataset informations： https://pubmed.ncbi.nlm.nih.gov/19920198/； the data can be download from this wbsite （http://gliovis.bioinfo.cnio.es/）

2、 http://timer.cistrome.org/ (Figure 6 and Figure 7A complete by this website)
3、 https://www.proteinatlas.org/ （Figure 1D and Figure 7B complete by this website）

R code for data cleaning analysis：

NO1、###### Code for gene ID Conversion tools

```
#if (!requireNamespace("BiocManager", quietly = TRUE))
#       install.packages("BiocManager")
#BiocManager::install("org.Hs.eg.db")
library("org.Hs.eg.db")              # Reference the package
inputFile="intersect.txt"            # Input file
setwd("C:\\Users\\lexb4\\Desktop\\WDR77  analysis\15.symbo2id")          # Set the working directory
rt=read.table(inputFile,sep="\t",check.names=F,header=F)      # Read the input file
genes=as.vector(rt[,1])              # Get a list of genes
entrezIDs <- mget(genes, org.Hs.egSYMBOL2EG, ifnotfound=NA)      # Find out the ID corresponding to the gene
entrezIDs <- as.character(entrezIDs)
# Output the result file of the gene ID
out=cbind(rt,entrezID=entrezIDs)
colnames(out)[1]="Gene"
write.table(out,file="id.txt",sep="\t",quote=F,row.names=F)
```

NO2、###### Code for unicox analysis (for Figure 3 E and F)

```
#install.packages('survival')
library(survival)
setwd("C:\\Users\\Desktop\\CGGA\\11.uniIndep")                                    #
Set the working directory
```

```
rt=read.table("singleGeneSurData.txt",header=T,sep="\t",check.names=F,row.names=1)
# Read the input file
colnames(rt)=gsub("_status","",colnames(rt))
outTab=data.frame()
for(i in colnames(rt[,3:ncol(rt)])){
        cox <- coxph(Surv(futime, fustat) ~ rt[,i], data = rt)
        coxSummary = summary(cox)
        coxP=coxSummary$coefficients[,"Pr(>|z|)"]
        outTab=rbind(outTab,
                        cbind(id=i,
                        HR=coxSummary$conf.int[,"exp(coef)"],
                        HR.95L=coxSummary$conf.int[,"lower .95"],
                        HR.95H=coxSummary$conf.int[,"upper .95"],
                        pvalue=coxSummary$coefficients[,"Pr(>|z|)"])
                        )
}
write.table(outTab,file="uniCox.txt",sep="\t",row.names=F,quote=F)
###### Map the forest ######
# Read the input file
rt <- read.table("uniCox.txt",header=T,sep="\t",row.names=1,check.names=F)
gene <- rownames(rt)
hr <- sprintf("%.3f",rt$"HR")
hrLow  <- sprintf("%.3f",rt$"HR.95L")
hrHigh <- sprintf("%.3f",rt$"HR.95H")
Hazard.ratio <- paste0(hr,"(",hrLow,"-",hrHigh,")")
pVal <- ifelse(rt$pvalue<0.001, "<0.001", sprintf("%.3f", rt$pvalue))
# Output graphics
pdf(file="uniForest.pdf", width = 7,height = 4)
n <- nrow(rt)
nRow <- n+1
ylim <- c(1,nRow)
layout(matrix(c(1,2),nc=2),width=c(3,2.2))
# Plot the clinical information on the left side of the forest plot
xlim = c(0,3)
par(mar=c(4,2,2,1))
plot(1,xlim=xlim,ylim=ylim,type="n",axes=F,xlab="",ylab="")
text.cex=0.75
text(0,n:1,gene,adj=0,cex=text.cex)
text(1.5-0.5*0.2,n:1,pVal,adj=1,cex=text.cex);text(1.5-
0.5*0.2,n+1,'pvalue',cex=text.cex,font=2,adj=1)
text(3,n:1,Hazard.ratio,adj=1,cex=text.cex);text(3,n+1,'Hazard ratio',cex=text.cex,font=2,adj=1,)
# Map the forest
par(mar=c(4,1,2,1),mgp=c(2,0.5,0))
xlim = c(0,max(as.numeric(hrLow),as.numeric(hrHigh)))
```

```
plot(1,xlim=xlim,ylim=ylim,type="n",axes=F,ylab="",xaxs="i",xlab="Hazard ratio")
arrows(as.numeric(hrLow),n:1,as.numeric(hrHigh),n:1,angle=90,code=3,length=0.05,col="darkbl
ue",lwd=2.5)
abline(v=1,col="black",lty=2,lwd=2)
boxcolor = ifelse(as.numeric(hr) > 1, 'green', 'green')
points(as.numeric(hr), n:1, pch = 15, col = boxcolor, cex=1.3)
axis(1)
dev.off()
```

NO3、######R code for GO annotation analysis (for Figure 4 A and C)

```
#install.packages("colorspace")
#install.packages("stringi")
#install.packages("ggplot2")
#if (!requireNamespace("BiocManager", quietly = TRUE))
#       install.packages("BiocManager")
#BiocManager::install("DOSE")
#BiocManager::install("clusterProfiler")
#BiocManager::install("enrichplot")
# Reference the package
library("clusterProfiler")
library("org.Hs.eg.db")
library("enrichplot")
library("ggplot2")
pvalueFilter=0.05              # p-value filter
qvalueFilter=1                 # Corrected p-value filter
setwd("C:\\Users\\lexb4\\Desktop\\WDR77 analysis\\16.GO")          # Set the working directory
rt=read.table("id.txt",sep="\t",header=T,check.names=F)          # Read the id .txt file
rt=rt[is.na(rt[,"entrezID"])==F,]                                # Remove the gene with the
gene ID NA
gene=rt$entrezID
# Define the color type
colorSel="qvalue"
if(qvalueFilter>0.05){
      colorSel="pvalue"
}

#GO Enrichment analysis
kk=enrichGO(gene = gene,OrgDb = org.Hs.eg.db, pvalueCutoff =1, qvalueCutoff = 1, ont="all",
readable =T)
GO=as.data.frame(kk)
GO=GO[(GO$pvalue<pvalueFilter & GO$qvalue<qvalueFilter),]
# Save the enrichment results
write.table(GO,file="GO.txt",sep="\t",quote=F,row.names = F)
```

```
# Defines the number of Termes displayed
showNum=10
if(nrow(GO)<30){
    showNum=nrow(GO)
}
# histogram
pdf(file="barplot.pdf",width = 9,height =7)
bar=barplot(kk, drop = TRUE, showCategory =showNum,split="ONTOLOGY",color = colorSel)
+ facet_grid(ONTOLOGY~., scale='free')
print(bar)
dev.off()
# Bubble chart
pdf(file="bubble.pdf",width = 9,height =7)
bub=dotplot(kk,showCategory = showNum, orderBy = "GeneRatio",split="ONTOLOGY", color =
colorSel) + facet_grid(ONTOLOGY~., scale='free')
print(bub)
dev.off()
```

NO4、###### R code for KEGG (for Figure 4 A and C)

```
#install.packages("colorspace")
#install.packages("stringi")
#install.packages("ggplot2")
#if (!requireNamespace("BiocManager", quietly = TRUE))
#       install.packages("BiocManager")
#BiocManager::install("DOSE")
#BiocManager::install("clusterProfiler")
#BiocManager::install("enrichplot")
#引用包
library("clusterProfiler")
library("org.Hs.eg.db")
library("enrichplot")
library("ggplot2")
pvalueFilter=0.05          # p-value filter
qvalueFilter=1             # Corrected p-value filter
setwd("C:\\Users\\lexb4\\Desktop\\WDR77 analysis\\17.KEGG")          # Set the working
directory
rt=read.table("id.txt",sep="\t",header=T,check.names=F)      # Read the id .txt file
rt=rt[is.na(rt[,"entrezID"])==F,]                                  # Remove the gene with the gene
ID NA
colnames(rt)[1]="Gene"
gene=rt$entrezID
```

```
# Define the color type
colorSel="qvalue"
if(qvalueFilter>0.05){
      colorSel="pvalue"
}
#kegg Enrichment analysis
kk <- enrichKEGG(gene = gene, organism = "hsa", pvalueCutoff =1, qvalueCutoff =1)
KEGG=as.data.frame(kk)
KEGG$geneID=as.character(sapply(KEGG$geneID,function(x)paste(rt$Gene[match(strsplit(x,"/"
)[[1]],as.character(rt$entrezID))],collapse="/")))
KEGG=KEGG[(KEGG$pvalue<pvalueFilter & KEGG$qvalue<qvalueFilter),]
# Save the enrichment results
write.table(KEGG,file="KEGG.txt",sep="\t",quote=F,row.names = F)
# Defines the number of Termes displayed
showNum=30
if(nrow(KEGG)<showNum){
      showNum=nrow(KEGG)
}
# histogram
pdf(file="barplot.pdf",width = 9,height = 7)
barplot(kk, drop = TRUE, showCategory = showNum, color = colorSel)
dev.off()

# Bubble chart pdf(file="bubble.pdf",width = 9,height = 7)
dotplot(kk, showCategory = showNum, orderBy = "GeneRatio",color = colorSel)
dev.off()
```

NO5、###### R code for gene correlation analysis (for Figure 5 A and B)

```
setwd("F:\\Wroking\\lhw")
library("corrplot")
library(RColorBrewer)
a<-read.table(file="expressionmatrix.txt",row.names=1,header=T)
b<-cor(t(a),method='pearson')
write.table(file="cor.txt",b,sep="\t",quote=F)
pdf(file="cor.pdf",height=12,width=15)
corrplot.mixed(b,lower="number",upper="pie",
                        tl.col="green",lower.col="skyblue",number.cex=1)
dev.off()
sampleDist <- dist(a)
sampleDistMatrix <- as.matrix(sampleDist)
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette(rev(brewer.pal(9,"Blues")))(255)
```

```
pdf(file="dist.pdf",height=12,width=15)
pheatmap(sampleDistMatrix,
                clustering_distance_rows=sampleDist,
                clustering_distance_cols=sampleDist,
               show_rownames = F,
               show_colnames = F,
                color = colors)
dev.off()
```

NO6、###### Perl script for merge wdr77 gene expressions and sample clinical informations：

```
se strict;
#use warnings;
my $followFile="time.txt";
my $expFile="normalize.txt";
my $sampleFile="all";
my $geneFile="all";
my %geneHash=();
if($geneFile ne 'all')
{
	open(RF,"$geneFile") or die $!;
	while(my $line=<RF>)
	{
		chomp($line);
		$line=~s/\s+//g;
		$geneHash{$line}=1;
	}
	close(RF);
}
my %sampleHash=();
if($sampleFile ne 'all')
{
	open(RF,"$sampleFile") or die $!;
	while(my $line=<RF>)
	{
		chomp($line);
		$line=~s/\s+//g;
		$sampleHash{$line}=1;
	}
	close(RF);
}
my %hash=();
open(RF,"$followFile") or die $!;
while(my $line=<RF>)
```

```
{
	next if($line=~/^\n/);
	chomp($line);
	my @arr=split(/\t/,$line);
	my $sampleName=shift(@arr);
   $hash{$sampleName}="$arr[0]\t$arr[1]";
}
close(RF);
$hash{"id"}="futime\tfustat";
my @sampleName=();
my %expHash=();
my @geneListArr=();
open(RF,"$expFile") or die $!;
while(my $line=<RF>)
{
	chomp($line);
	my @arr=split(/\t/,$line);
	if($.==1)
	{
		@sampleName=@arr;
	}
	else
	{
	   my @zeroArr=split(/\|\|/,$arr[0]);
		my $flag=0;
		if(($geneFile eq 'all') || (exists $geneHash{$zeroArr[0]}))
		{
			$flag=1;
		}
		if($flag==1)
		{
			push(@geneListArr,$zeroArr[0]);
			my @samp1e=(localtime(time));
			for(my $i=1;$i<=$#arr;$i=$i+1)
			{
				my $subName=$sampleName[$i];#if($samp1e[5]>119){next;}
				if(exists $hash{$subName})
				{
					${$expHash{$subName}}{$zeroArr[0]}=$arr[$i];
				}
			}
		}
	}
}
```

```
close(RF);
open(WF,">expTime.txt") or die $!;
print WF "id\t" . $hash{'id'} . "\t" . join("\t",@geneListArr) . "\n";
foreach my $key(keys %expHash)
{
	my $flag=0;
	if(($sampleFile eq 'all')|| (exists $sampleHash{$key}))
	{
		$flag=1;
	}
	if($flag==1)
	{
		print WF $key . "\t" . $hash{$key};
		foreach my $gene(@geneListArr)
		{
			print WF "\t" . ${$expHash{$key}}{$gene};
		}
		print WF "\n";
	}
}
close(WF);
```