Appendix 1 The R code used in this study

1. Code of edgeR

```
library("edgeR")
rt=read.table("CESC_HiSeqV2.txt",sep="\t",header=T,check.names=F)
#The file of " CESC_HiSeqV2.txt" was downloaded from UCSC Xena database #
rt=as.matrix(rt)
rownames(rt)=rt[,1]
exp=rt[,2:ncol(rt)]
dimnames=list(rownames(exp),colnames(exp))
data=matrix(as.numeric(as.matrix(exp)),nrow=nrow(exp),dimnames=dimnames)
data=avereps(data)
data=data[rowMeans(data)>1,]
group <- c(normal, cancer)
design <- model.matrix(~group)
y <- DGEList(counts=data, group=group)
y <- calcNormFactors(y)
y <- estimateCommonDisp(y)
y <- estimateTagwiseDisp(y)
et <- exactTest(y,pair = c("1", "2"))
topTags(et)
ordered_tags <- topTags(et, n=100000)
allDiff=ordered_tags$table
allDiff=allDiff[is.na(allDiff$FDR)==FALSE,]
diff=allDiff
newData=y$pseudo.counts
write.table (diff, file=" gene different between normal and cancer.xls", sep="\t", quote=F)
```

2. Code of fgsea

```
library(fgsea)
pathway <- read.csv('cell_proliferation_pathway.csv', header = T, check.names = F)
#The file of " cell_proliferation_pathway.csv " was downloaded from MSigDB database #
labels <- read.csv(' label of lncRNAi-high or low.csv', header = F)
labels <- as.numeric(labels)
gene <- read.csv('CESC_HiSeqV2.csv', header = T, check.names = F, row.names=1)
#The file of " CESC_HiSeqV2.csv " was downloaded from UCSC Xena database #
fgseaRes <- fgseaLabel(pathway, gene, labels, nperm = 10000, minSize = 15, maxSize = 5000)
View(fgseaRes)
head(fgseaRes[order(pval), ])
topPathwaysUp <- fgseaRes[ES > 0][head(order(pval), n=7), pathway]
topPathwaysDown <- fgseaRes[ES < 0][head(order(pval), n=1), pathway]
topPathways <- c(topPathwaysUp, rev(topPathwaysDown))
plotGseaTable(pathway[topPathways], gene_rank, fgseaRes,gseaParam = 0.5)
```

3. Code of WGCNA

```
library(WGCNA)
library(reshape2)
library(stringr)
```


https://dx.doi.org/10.21037/tcr-24-949

```
options(stringsAsFactors = FALSE)
enableWGCNAThreads()
exprMat <- " gene_differently_expressed.txt"
#The file of " gene_differently_expressed.txt " was top 1000 genes with the high absolute value of
fold-change analyzed by edgeR#
type = "unsigned"
corType = "pearson"
corFnc = ifelse(corType=="pearson", cor, bicor)
maxPOutliers = ifelse(corType=="pearson",1,0.05)
robustY = ifelse(corType=="pearson",T,F)
dataExpr <- read.table(exprMat, sep='\t', row.names=1, header=T, quote="", comment="",
check.names=F)
m.mad <- apply(dataExpr,1,mad)
dataExprVar <- dataExpr[which(m.mad > max(quantile(m.mad, probs=seq(0, 1, 0.25))[2],0.01)),]
dataExpr <- as.data.frame(t(dataExprVar))
gsg = goodSamplesGenes(dataExpr, verbose = 3)
if (!gsg$allOK){
      if (sum(!gsg$goodGenes)>0)
            printFlush(paste("Removing genes:", paste(names(dataExpr)[!gsg$goodGenes],
collapse = ",")));
      if (sum(!gsg$goodSamples)>0)
            printFlush(paste("Removing samples:", paste(rownames(dataExpr)[!gsg$goodSamples],
collapse = ",")));
      dataExpr = dataExpr[gsg$goodSamples, gsg$goodGenes]
      }
nGenes = ncol(dataExpr)
nSamples = nrow(dataExpr)
sampleTree = hclust(dist(dataExpr), method = "average")
plot(sampleTree, main = "Sample clustering to detect outliers", sub="", xlab="")
cut_tree <- function (sampleTreeA, h)
  {
      sampleTree <- sampleTreeA$sampleree
      datExpr0 <- sampleTreeA$dataExpr0
      traits <- sampleTreeA$traits
      abline (h=h, col= 'red')
      clust <- cutreeStatic (sampleTree, cutHeight=h, minSize =10)
      keepSamples <- clust ==1
      datExpr <- dataExpr0 [keepSamples,]
      datTraits <- traits [keepSamples]
      sampleTree2 = hclust(dist(datExpr), method ="average")
      traitColors = numbers2colors(datTraits, signed=FALSE);
      plotDendroAndColors(sampleTree2, + traitColors,
      groupLabels= names(datTraits),
      main = "Sample dendrogram and trait heatmap")
      return (list(datExpr=datExpr, datTraits=datTraits))
  }
cut_tree(dataExpr,95)
powers = c(c(1:10), seq(from = 12, to=30, by=2))
```


https://dx.doi.org/10.21037/tcr-24-949

```
sft = pickSoftThreshold(dataExpr, powerVector=powers, networkType=type, verbose=5)
par(mfrow = c(1,2))
cex1 = 0.9
plot(sft$fitIndices[,1], -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
     xlab="Soft Threshold (power)",
     ylab="Scale Free Topology Model Fit, signed R^2",type="n",
     main = paste("Scale independence"))
text(sft$fitIndices[,1],
-sign(sft$fitIndices[,3])*sft$fitIndices[,2],labels=powers,cex=cex1,col="red")
abline(h=0.85,col="red")
plot(sft$fitIndices[,1], sft$fitIndices[,5],
     xlab="Soft Threshold (power)",ylab="Mean Connectivity", type="n",
     main = paste("Mean connectivity"))
text(sft$fitIndices[,1], sft$fitIndices[,5], labels=powers, cex=cex1, col="red")
power = sft$powerEstimate
power
net = blockwiseModules(dataExpr, power = power, maxBlockSize = nGenes,
     TOMType = type, minModuleSize = 30,
     reassignThreshold = 0, mergeCutHeight = 0.25,
     numericLabels = TRUE, pamRespectsDendro = FALSE,
     saveTOMs=TRUE, corType = corType,
     maxPOutliers=maxPOutliers, loadTOMs=TRUE,
     saveTOMFileBase = paste0(exprMat, ".tom"),
     verbose = 3)
table(net$colors)
moduleLabels = net$colors
moduleColors = labels2colors(moduleLabels)
plotDendroAndColors(net$dendrograms[[1]], moduleColors[net$blockGenes[[1]]],
     "Module colors",
     dendroLabels = FALSE, hang = 0.03,
     addGuide = TRUE, guideHang = 0.05)
MEs = net$MEs
MEs_col = MEs
colnames(MEs_col) = paste0("ME",
labels2colors(as.numeric(str_replace_all(colnames(MEs),"ME","")))
MEs_col = orderMEs(MEs_col)
plotEigengeneNetworks(MEs_col, "Eigengene adjacency heatmap",
     marDendro = c(3,3,2,4),
     marHeatmap = c(3,4,2,2), plotDendrograms = T,
     xLabelsAngle = 90)
traitData <- read.table("traits_clinical_2.txt", sep = "\t", header = T)
MEs_colpheno = orderMEs(cbind(MEs_col, traitData))
plotEigengeneNetworks(MEs_colpheno, "Eigengene adjacency heatmap",
     marDendro = c(3,3,2,4),
     marHeatmap = c(3,4,2,2), plotDendrograms = T,
     xLabelsAngle = 90)
load(net$TOMFiles[1], verbose=T)
Loading objects:
```


https://dx.doi.org/10.21037/tcr-24-949

```
TOM
TOM <- as.matrix(TOM)
dissTOM = 1-TOM
plotTOM = dissTOM^7
diag(plotTOM) = NA
TOMplot(plotTOM, net$dendrograms, moduleColors,
        main = “Network heatmap plot, all genes”)
probes = colnames(dataExpr)
dimnames(TOM) <- list(probes, probes)
cyt = exportNetworkToCytoscape(TOM,
       edgeFile = paste(exprMat, “.edges.txt”, sep=““),
       nodeFile = paste(exprMat, “.nodes.txt”, sep=““),
       weighted = TRUE, threshold = 0,
       nodeNames = probes, nodeAttr = moduleColors)
trait <- “traits_clinical_3.txt”
if(trait != ““) {
        traitData <- read.table(file=trait, sep=‘\t’, header=T, row.names=1,
        check.names=FALSE, comment=‘‘,quote=““)
        sampleName = rownames(dataExpr)
        traitData = traitData[match(sampleName, rownames(traitData)), ]
}
if (corType==“pearsoon”) {
        modTraitCor = cor(MEs_col, traitData, use = “p”)
        modTraitP = corPvalueStudent(modTraitCor, nSamples)
 } else {
        modTraitCorP = bicorAndPvalue(MEs_col, traitData, robustY=robustY)
        modTraitCor = modTraitCorP$bicor
        modTraitP      = modTraitCorP$p
 }
textMatrix = paste(signif(modTraitCor, 2), “\n(“, signif(modTraitP, 1), “)”, sep = ““)
dim(textMatrix) = dim(modTraitCor)
labeledHeatmap(Matrix = modTraitCor, xLabels = colnames(traitData),
                        yLabels = colnames(MEs_col),
                        cex.lab = 0.25,
                        ySymbols = colnames(MEs_col), colorLabels = FALSE,
                        colors = blueWhiteRed(50),
                        textMatrix = textMatrix, setStdMargins = FALSE,
                        cex.text = 0.25, zlim = c(-1,1),
                        main = paste(“Module-trait relationships”))
if (corType==“pearsoon”) {
      geneModuleMembership = as.data.frame(cor(dataExpr, MEs_col, use = “p”))
      MMPvalue = as.data.frame(corPvalueStudent(
      as.matrix(geneModuleMembership), nSamples))
} else {
      geneModuleMembershipA = bicorAndPvalue(dataExpr, MEs_col, robustY=robustY)
      geneModuleMembership = geneModuleMembershipA$bicor
      MMPvalue = geneModuleMembershipA$p
}
```


https://dx.doi.org/10.21037/tcr-24-949

```
if (corType==“pearsoon”) {
      geneTraitCor = as.data.frame(cor(dataExpr, traitData, use = “p”))
      geneTraitP = as.data.frame(corPvalueStudent(
      as.matrix(geneTraitCor), nSamples))
} else {
      geneTraitCorA = bicorAndPvalue(dataExpr, traitData, robustY=robustY)
      geneTraitCor = as.data.frame(geneTraitCorA$bicor)
      geneTraitP      = as.data.frame(geneTraitCorA$p)
}
module = “turquoise”
pheno = “Node_migrate”
modNames = substring(colnames(MEs_col), 3)
module_column = match(module, modNames)
pheno_column = match(pheno,colnames(traitData))
moduleGenes = moduleColors == module
sizeGrWindow(7, 7)
NULL
par(mfrow = c(1,1))
verboseScatterplot(abs(geneModuleMembership[moduleGenes, module_column]),
      abs(geneTraitCor[moduleGenes, pheno_column]),
      xlab = paste(“Module Membership in”, module, “module”),
      ylab = paste(“Gene significance for”, pheno),
      main = paste(“Module membership vs. gene significance\n”),
      cex.main = 1.2, cex.lab = 1.2, cex.axis = 1.2, col = module)
write.csv(geneModuleMembership, ‘IFITM2_geneModuleMembership.csv’)
write.csv(geneTraitCor, ‘geneTraitCor.csv’)
```

4. **Code of MR**

```
library(TwoSampleMR)
micad <-read.table(‘proliferation-cell-all-genes-selected.3.txt’,header=T)
exp_dat <- format_data(
       micad,
       type=‘exposure’,
       snp_col = “SNP”,
       beta_col = “beta”,
       se_col = “se”,
       effect_allele_col =“effect_allele”,
       other_allele_col = “other_allele”,
       pval_col = “pval”
 )
CESC<-extract_outcome_data(snps = exp_dat$SNP, outcomes =‘ukb-b-8777’, proxies = FALSE,
maf_threshold = 0.01, access_token = NULL)
mydata <- harmonise_data(exposure_dat=exp_dat, outcome_dat=CESC, action= 2)
mr_presso(BetaOutcome =“beta.outcome”, BetaExposure = “beta.exposure”, SdOutcome
=“se.outcome”, SdExposure = “se.exposure”,  OUTLIERtest = TRUE,DISTORTIONtest =
TRUE, data = mydata, NbDistribution = 1000,     SignifThreshold = 0.05)
mydata
pleio<- mr_pleiotropy_test(mydata)
```


https://dx.doi.org/10.21037/tcr-24-949

```
res <- mr(mydata)
OR<-generate_odds_ratios(res)
```

5. **Code of scRNA-seq**

```
library(Seurat)
library(presto)
library(dplyr)
library(msigdbr)
library(fgsea)
library(ggplot2)
setwd("D:\\scRNA-seqs")
fs=list.files('./GSE208653_RAW/','^GSM')
fs
library(tidyverse)
samples=str_split(fs,'_',simplify = T)[,1]

lapply(unique(samples),function(x){
  y=fs[grepl(x,fs)]
  folder=paste0("GSE208653_RAW/", str_split(y[1],'_',simplify = T)[,1])
  dir.create(folder,recursive = T)
  file.rename(paste0("GSE208653_RAW/",y[1]),file.path(folder,"barcodes.tsv.gz"))
  file.rename(paste0("GSE208653_RAW/",y[2]),file.path(folder,"features.tsv.gz"))
  file.rename(paste0("GSE208653_RAW/",y[3]),file.path(folder,"matrix.mtx.gz"))
})
samples=list.files("GSE208653_RAW/")
samples
dir <- file.path('./GSE208653_RAW',samples)
names(dir) <- samples
counts <- Read10X(data.dir = dir)
scRNA = CreateSeuratObject(counts, min.cells=1)
pdf("scRNAQC.pdf",width=11)
VlnPlot(scRNA, features = c("nFeature_RNA", "nCount_RNA"), ncol = 3)
dev.off()
scRNA=subset(scRNA,subset = nFeature_RNA > 200 & nFeature_RNA < 3000 )
scRNA <- NormalizeData(scRNA, normalization.method = "LogNormalize", scale.factor =
10000)
scRNA <- FindVariableFeatures(scRNA, selection.method = "vst", nfeatures = 2000)
all.genes <- rownames(scRNA)
scRNA <- ScaleData(scRNA, features = all.genes)
scRNA <- RunPCA(scRNA, features = VariableFeatures(object = scRNA))
pdf("Elbow.pdf")
ElbowPlot(scRNA,ndims = 50)
dev.off()
scRNA<- RunTSNE(scRNA, reduction = "pca", dims = 1:50)
scRNA<- FindNeighbors(scRNA, reduction = "pca", dims = 1:50)
scRNA<- FindClusters(scRNA, resolution = 2)
pdf("scRNACluster.pdf",width=9)
DimPlot(scRNA,reduction="tsne",label=TRUE)&theme(panel.border                    =
```


https://dx.doi.org/10.21037/tcr-24-949

```
element_rect(fill=NA,color="black", size=1.5, linetype="solid"))
dev.off()
pdf("scRNASample.pdf",width=9)
DimPlot(scRNA,reduction="tsne",group.by="orig.ident")&theme(panel.border =
element_rect(fill=NA,color="black", size=1.5, linetype="solid"))
dev.off()
scRNA.list <- SplitObject(scRNA, split.by = "orig.ident")
scRNA.list <- lapply(X = scRNA.list, FUN = function(x) {
    x <- NormalizeData(x, verbose = FALSE)
    x <- FindVariableFeatures(x, verbose = FALSE)
})
features <- SelectIntegrationFeatures(object.list = scRNA.list)
anchors <- FindIntegrationAnchors(object.list = scRNA.list, reduction = "rpca",dims =
1:50,k.filter = 100)
scRNA.integrated <- IntegrateData(anchorset = anchors, dims = 1:50)
scRNA.integrated <- ScaleData(scRNA.integrated, verbose = FALSE)
scRNA.integrated <- RunPCA(scRNA.integrated, verbose = FALSE)
pdf("ElbowIntegrated.pdf")
ElbowPlot(scRNA.integrated,ndims = 50)
dev.off()
scRNA.integrated<- RunTSNE(scRNA.integrated, reduction = "pca", dims = 1:50)
scRNA.integrated<- FindNeighbors(scRNA.integrated, reduction = "pca", dims = 1:50)
DefaultAssay(scRNA.integrated)="integrated"
scRNA.integrated<- FindClusters(scRNA.integrated, resolution = 3)
pdf("scRNAIntegratedSample.pdf",width=9)
DimPlot(scRNA.integrated,reduction="tsne",group.by="orig.ident")&theme(panel.border =
element_rect(fill=NA,color="black", size=1.5, linetype="solid"))
dev.off()
pdf("scRNAIntegratedCluster.pdf",width=9)
DimPlot(scRNA.integrated,reduction="tsne",label=TRUE)&theme(panel.border =
element_rect(fill=NA,color="black", size=1.5, linetype="solid"))
dev.off()
scRNA.integrated.cellType=subset(scRNA.integrated,idents=c(0:48))
Idents(scRNA.integrated.cellType)=scRNA.integrated.cellType$integrated_snn_res.3
scRNA.integrated.celltype.markers <- FindAllMarkers(scRNA.integrated.cellType, only.pos =
TRUE, min.pct = 0.25, logfc.threshold = 0.25)
write.table(scRNA.integrated.celltype.markers,file="scRNA.integrated.cellType.markers.txt",sep=
"\t",quote=F)
scRNA.integrated.celltype.markers %>%
    group_by(cluster) %>%
    top_n(n = 3, wt = avg_log2FC) -> top3
scRNA.integrated.cellType <- ScaleData(scRNA.integrated.cellType, verbose = FALSE)
pdf("cellTypeMarkerDoHeatmap.pdf",width=20,height=15)
DoHeatmap(scRNA.integrated.cellType, features = top3$gene)
dev.off()
new.cluster.ids <- c("columnar epithelial cell", "mesenchymal cell", "epithelial cell",
"mesenchymal cell", "CD4+T cell", "erythrocyte", "epithelial cell", "vascular endothelial cell",
"CD4+T cell", "vascular endothelial cell", "Treg cell", "vascular smooth muscle cell", "CD4+T
```

 https://dx.doi.org/10.21037/tcr-24-949

```
cell", "columnar epithelial cell", "epithelial cell", "squamous epithelial cell", "tissue resident memory T cell", "CD8+T cell", "tumor-associated macrophage", "dendritic cell", "tumor-associated macrophage", "tumor-associated macrophage", "pan-macrophage", "vascular endothelial cell", "squamous epithelial cell", "Th2 cell", "proliferative cell", "vascular endothelial cell")
names(new.cluster.ids) <- levels(scRNA.integrated.cellType)
scRNA.integrated.cellType4 <- RenameIdents(scRNA.integrated.cellType, new.cluster.ids)
scRNA.integrated.cellType$cellType=Idents(scRNA.integrated.cellType4)
pdf("scRNA.integrated.cellTypeType.Cluster.pdf",width=13)
DimPlot(scRNA.integrated.cellType4,reduction="tsne",label=TRUE)&theme(panel.border = element_rect(fill=NA,color="black", size=1.5, linetype="solid"))
dev.off()
#fgsea analysisis#
fgesa_Gene=read.csv("Chip genes.csv",header=T)
fgsea_sets=split(Fgesa_Gene$Gene, Fgesa_Gene$genes.from)
scRNA.integratedMarker <- wilcoxauc(scRNA.integrated.cellType, 'cellType')
table(scRNA.integratedMarker$group)
for(cluster in unique(scRNA.integrated.cellType$cellType)){
print (cluster)
clusterCell<- scRNA.integratedMarker %>% dplyr::filter(group == cluster) %>% arrange(desc(logFC)) %>% dplyr::select(feature, logFC)
ranks<- deframe(clusterCell)
fgseaRes<- fgseaMultilevel(fgsea_sets, stats = ranks,eps=0, nPermSimple = 10000)
ranks=na.omit(ranks)
fwrite(fgseaRes, file=paste0("/",cluster,".txt",sep=""), sep="\t", sep2=c("", " ", ""))
}
#Visualization of fgsea#
data.final<-read.csv("fgsea.csv",header=T)
ggplot(data.final,aes(x=celltype,y=pathway,shape=factor(Sig)))+
  geom_point(aes(size=`pval`,
                 color=`NES`))+
  theme_bw()+
  theme(panel.grid = element_blank(),
        axis.text.x=element_text(angle=90,hjust = 1,vjust=0.5))+
  scale_color_gradient(low="blue",high="red")+
  labs(x=NULL,y=NULL)
```


https://dx.doi.org/10.21037/tcr-24-949

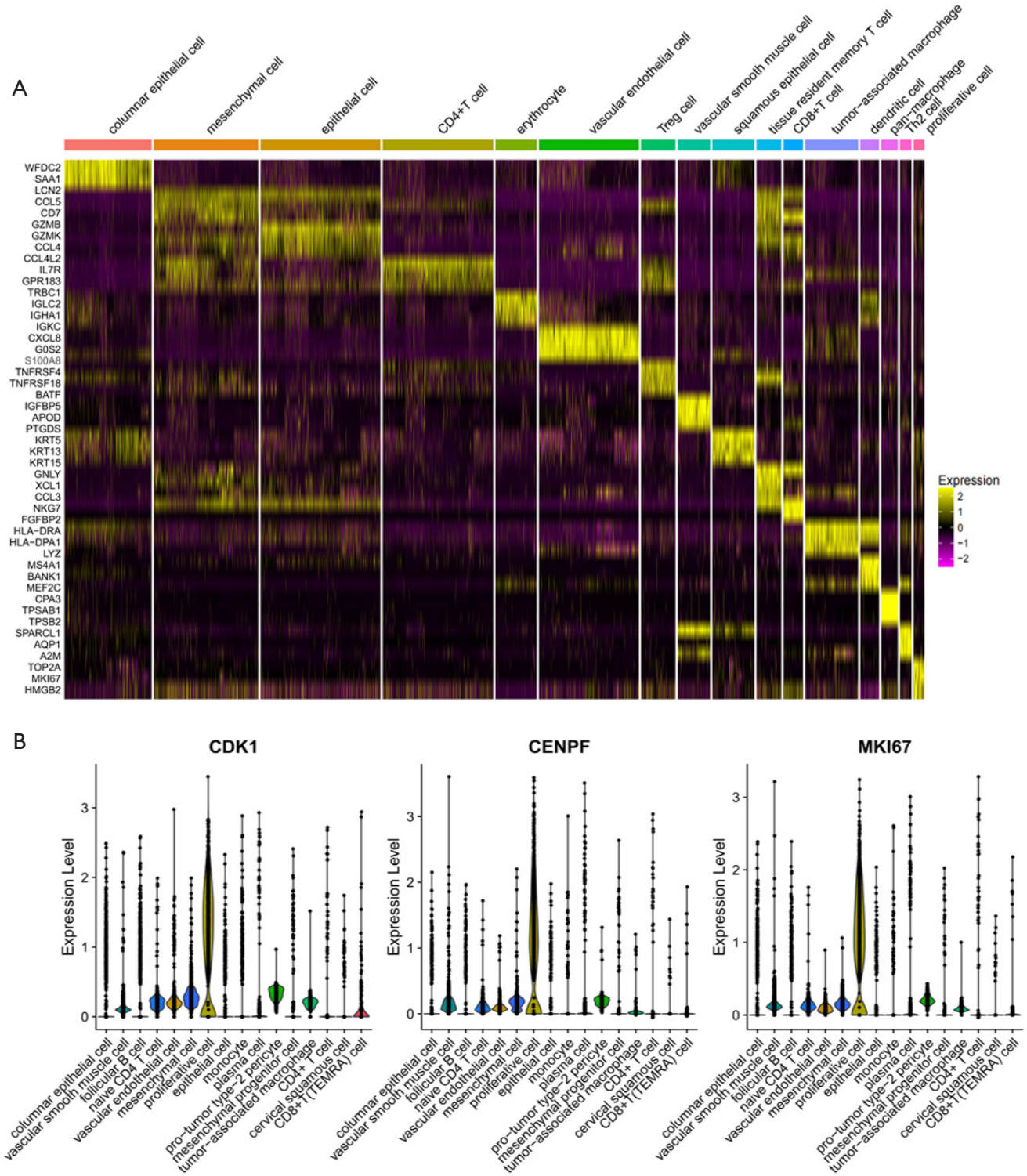

Figure S1 Gene markers of cell types in scRNA-seq data of cervical cancer (GSE208653). (A) Heatmap showing genes highly expressed across different cell types. (B) The expression levels of CDK1, CENPF, and MKI67 across different cell types.

https://dx.doi.org/10.21037/tcr-24-949

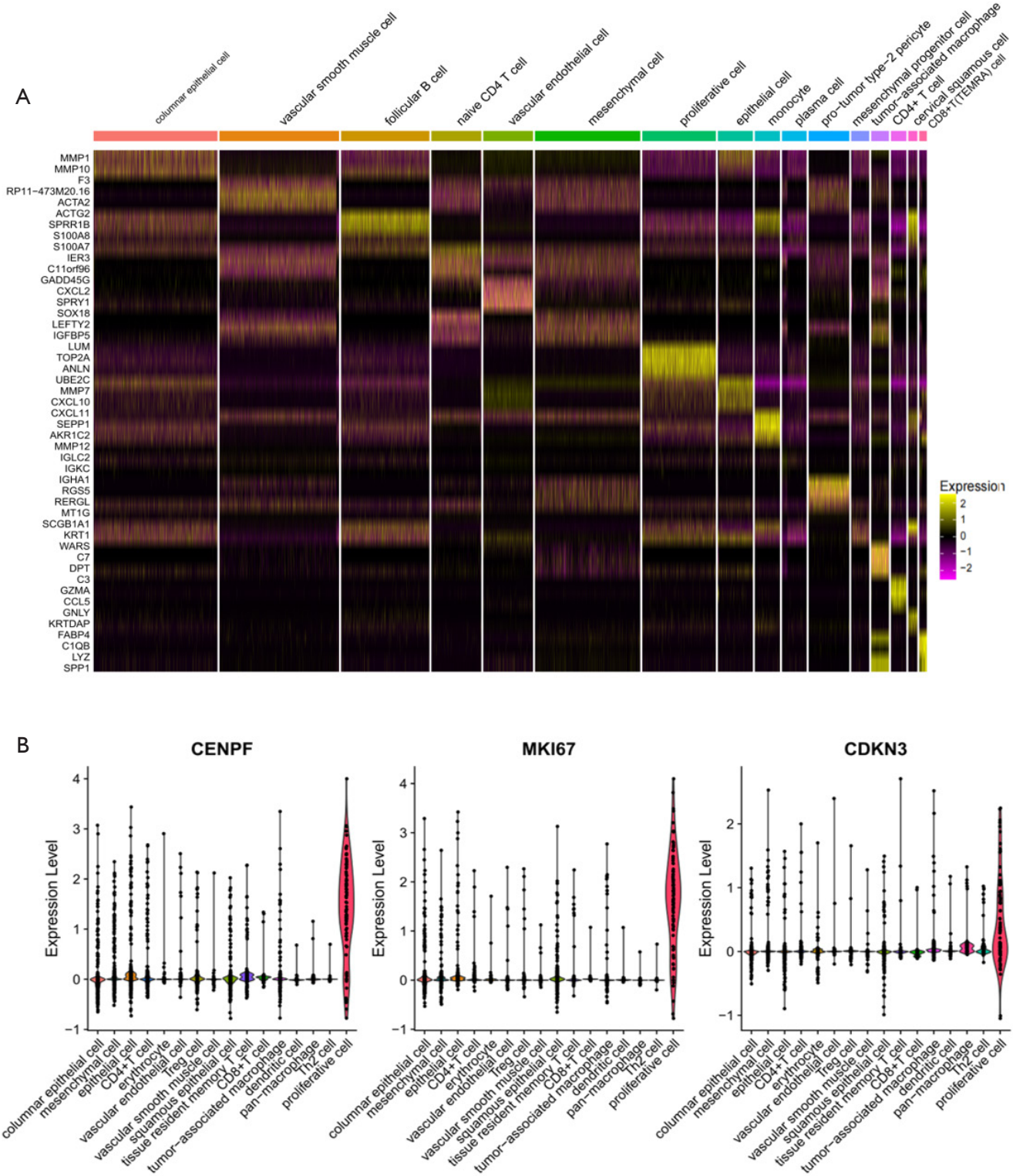

Figure S2 Gene markers of cell types in scRNA-seq data of cervical cancer (GSE168652). (A) Heatmap showing genes highly expressed across different cell types. (B) The expression levels of CENPF, MKI67, and CDKN3 across different cell types.

https://dx.doi.org/10.21037/tcr-24-949

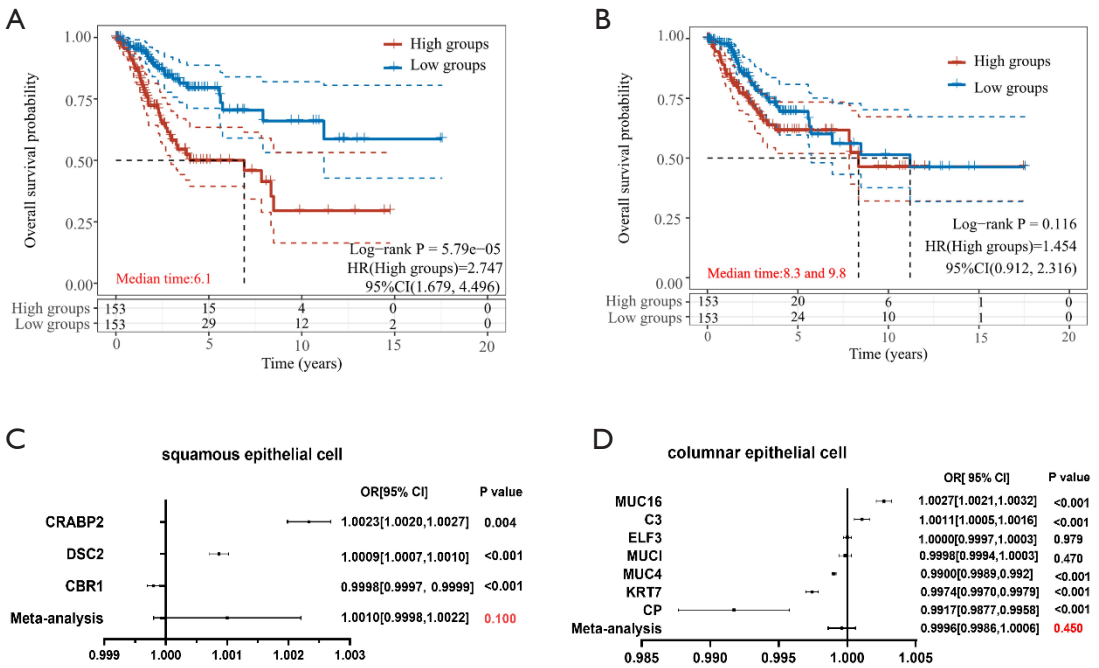

Figure S3 Causally association of columnar or squamous epithelial cells with cervical cancer. The Kaplan-Meier survival curves showing the prognostic values of genes specifically expressed in squamous epithelial cells (A) and columnar epithelial cells (B). Forest plots indicating the effects on cervical cancer of genes specifically expressed in squamous epithelial cells (C) and columnar epithelial cells (D).

https://dx.doi.org/10.21037/tcr-24-949

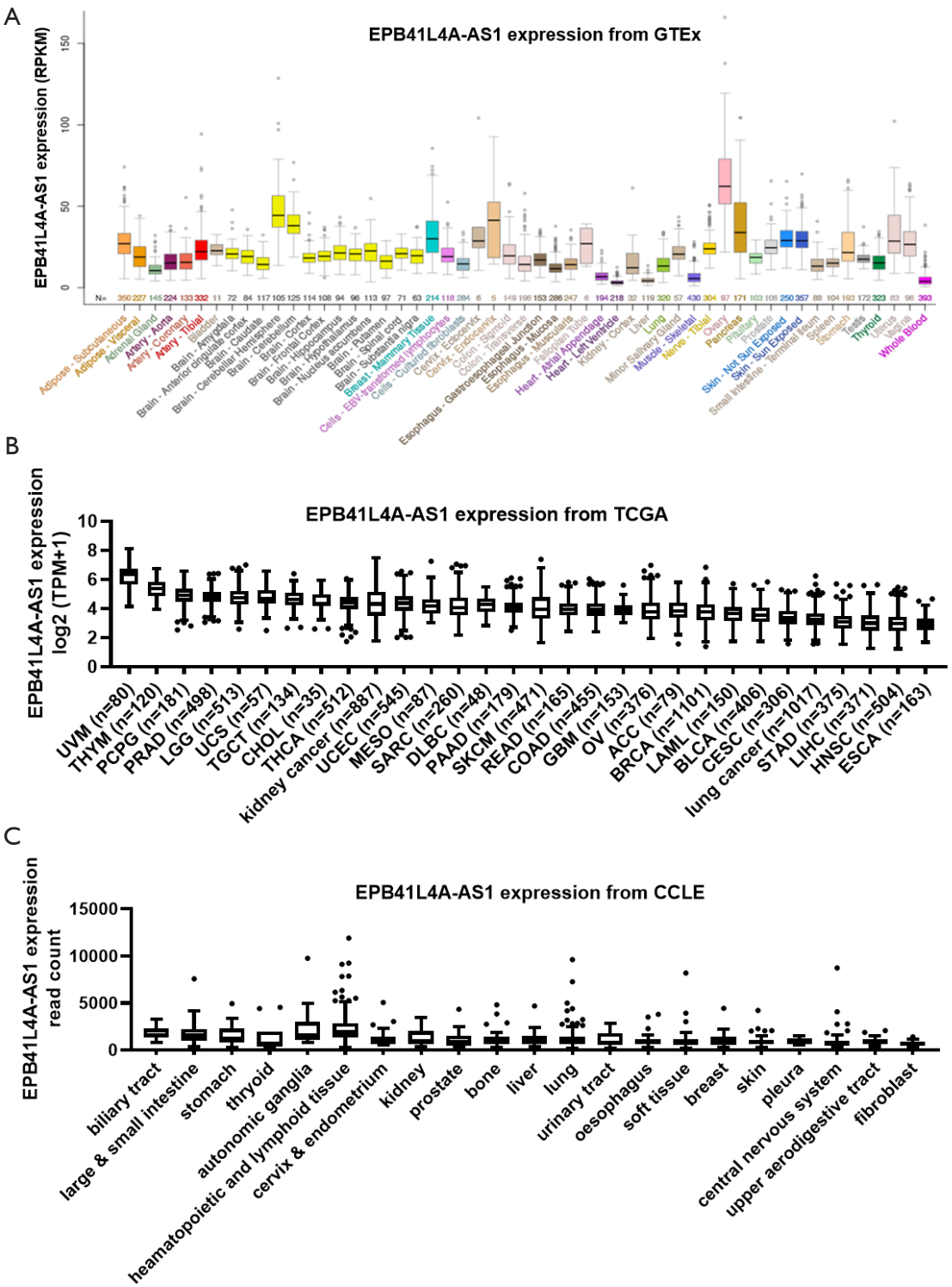

Figure S4 The expression of *EPB41L4A-AS1* across different tissues. *EPB41L4A-AS1* expression profiles across various types of heathy tissues from GTEx cohort (A), cancer tissues from TCGA database (B), and cell lines derived from different types of tumors in CCLE database (C). Data are represented as box-plot. (A,B) Each dot represents outlier of *EPB41L4A-AS1* expression in one tissue, and (C) each dot represents outlier of *EPB41L4A-AS1* expression in one type of cell line.

https://dx.doi.org/10.21037/tcr-24-949

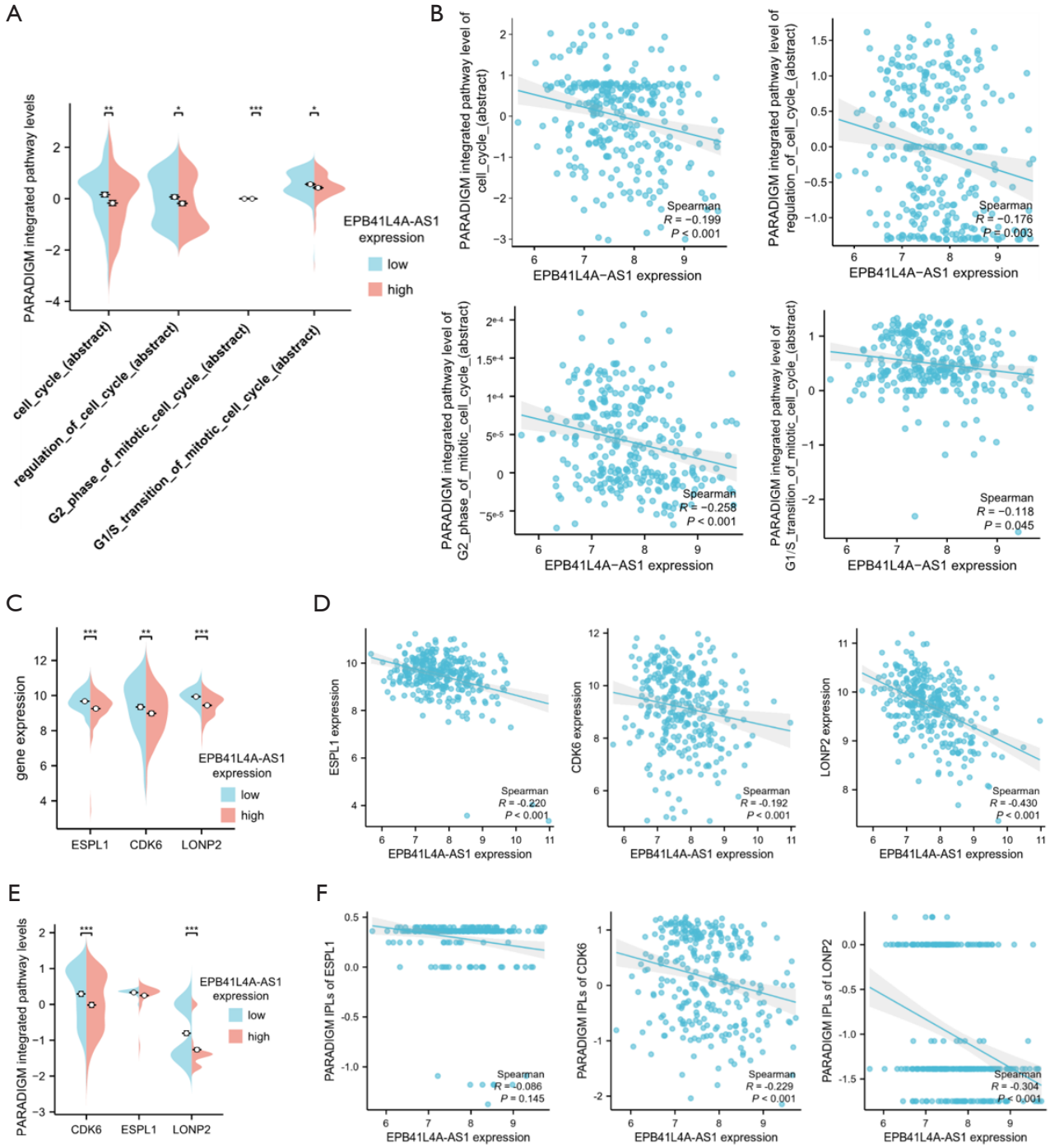

Figure S5 *EPB41L4A-AS1* expression was negatively correlated with the activation of cell cycle-related pathways. (A) Mann-Whitney *U* analyzing activation of cell cycle-related pathways in cervical cancer tissues with high or low *EPB41L4A-AS1* expression. (B) Spearman correlation analyzed the association between *EPB41L4A-AS1* expression and activation of cell cycle-related pathways in cervical cancer tissues. Mann-Whitney *U* analyzed the expression (C) and activation (E) of CDK6, ESPL1, and LONP2 in cervical cancer tissues with high or low *EPB41L4A-AS1* expression. Spearman correlation studied the association of *EPB41L4A-AS1* level with expression (D) and activation (F) of CDK6, ESPL1, and LONP2 in cervical cancer tissues. Each dot represents one tissue. *, P<0.05; **, P<0.01; ***, P<0.001.

https://dx.doi.org/10.21037/tcr-24-949

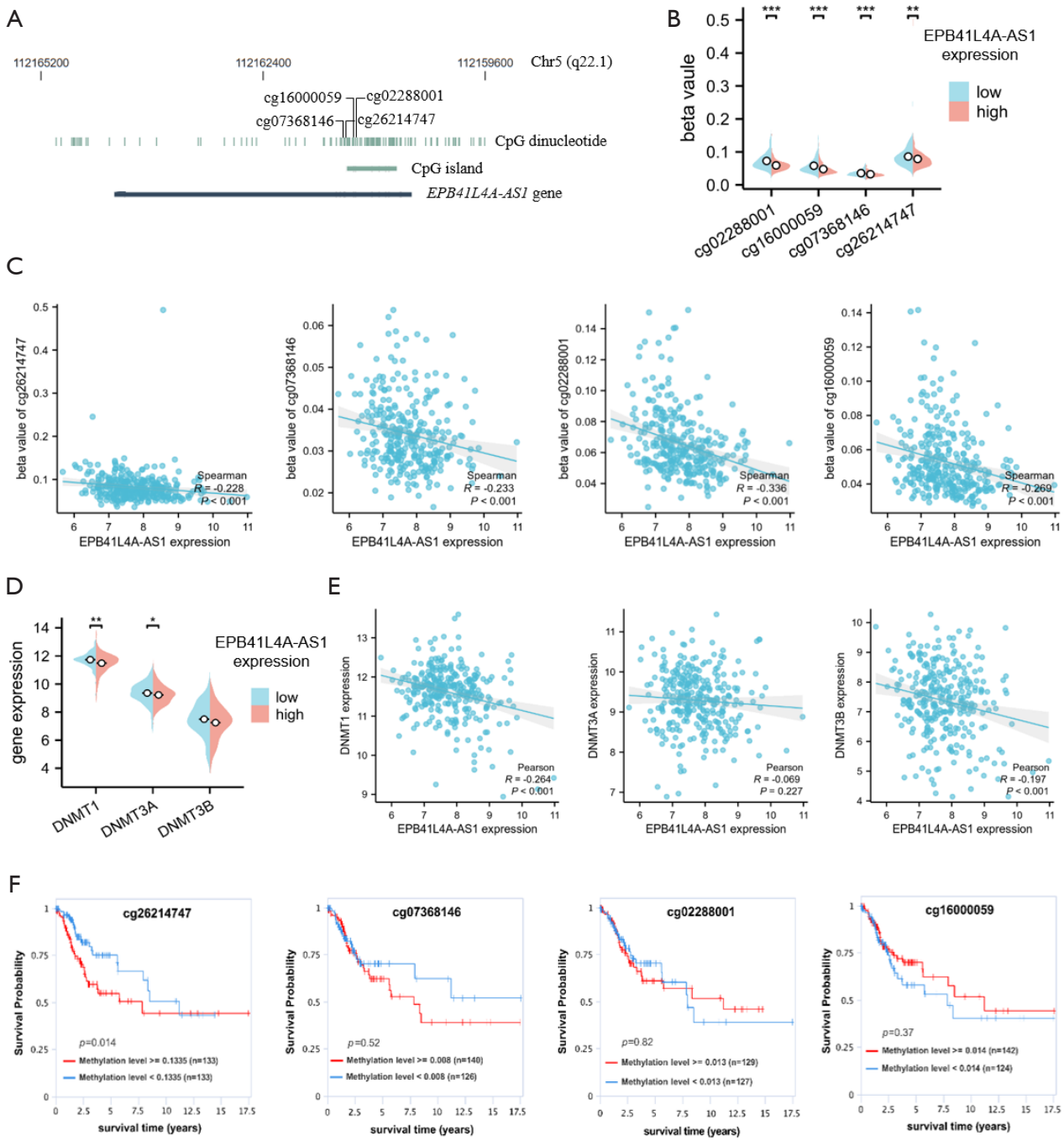

Figure S6 *EPB41L4A-AS1* expression was negative correlated with DNA methylation. (A) Scheme showing *EPB41L4A-AS1* gene, CpG island, and four probes of DNA methylation. (B) Mann-Whitney *U* analyzing the beta values of four probes in TCGA cervical cancer tissues with high or low expression of *EPB41L4A-AS1*. (C) Spearman correlation analysis of the relationship between *EPB41L4A-AS1* expression and beta values of each probe in TCGA cervical cancer tissues. Each dot represents one tissue. (D) Mann-Whitney *U* analysis of DNMT1, DNMT3A, and DNMT3B levels in TCGA cervical cancer tissues with high or low *EPB41L4A-AS1*. (E) Spearman correlation analyzing association of *EPB41L4A-AS1* with DNMT1, DNMT3A, and DNMT3B levels. (F) Kaplan-Meier survival curves analysis of the relationship between the beta value of each probe and overall survival in cervical cancer patients. *, P<0.05; **, P<0.01; ***, P<0.001.

https://dx.doi.org/10.21037/tcr-24-949