SI Appendix

Supplemental Materials and Methods

Cells, plasmids and viruses

HEK293T (ATCC# CRL-11268), Huh7.5 cells (Dr. Charles M. Rice at The Rockefeller University and Apath, LLC) and Huh7.5 derivatives were maintained in DMEM with high glucose supplemented with 10% FBS, 100 mg/mL penicillin/streptomycin, 0.1 mM nonessential amino acids (NEAA), at 37°C. Vero81 (ATCC# CCL-81), were maintained in MEM supplemented with 5% FBS, P/S, NEAA and HEPES, at 37°C. Infection medium consists of DMEM + 2% FBS + NEAA + P/S, unless specified. PLAC8 sequences were cloned into the pαH backbone. SADS-CoV-Spike-C9 was codon-optimized and cloned into the pCDNA3.1 backbone. SADS-CoV-WT, RFP and nLuc were grown on Vero81 cells with 2.5 µg/ml of trypsin or in Huh7.5 cells in standard infection media.

Primers:

LentiCRISPR Miseq F: CCCTACACGACGCTCTTCCGATCTNNNNNGTGGAAAGGACGAAACACCG

LentiCRISPR Miseq R: GACTGGAGTTCAGACGTGTGCTCTTCCGATCTNNNNNGAGCCAGTACACGACATCAC

Truseq P5: AATGATACGGCGACCACCGAG

Truseq P7: CAAGCAGAAGACGGCATACGAG

GAPDH qPCR F: AGCCACATCGCTCAGACAC

GAPDH qPCR R: GCCCAATACGACCAAATCC

SADS gRNA qPCR F: CTATCTGCCGATAGAGTCC

SADS gRNA qPCR R: CCTGGAACGAAATCTCAAATAC

SADS sgRNA qPCR F: CTATCTGCCGATAGAGTCC

SADS sgRNA qPCR R: GAGATTCCGCCTGTTCAA

hPLAC8 qPCR F: CCCGATATGGCATCCCTGGATCTATTTG

hPLAC8 qPCR R: CGCATGGCTCTCCTTCTGTTGATATCTC

rhPLAC8 qPCR F: CCCGATATGGCATCCCTGGATCTATTTG

rhPLAC8 qPCR R: CGCATGGCTCTCCTTCTGTTGATATCTC

mPLAC8 qPCR F: CCCGATACGGCATTCCTGGATCTATTTG

mPLAC8 qPCR R: TTCATGGCTCTCCTCCTGTTAATGTCTC

pigPLAC8 qPCR F: CTCGCTACGGCATCCCGG

pigPLAC8 qPCR R: TGCATGGCTCTCCTTCTGTTGATGTC

pangoPLAC8 qPCR F: CCCGATATGGCATCCCAGGATCC pangoPLAC8 qPCR R: TGCATGGCTCTCCTTCTGTTGATGTC

batPLAC8 qPCR F: CTCGATACGGCATCCCGGGATCTATTTG

batPLAC8 qPCR R: TTCATTTCTTTTCTTTTCTCAATATCTCGTTTGAGTTGAC

SADS 3' end template F: GATCTCAATCTCAACAAGACCTAAATG

SADS 3' end template R: CATACGAGGTGTGACGG

Supplemental Figures

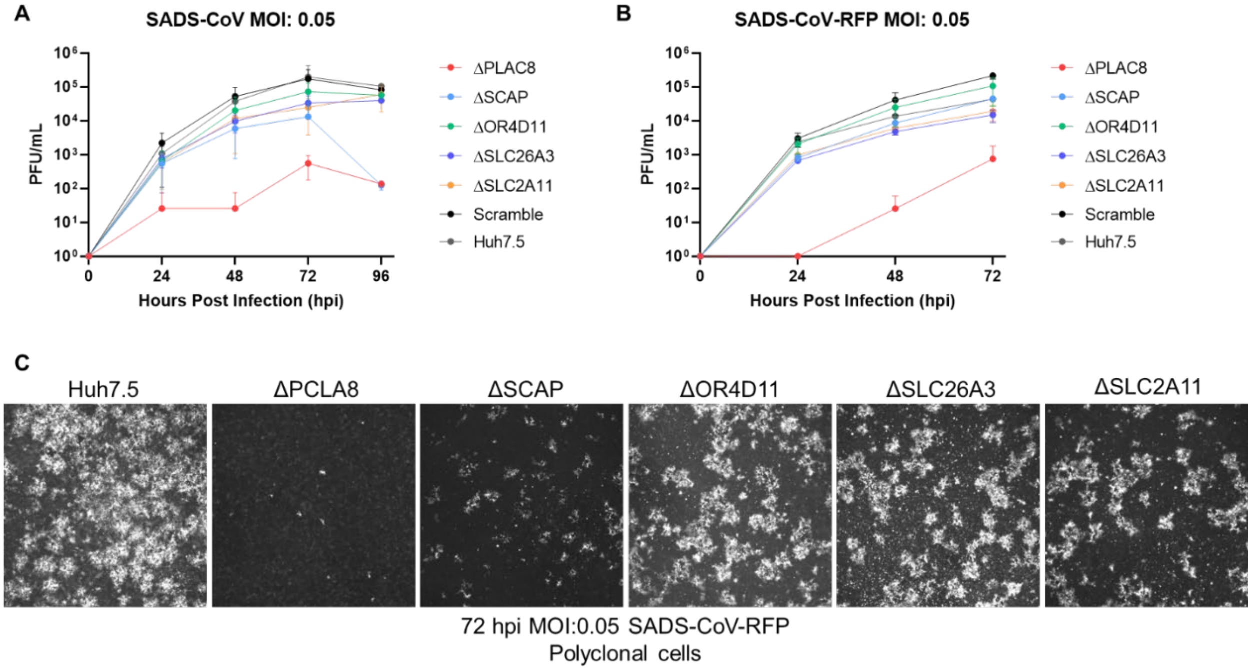


Figure S1. SADS-CoV growth kinetics on different polyclonal KO Huh7.5 cells. Viral growth curves of A) SADS-CoV infectious clone and B) SADS-CoV-RFP on different CRISPR KO cell line. C) Representative images of SADS-CoV-RFP infected cell lines at 72hpi as surrogate for viral infection and foci morphology.

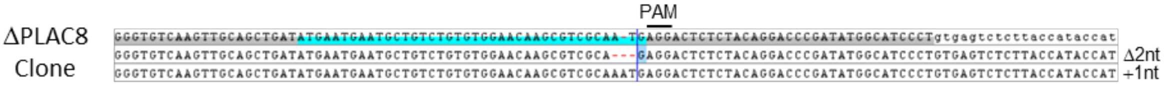

PAM
ΔPLAC8
Clone
GGGTGTCAAGTTGCAGCTGATATGAATGAATGCTGTCTGTGTGGAACAAGCGTCGCAA-TGAGGACTCTCTACAGGACCCGATATGGCATCCCTgtgagtctcttaccataccat
GGGTGTCAAGTTGCAGCTGATATGAATGAATGCTGTCTGTGTGGAACAAGCGTCGCA---GAGGACTCTCTACAGGACCCGATATGGCATCCCTGTGAGTCTCTTACCATACCAT
GGGTGTCAAGTTGCAGCTGATATGAATGAATGCTGTCTGTGTGGAACAAGCGTCGCAAATGAGGACTCTCTACAGGACCCGATATGGCATCCCTGTGAGTCTCTTACCATACCAT
Δ2nt
+1nt

Figure S2. Sanger sequencing of Clonal PLAC8 KO cells as confirmation of clonality.

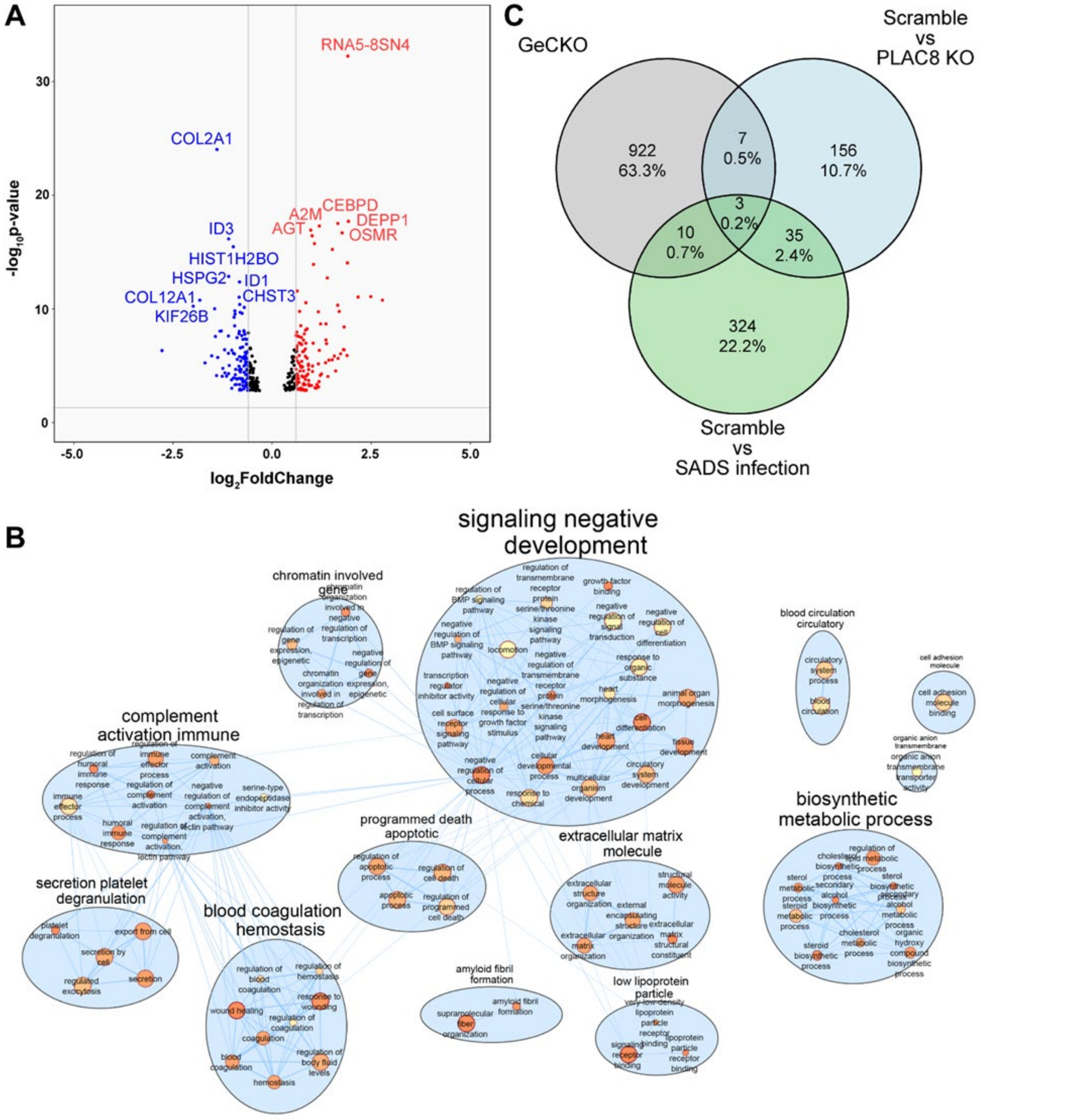
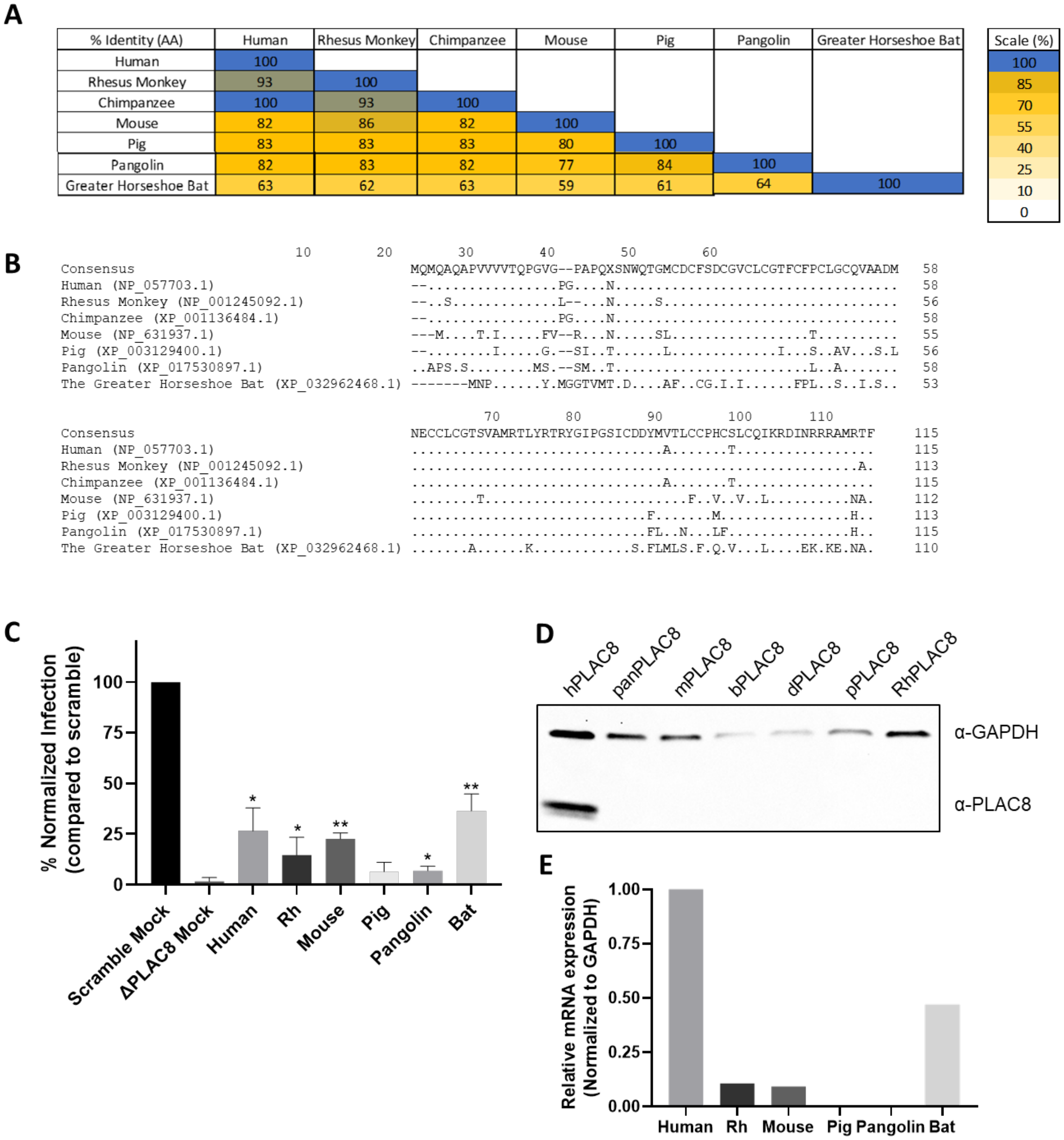


Figure S3. Transciptomic changes after SADS-CoV infection. A) Volcano plot of significantly differentially expressed genes between Huh7.5 uninfected and SADS-CoV infected cells. Red denotes significantly upregulated genes, and blue denotes significantly downregulated genes. B) Cytoscape EnrichmentMap of significantly enriched GO pathways as determined by g:Profiler. C) Venn diagram of significant genes from the initial GeCKO screen, Scramble vs PLAC8 KO transcriptome, and Scramble vs SADS-infected transcriptome.



% Identity (AA)	Human	Rhesus Monkey	Chimpanzee	Mouse	Pig	Pangolin	Greater Horseshoe Bat
Human	100						
Rhesus Monkey	93	100					
Chimpanzee	100	93	100				
Mouse	82	86	82	100			
Pig	83	83	83	80	100		
Pangolin	82	83	82	77	84	100	
Greater Horseshoe Bat	63	62	63	59	61	64	100

Scale (%)
100
85
70
55
40
25
10
0

```
                             10        20        30        40        50        60
Consensus                                  MQMQAQAPVVVVTQPGVG--PAPQXSNWQTGMCDCFSDCGVCLCGTFCFPCLGCQVAADM   58
Human (NP_057703.1)                        --................PG....N...................................   58
Rhesus Monkey (NP_001245092.1)             --..S.............L--...N.....S.............................   56
Chimpanzee (XP_001136484.1)                --................PG....N...................................   58
Mouse (NP_631937.1)                        ---M....T.I.....FV--R...N.....SL.................T..........   55
Pig (XP_003129400.1)                       --........I.....G.--SI..T......L.............I...S..AV...S.L   56
Pangolin (XP_017530897.1)                  ..APS.S........MS.--SM..T........................L..A.......   58
The Greater Horseshoe Bat (XP_032962468.1) -------MNP......Y.MGGTVMT.D....AF..CG.I.I......FPL..S..I.S..   53

                                                    70        80        90        100       110
Consensus                                  NECCLCGTSVAMRTLYRTRYGIPGSICDDYMVTLCCPHCSLCQIKRDINRRRAMRTF     115
Human (NP_057703.1)                        ...............................A.......T.................     115
Rhesus Monkey (NP_001245092.1)             .......................................................A.     113
Chimpanzee (XP_001136484.1)                ...............................A.......T.................     115
Mouse (NP_631937.1)                        ........T.........................F..V..V..L..........NA.     112
Pig (XP_003129400.1)                       .............................F.......M................H..     113
Pangolin (XP_017530897.1)                  .............................FL..N...LF...............H..     115
The Greater Horseshoe Bat (XP_032962468.1) .......A......K............S.FLMLS.F.Q.V...L....EK.KE.NA.     110
```

Figure S4. Complementation of PLAC8 KO cell with PLAC8 from eight different species. A) PLAC8 amino acid sequence identity between species (human, rhesus monkey, chimpanzee, mouse, pig, pangolin and bat). B) PLAC8 sequence alignment for all the above species. C) Complementation assay of PLAC8 KO Huh7.5 cells on SADS-CoV-nLuc infection after transient transfection of PLAC8 (not codon-optimized or tagged) from different species. D) Western blot image of PLAC8 from different species after transient transfection using mouse anti-hPLAC8 Ab. E) mRNA expression of PLAC8 of different species after transient transfection. F) Localization of hPLAC8-FLAG, codon-optimized-pigPLAC8-FLAG and codon-optimized-pangoPLAC8-FLAG from stable cell line using mouse anti-FLAG Ab.

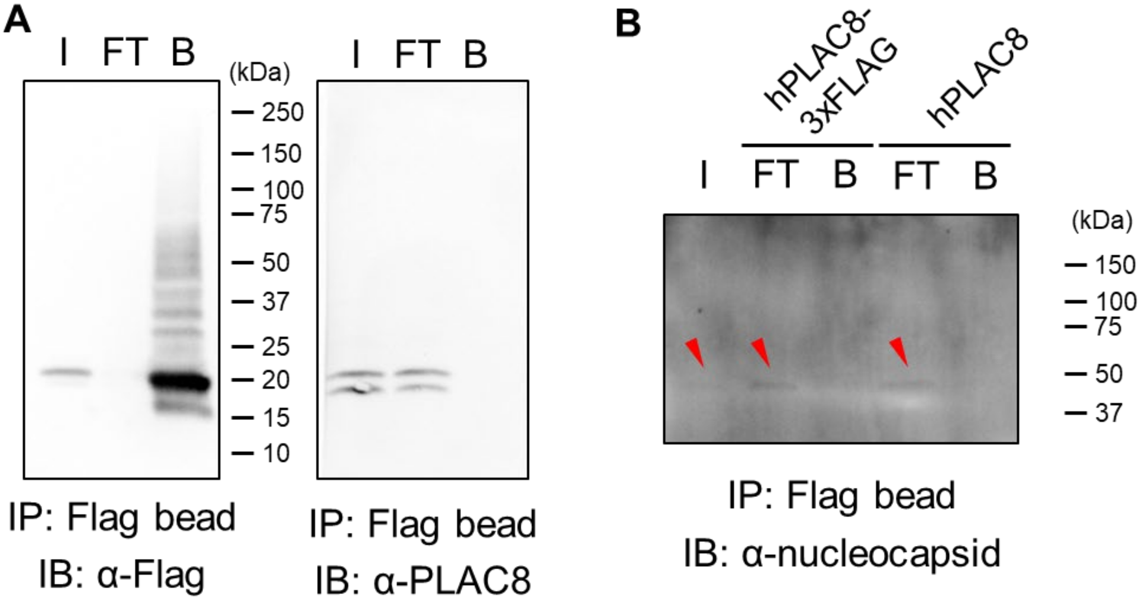


Figure S5. Immunoprecipitation of PLAC8 with intact SADS-CoV viral particles. A) Western blot of hPLAC8-Flag blotted with anti-FLAG antibody and pulled down by magnetic anti-FLAG beads. B) Following pull down, the presence of SADS-CoV was blotted with a mouse SADS-CoV N protein antiserum. I:input, FT: flow-through, B: Bound. Red arrows represent the expected bands of SADS-CoV N proteins.

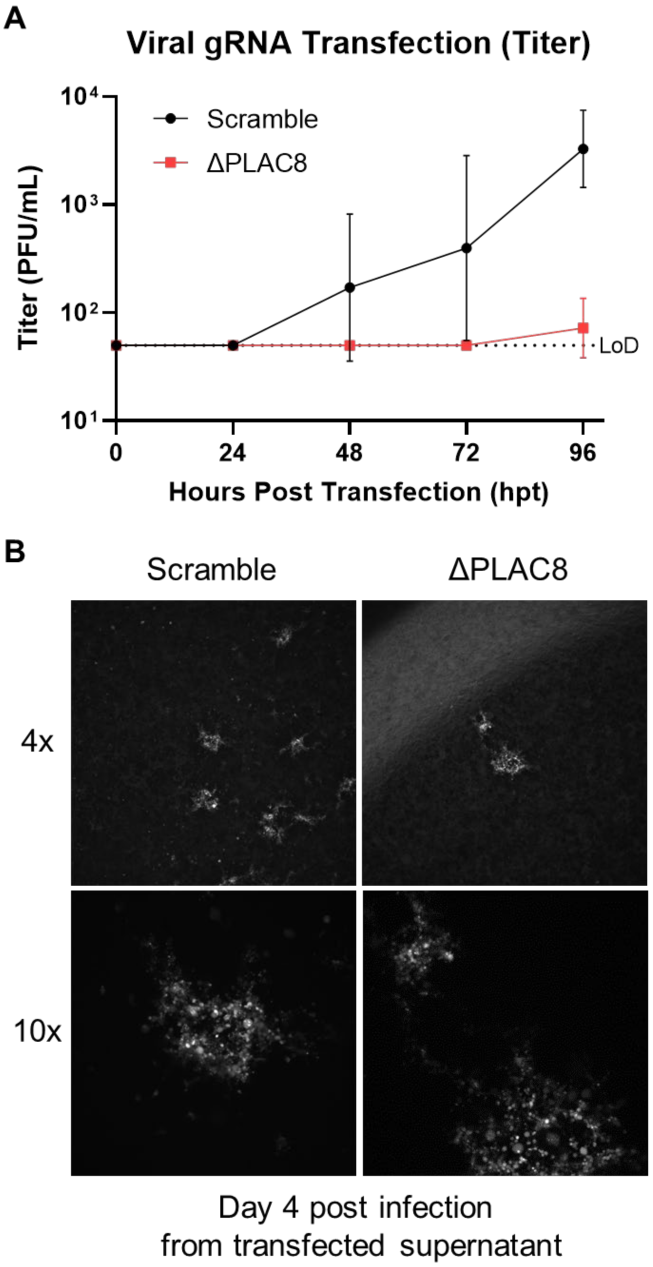


Figure S6. Progeny virus propagation after transfection of genomic RNA isolated from SADS-CoV-nLuc and SADS-CoV-RFP in Scramble and PLAC8 KO Huh7.5 cells. A) Viral titer of supernatant harvested from cells transfected with SADS-CoV-nLuc genomic RNA. B) Representative images of RFP signal in Scramble cells infected with the supernatant of Scramble and PLAC8 KO cells transfected with gRNA of SADS-CoV-RFP. Images were captured at 96 hpt with a 4x and 10x objectives, using RFP expression as a surrogate for viral infection.

Table Legends

Rank	Gene	Protein	Cellular expression	Tissue Distribution	Function
1*	PLAC8	Placenta Specific Gene 8 Protein	Extracellular, Lysosome	Stomach, Small Intestine, Colon, Appendix, Lung	
2*	SCAP	SREBP Cleavage Activating Protein	PM, ER, Golgi	Ubiquitous, Testis	Cholesterol binding, ER to Golgi transport
3	TDO2	Tryptophan 2,3-Dioxygenase	Cytosol	Liver, Appendix	
4*	OR4D11	Olfactory Receptor 4D11	PM		GPCR (Signal Transduction)
5*	SLC26A3	Solute Carrier Family 26 Member 3	PM	Colon, Duodenum, Small Intestine	Chloride Anion Exchanger
6	PSG8	Pregnancy Specific Beta-1-Glycoprotein 8	Extracellular	Placenta	
7	ACP2	Lysosomal Acid Phosphatase	Lysosome	Ubiquitous, Duodenum	
8	C2orf83	Folate Transporter-Like Protein C2orf83	Extracellular	Placenta, Testis	Paralog of SLC19A3
9	WDR86	WD Repeat Domain 86		Ubiquitous	
10*	SLC2A11	Solute Carrier Family 2 Member 11	PM, Nucleus	Kidney	Glucose Transporter

Table S1. Top 10 enriched guide RNAs from SADS-CoV CRISPR screen, their cellular expression, tissue distribution, and a brief description. Asterisks represent targets selected for further work in this study.

	Wang et al.			**Schneider et al.**					**Daniloski et al.**		**Wei et al.**	
	Huh7.5			**Huh7.5**					**A549-hACE2**		**Vero-E6**	
	SARS2	229E	OC43	SARS2 (37ºC)	SARS2 (33ºC)	OC43	NL63	229E	SARS2 (moi:1)	SARS2 (moi:3)	SARS2	SARS2
PLAC8	>1000	>1000	>1000	NS	NS	NS	39 (3.21)	NS	>1000	>1000	0.177 (9639)	0.913 (2522)
SCAP	2	>1000	117	12 (5.85)	49 (3.99)	45 (4.41)	NS	NS	>1000	>1000	-2.281	-1.070
TDO2	>1000	>1000	>1000	NS	NS	NS	NS	NS	>1000	>1000	0.806 (3992)	-0.171
OR4D11	>1000	>1000	>1000	NS	NS	NS	NS	NS	>1000	>1000	n/a	
SLC26A3	>1000	>1000	773	NS	NS	NS	NS	NS	>1000	>1000	-0.055	1.227 (1148)
PSG8	497	>1000	>1000	NS	NS	NS	NS	NS	>1000	>1000	n/a	
ACP2	>1000	>1000	876	NS	NS	NS	NS	NS	>1000	>1000	0.311 (8313)	-1.244
C2orf83	>1000	>1000	>1000	NS	NS	NS	NS	NS	>1000	>1000	n/a	
WDR86	>1000	>1000	>1000	NS	NS	NS	NS	NS	>1000	>1000	-0.025	-0.244
SLC2A11	>1000	>1000	>1000	NS	NS	NS	NS	NS	>1000	>1000	0.065 (10740)	0.260 (8421)
	Rank			Rank (Z-score), NS - not significant (FDR > 0.05)					Rank		Z-score (Rank)	

Table S2. Comparison of the top 10 enriched guide RNAs of this study to multiple CRISPR screen on SARS-CoV-2 and other CoVs. Green highlighted cells represent overlaps between studies.

Supporting Scripts

Truncateguides.pl

```
#this program parses through your reads taking only those with caccg
#it then truncates the sequence to what is in between caccg and gttta
#it outputs these truncated guides in an out file
#it tells you how many reads it put in the out file in the command line
use strict; use warnings;
die "please specify an input .fastq file and an output .txt file" unless @ARGV == 2;
open(IN, "<$ARGV[0]") or die "can't open file $ARGV[0]";
open(OUT,">$ARGV[1]") or die "can't open file $ARGV[1]";
my @lines = ();
my $line = 0;
while (<IN>) {
push (@lines, $_);
}
#print "The amount of lines is " . @lines . " \n ";
my $readcount = 0;
my $motif = "CACCG";
my @guides = ();
my @headers = ();
my @plus = ();
my @quality = ();
foreach (@lines) {$line = $_;
if ( $line =~ /$motif/) {$readcount = ($readcount+1);}
if ( $line =~ /$motif/) {push (@guides, $line) ;}
elsif ( $line =~ m/^\@/) {push (@headers, $line) ;}
elsif ( $line =~ m/^\+/) {push (@plus, $line) ;}
else {push (@quality, $line) ;}
}
my @guides1 = ();
my @guides2 = ();
my $guides = 0;
foreach (@guides) {$guides = $_; @guides1 = split ("CACCG", $guides);
push @guides2, $guides1[1];}
my @guides3 = ();
my $guides2 = 0;
my @guides4 = ();
my $numberguides2 = scalar(@guides2);
my $number = $numberguides2 - 1;
for (my $i = 0; $i < $number; $i++) {if($guides2[$i] =~ /GTTTTA/)
{push (@guides3, $guides2[$i])}};
my @guides5 = ();
foreach (@guides3) {my $guides3 = $_; @guides4 = split ("GTTTTA",
$guides3); {push
@guides5,
$guides4[0]}};
#there may be an issue if there is no gtttta, may get some false reads
print "The number of reads is " . $readcount . "\n";
my $joinedguides = join (" \n", @guides5);
my $numberguides5 = scalar(@guides5);
my $number2 = $numberguides5 - 1;
for (my $i = 0; $i < $number2; $i++) {
print OUT ($headers[$i]);
print OUT ($guides5[$i] . "\n");
print OUT ($plus[$i]);
print OUT ($quality[$i]);
}
close IN or die "Cannot close input";
close OUT or die "Cannot close output \n";
# print join(',', @lines); # see if it worked
```

Ggplot_mageck_genefunction.r

```
library("ggplot2")
library("scale")
```

```
data<- read.csv("SADS.gene_summary.txt", header=TRUE, sep="\t", stringsAsFactors = FALSE)
data$Rand <- sample(5:500, replace=T, nrow(data))
data$log.p<- -1 * log10(data$pos.p.value)
top10<- data[1:10,]
scale=c(0,1,2,3,4,5,6,7)
label=c(0,1,2,3,4,5,6,7)
windowsFonts(A=windowsFont("Arial"))
library(viridis)
tiff("SADS gene functions.jpg", units="in", width=12, height=12, res=300)
qplot(x=Rand, y=log.p, data=data, color=Function, size=pos.lfc, shape=TM, alpha=0.5, geom="point",
xlab="", ylab="-log(p-value)") +
scale_radius(range=c(0.001,20)) +
scale_y_continuous(limits=c(0,7), breaks=scale, labels=label) +
theme(panel.grid.major=element_blank(), panel.grid.minor=element_blank(),
axis.text.x = element_blank(), axis.ticks.x = element_blank(),
axis.text.y=element_text(family="A", color="black"),
axis.title.y=element_text(family="A", color="black"), legend.position="none",
panel.border=element_rect(colour="black", fill=NA,
size=1),panel.background=element_rect(fill="grey98")) +
annotate("text", x=as.numeric(top10$Rand), y=as.numeric(top10$log.p), family="A", label=top10$id ) +
scale_color_manual(values =
c("deeppink2","darkorange1","black","gold","red2","navy","darkorchid4")) +
scale_shape_manual(values=c(16,18))
dev.off()
DESeq-heatmap.r
library(DESeq2)
counts <- read.table(file="counts_S1v1b2.txt",sep="\t",header=T)
rownames(counts) <- make.names(counts[,1], unique = TRUE)
meta <- read.table(file="meta_S1v1b2.txt",sep="\t",header=T,row.names=1)
#read in data
counts.ann <- counts[,2:6]
counts.num <- counts[,-1*1:6]
counts.num[is.na(counts.num)] <- 0
counts.num[ , 1:6] <- apply(counts.num[ , 1:6], 2,function(x) as.integer(x))
#separate into annotation columns and count columns
temp <- dimnames(counts.num)
#creates variable holding the column and row names of counts.num
counts.num<- apply(counts.num,2,function(counts.num){as.numeric(as.vector(counts.num))})
#converts numbers from factor variable to numeric
rc<- round( colSums(counts.num) / 1e6, 1 )
#takes the sum of the columns, divide by 10^6 for millions of reads
dimnames(counts.num)<-temp
#puts row and column names back on counts.num
counts.num<- counts.num[,rownames(meta)]
#ensures that columns of counts.num and rows of meta are in the same order
meds<- apply(counts.num,2,function(counts.num){quantile(counts.num[counts.num>0],0.75)})
#takes the top quartile of all reads above 0
normFactor<- median(meds)/meds
#creates normalization factor from the quartile variable
counts.norm<- t(apply(counts.num,1,function(x,y){y*x},normFactor))
#multiply the reads by normalization factor
means<- apply(counts.num,1,mean)
sds<- apply(counts.num,1,sd)
#mean and standard deviation of columns of counts.num
means.n<- apply(counts.norm,1,mean)
sds.n<- apply(counts.norm,1,sd)
#same metrics as above, with normalized data
counts.ns<- t(scale(t(log10(counts.norm+1)),scale=F))
#scale centers the means; only works on columns, so transform; =F means won't scale variance
#function creating a list of colors for heatmap
cols <- function(lowi = "yellow", highi = "blue", ncolors = 20) {
low <- col2rgb(lowi)/255
high <- col2rgb("black")/255
col1 <- rgb(seq(low[1], high[1], len = ncolors), seq(low[2],
high[2], len = ncolors), seq(low[3], high[3], len = ncolors))
low <- col2rgb("black")/255
high <- col2rgb(highi)/255
col2 <- rgb(seq(low[1], high[1], len = ncolors), seq(low[2],
high[2], len = ncolors), seq(low[3], high[3], len = ncolors))
col<-c(col1[1:(ncolors-1)],col2)
return(col)
}
#supervised analysis
filt<- rowSums(counts.num)>1
counts.num<- counts.num[filt,]
counts.ns<- counts.ns[filt,]
counts.norm<- counts.norm[filt,]
counts.ann<- counts.ann[filt,]
chrs<- chrs[filt]
#running tests for differential expression
des.Variable<- DESeqDataSetFromMatrix(countData = counts.num, colData = meta, design = ~ Variable)
des.Variable<- DESeq(des.Variable)
results<- results(des.Variable, cooksCutoff=FALSE)
sum(results$padj< 0.05, na.rm=T)
#how many are significant
gfilt<- results$padj<0.05 & !is.na(results$padj)
#filter to pull out significant genes
rownames(counts.num) [gfilt]
#puts out genes that are significant
sum(rownames(results)==rownames(counts.ns))
#checks to make sure you have the same row names, in the same order
library(heatmap3)
counts.norm_sig <- counts.norm[gfilt,]
tiff("S1v1b2_heatmap.tiff", units="in", width=8, height=8, res=300)
heatmap3(counts.norm_sig,col=cols(),labRow=FALSE)
dev.off()
```

```
write.table(results,"S1v1b2_results.txt",sep="\t",col.names=NA)
write.table(results[gfilt,],"S1v1b2_sig_results.txt",sep="\t",col.names=NA)
```

Volcanoplot.r

```
library(ggplot2)
library(ggrepel)
results <- read.table(file="S1v1b2_results.txt",sep="\t",header=T)
results$diffexpressed <- "NO"
results$diffexpressed[results$log2FoldChange > 0.6 & results$pvalue < 0.05] <- "UP"
results$diffexpressed[results$log2FoldChange < -0.6 & results$pvalue < 0.05] <- "DOWN"
results <- results[order(results$pvalue),]
results$delabel <- NA
results$delabel[results$diffexpressed != "NO"] <- results$X[results$diffexpressed !="NO"]
up <- results[results$diffexpressed == "UP",]
down <- results[results$diffexpressed == "DOWN",]
topup <- up[1:3,]
topdown <- down[1:8,]
top <- rbind(topup,topdown)
windowsFonts(A=windowsFont("Arial"))
tiff("S1v1b2_volcano.tiff", units="in", width=8, height=8, res=300)
ggplot(data=results, aes(x=log2FoldChange, y=-log10(pvalue), col=diffexpressed)) + geom_point() +
#geom_text_repel(data=top, aes(label=delabel)) +
theme(panel.grid.major=element_blank(), panel.grid.minor=element_blank(),
axis.text.x=element_text(family="A", color="black"),
axis.text.y=element_text(family="A", color="black"), axis.title.x=element_text(family="A",
color="black"),
axis.title.y=element_text(family="A", color="black"), legend.position="none",
panel.border=element_rect(colour="black", fill=NA,
size=1),panel.background=element_rect(fill="grey98")) +
geom_vline(xintercept=c(-0.6,0.6), col="gray") + geom_hline(yintercept=-log10(0.05),col="gray") +
xlim(-8,8) + ylim(0,30) + scale_color_manual(values=c("blue", "black","red"))
dev.off()
```