Name: Holin Chen

# Comparing the GC Contents and CpG Deficiencies among serval coronaviruses: SAR-CoV, SAR-CoV-2, MERS-CoV and 2 two bat-derived severe acute respiratory syndrome (SARS)-like coronaviruses (bat-SL-CoVZC45 and bat-SL-CoVZXC21)

## Introduction

Coronaviruses (CoV) usually exhibit strong CpG deficiency on their genome. This coronavirus pattern of lacking CG dinucleotides can be explained by evasion of the immune response by the zinc finger antiviral protein (ZAP), an important component in mammalian interferon. ZAP shows CpG-specific antivirus activities by binding to the CG dinucleotides in viral RNA genomes to inhibit viral replication and meditate viral genome degradation. To escape this CpG-specific antivirus activities by ZAP, many RNA viruses, including coronavirus, have naturally evolved to decrease their CpG occurrence to better fit with the host's immune system [1].

Therefore, the magnitude of CpG deficiencies in coronavirus can be served as an index of how potentially effective the coronavirus can escape from human's immune system. As the current pandemic of COVID-19, lots of pathogenic studies related to COVID-19 and genomic research for SARS-CoV-2 have been conducted. In this study, I extracted the virus sequences of sever acute respiratory syndrome coronavirus (SARS-CoV), sever acute respiratory syndrome coronavirus 2 (SARS-CoV-2), middle east respiratory syndrome coronavirus (MERS-CoV) and two bat-derived severe acute respiratory syndrome-like coronaviruses (bat-SL-CoVZC45 and bat-SL-CoVZXC21) to compare their GC contents and CpG deficiencies to see which coronavirus has the most potentially effective mechanism to escape the anti-virus activities by ZAP.

## Methods

The fasta files of whole-genome virus sequences of SARS-CoV (accession number: NC_004718.3), SARS-CoV-2 (accession number: NC_045512.2) and MERS-CoV (accession number: NC_ 038294.1) were extracted from GenBank. The sequences of the two bat-derived severe acute respiratory syndrome-like coronaviruses, bat-SL-CoVZC45 (accession number: MG772933) and bat-SL-CoVZXC21 (accession number: MG772934) were extracted from China National Microbiological Data Center website[2].

Name: Holin Chen

By using the Biostrings package, I calculated the occurrences and percentages of C, G and CG dinucleotides of each coronavirus whole-genome sequence. Then I split each sequence into 29 to 30 1000 bp windows based on their sequence length. I calculated both the GC content (pC + pG, where pC and pG are the percentages of bases being G or C) and the CpG deficiency (observed-to-expected CG ratio: calculated by pCG/(pC*pG), pCG is the percent of CG dinucleotides occurrence) on each 1000 bp window for each coronavirus sequence, and generated the density plots and scatter plot of GC contents and CpG deficiency for all these 5 virus sequences to see if there is any heterogeneity of CpG deficiency among these coronaviruses.

**Results**

| Virus | Total length | A | C | G | T | GC Content | I_CpG |
|---|---|---|---|---|---|---|---|
| Bat-SL-CoVZC45 | 29802 | 8740 | 5574 | 6020 | 9468 | 0.3890 | 0.4476 |
| bat-SL-CoVZXC21 | 29732 | 8734 | 5567 | 5976 | 9455 | 0.3882 | 0.4495 |
| SARS-CoV | 29751 | 8481 | 5940 | 6187 | 9143 | 0.4076 | 0.4598 |
| MERS-CoV | 30111 | 7897 | 6096 | 6303 | 9815 | 0.4118 | 0.5595 |
| SARS-CoV-2 | 29903 | 8954 | 5492 | 5863 | 9594 | 0.3797 | 0.4077 |

Table 1. the nucleotide base counts, GC contents and CpG deficiencies in the whole genomes for the five coronaviruses. I_CpG denoted as the conventional index of CpG deficiency (the index is expected to be 1; < 1 if deficiency)

From table 1, we can see that the lengths of genome sequences of these 5 coronaviruses are really close, and their nucleotide occurrences are also similar. All the 5 coronaviruses have strong CpG deficiencies, ranging from 0.40 to 0.56. Within the 5 coronaviruses, SAR-CoV-2 has the lowest GC content and most CpG deficiency.

Name: Holin Chen

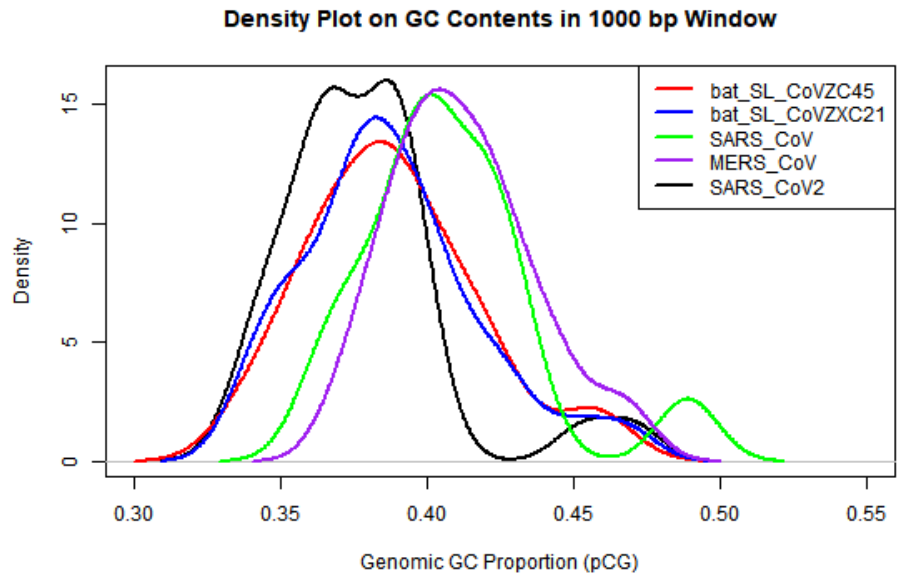


Figure 1. the density plot on GC contents in 1000 bp window for the five coronaviruses.

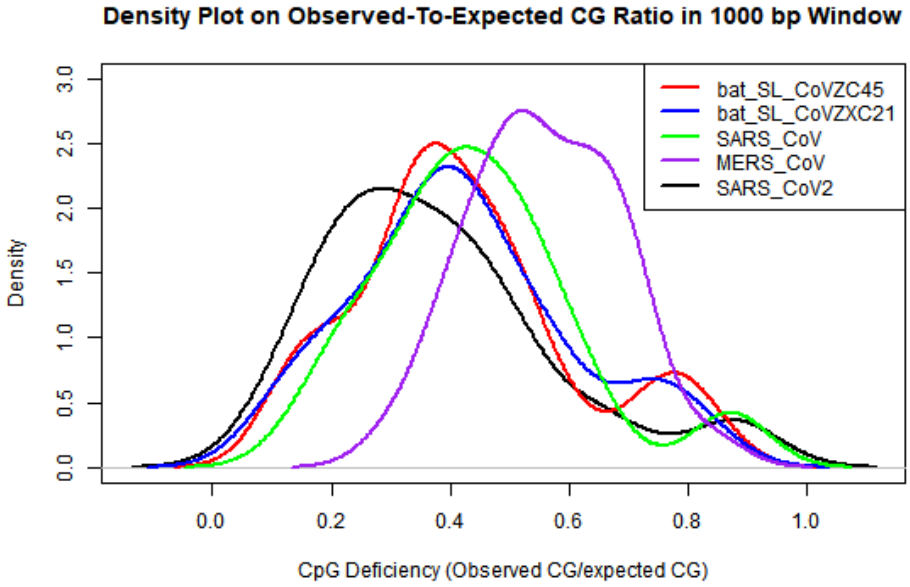


Figure 2. the density plot on CpG deficiencies/observed-to-expected CG ratio in 1000 bp window for the five coronaviruses

Figure 1 and 2 tell us more details about the distributions of GC contents and CpG deficiencies along the genomes of these 5 coronaviruses. Both the curves of GC contents and CpG deficiencies for bat-SL-CoVZC45 and bat-SL-CoVZXC21 almost

overlap, indicating these 2 bat derived coronaviruses have strong similarity. The curves of GC contents and CpG deficiency for SAR-CoV-2 reach peaks the most quickly than other 4 coronaviruses, meaning the CpG deficiency for SAR-CoV-2 is the most apparent among these 5 coronaviruses. The curve of CpG deficiency for MERS-CoV reaches peak at last in high density, indicating the CpG deficiency for MERS-CoV is relatively moderate comparing to other coronaviruses.

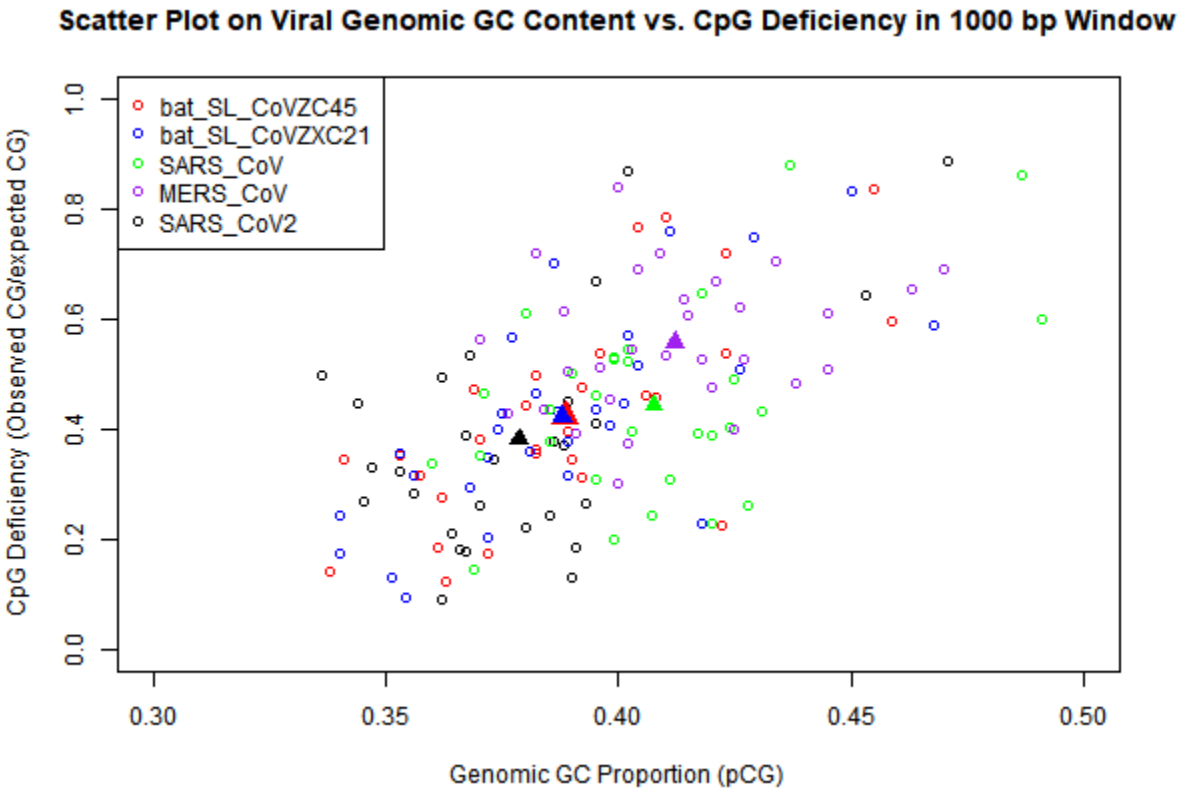


Figure 3. the scatter plot of GC content vs. CpG deficiency in 1000 bp window for the five coronaviruses. The filled triangles locate at the means of GC content and the means of CpG deficiency among all 1000 bp windows for each coronavirus.

Figure 3 shows that most scatters of SAR-CoV-2 with its mean point (pCG = 0.3786, CpG deficiency=0.3803) locate at the bottom left area of the plot, proving again that SAR-CoV-2 has the lowest CG content and strongest CpG deficiency among the five coronaviruses. The mean CpG deficiencies among 1000 bp windows for SAR-CoV, MERS-CoV, bat-SL-CoVZC45 and bat-SL-CoVZXC21 are 0.4432, 0.5577, 0.4224, 0.4224 separately, which are all larger than that in SAR-CoV-2 of 0.3803.

Name: Holin Chen

This study finds out that SAR-CoV-2 has the strongest CpG deficiency compared to SAR-CoV, MERS-CoV, bat-SL-CoVZC45 and bat-SL-CoVZXC21. This phenomenon may lead SAR-CoV-2 have more mature mechanism to escape from the CpG-specific antivirus activities by ZAP in human immune system than other common coronaviruses like SAR-CoV and MERS-CoV.

## Reference

[1] Xuhua Xia, Extreme Genomic CpG Deficiency in SARS-CoV-2 and Evasion of Host Antiviral Defense, *Molecular Biology and Evolution*, Volume 37, Issue 9, September 2020, Pages 2699–2705, https://doi.org/10.1093/molbev/msaa094

[2] Lu, Roujian, et al. "Genomic Characterisation and Epidemiology of 2019 Novel Coronavirus: Implications for Virus Origins and Receptor Binding." *Lancet (London, England),* vol. 395, no. 10224, 2020, pp. 565-574.

Name: Holin Chen

# Appendix Code

```
library(Biostrings)

filelist <- list.files(pattern = "*.fasta")

filelist2 <- filelist[-grep("NMDC",filelist)]
fa_seq2 <- lapply(filelist2, readDNAStringSet)
fa_seq2 <- do.call(c, fa_seq2)

fa_seq2@ranges@NAMES[3] <- "NC_004718.3_SARS-CoV"
fa_seq2@ranges@NAMES[4] <- "NC_038294.1_MERS-CoV"
fa_seq2@ranges@NAMES[5] <- "NC_045512.2_SARS-CoV-2"

## get GC content
bases=alphabetFrequency(fa_seq2,baseOnly=TRUE)
bases[1:5,1:4]

##         A    C    G    T
## [1,] 8740 5574 6020 9468
## [2,] 8734 5567 5976 9455
## [3,] 8481 5940 6187 9143
## [4,] 7897 6096 6303 9815
## [5,] 8954 5492 5863 9594

ntotBases=apply(bases[1:5,1:4], 1, sum)
baseFreq=bases[1:5,1:4]/ntotBases
GCcontent=baseFreq[,"C"]+baseFreq[,"G"]
ATcontent=baseFreq[,"A"]+baseFreq[,"T"]

## look at CG dinucleotide content
cg=vmatchPattern("CG", fa_seq2)
ncg=lengths(cg)
## compute the observed to expected ratio
ncg/(baseFreq[,"C"]*baseFreq[,"G"]*ntotBases) ## this shows CG rarely stay to
gether.

## [1] 0.4476231 0.4495317 0.4598152 0.5595400 0.4076889

## compare to the observed to expected ratio of TG
tg=vmatchPattern("TG", fa_seq2)
ntg=lengths(tg)
ntg/(baseFreq[,"T"]*baseFreq[,"G"]*ntotBases) ## this shows TG presented more
 than expected.

## [1] 1.358930 1.356021 1.401093 1.315143 1.376345

######### look at GC content and CG dinucleotide distribution in 1000 bp wind
ows in whole genome.

ss_1=seq(1, lengths(fa_seq2)[1], by=1000)
```

```
ss_1=ss_1[-length(ss_1)] ## remove the last one
bat_SL_CoVZC45=DNAStringSet(fa_seq2$MG772933, start=ss_1, end=ss_1+999)
ff_1=alphabetFrequency(bat_SL_CoVZC45, baseOnly=TRUE)
pCG_1=(ff_1[,"C"]+ff_1[,"G"])/rowSums(ff_1)
hist(pCG_1[pCG_1>0],100, main = "GC content in 1000 bp window")

## CG occurance
nCG_1=vcountPattern("CG", bat_SL_CoVZC45)
obsExp_1=nCG_1*1000/(ff_1[,"C"]*ff_1[,"G"])
mean(obsExp_1,na.rm=TRUE)

## [1] 0.4224042

hist(obsExp_1,100,main = "observed-to-expected CG ratio in 1000 bp window") #
# see a long tail, those are CpG islands
abline(v=mean(obsExp_1,na.rm=TRUE),col="red", lwd=1, lty=2)

#########################################################################

ss_2=seq(1, lengths(fa_seq2)[2], by=1000)
ss_2=ss_2[-length(ss_2)] ## remove the last one
bat_SL_CoVZXC21=DNAStringSet(fa_seq2$MG772934, start=ss_2, end=ss_2+999)
ff_2=alphabetFrequency(bat_SL_CoVZXC21, baseOnly=TRUE)
pCG_2=(ff_2[,"C"]+ff_2[,"G"])/rowSums(ff_2)
hist(pCG_2[pCG_2>0],100, main = "GC content in 1000 bp window")

## CG occurance
nCG_2=vcountPattern("CG", bat_SL_CoVZXC21)
obsExp_2=nCG_2*1000/(ff_2[,"C"]*ff_2[,"G"])
mean(obsExp_2,na.rm=TRUE)

## [1] 0.4223516

hist(obsExp_2,100,main = "observed-to-expected CG ratio in 1000 bp window") #
# see a long tail, those are CpG islands
abline(v=mean(obsExp_2,na.rm=TRUE),col="red", lwd=1, lty=2)

#########################################################################

ss_3=seq(1, lengths(fa_seq2)[3], by=1000)
ss_3=ss_3[-length(ss_3)] ## remove the last one
SARS_CoV=DNAStringSet(fa_seq2$`NC_004718.3_SARS-CoV`, start=ss_3, end=ss_3+99
9)
ff_3=alphabetFrequency(SARS_CoV, baseOnly=TRUE)
pCG_3=(ff_3[,"C"]+ff_3[,"G"])/rowSums(ff_3)
hist(pCG_3[pCG_3>0],100, main = "GC content in 1000 bp window")

## CG occurance
nCG_3=vcountPattern("CG", SARS_CoV)
obsExp_3=nCG_3*1000/(ff_3[,"C"]*ff_3[,"G"])
mean(obsExp_3,na.rm=TRUE)
```

```
## [1] 0.4431991

hist(obsExp_3,100,main = "observed-to-expected CG ratio in 1000 bp window") #
# see a long tail, those are CpG islands
abline(v=mean(obsExp_3,na.rm=TRUE),col="red", lwd=1, lty=2)

#########################################################################

ss_4=seq(1, lengths(fa_seq2)[4], by=1000)
ss_4=ss_4[-length(ss_4)] ## remove the last one
MERS_CoV=DNAStringSet(fa_seq2$`NC_038294.1_MERS-CoV`, start=ss_4, end=ss_4+99
9)
ff_4=alphabetFrequency(MERS_CoV, baseOnly=TRUE)
pCG_4=(ff_4[,"C"]+ff_4[,"G"])/rowSums(ff_4)
hist(pCG_4[pCG_4>0],100, main = "GC content in 1000 bp window")

## CG occurance
nCG_4=vcountPattern("CG", MERS_CoV)
obsExp_4=nCG_4*1000/(ff_4[,"C"]*ff_4[,"G"])
mean(obsExp_4,na.rm=TRUE)

## [1] 0.5577203

hist(obsExp_4,100,main = "observed-to-expected CG ratio in 1000 bp window") #
# see a long tail, those are CpG islands
abline(v=mean(obsExp_4,na.rm=TRUE),col="red", lwd=1, lty=2)

#########################################################################

ss_5=seq(1, lengths(fa_seq2)[5], by=1000)
ss_5=ss_5[-length(ss_5)] ## remove the last one
SARS_CoV2=DNAStringSet(fa_seq2$`NC_045512.2_SARS-CoV-2`, start=ss_5, end=ss_5
+999)
ff_5=alphabetFrequency(SARS_CoV2, baseOnly=TRUE)
pCG_5=(ff_5[,"C"]+ff_5[,"G"])/rowSums(ff_5)
hist(pCG_5[pCG_5>0],100, main = "GC content in 1000 bp window")

## CG occurance
nCG_5=vcountPattern("CG", SARS_CoV2)
obsExp_5=nCG_5*1000/(ff_5[,"C"]*ff_5[,"G"])
mean(obsExp_5,na.rm=TRUE)

## [1] 0.3803307

hist(obsExp_5,100,main = "observed-to-expected CG ratio in 1000 bp window") #
# see a long tail, those are CpG islands
abline(v=mean(obsExp_5,na.rm=TRUE),col="red", lwd=1, lty=2)

#########################################################################

## compare with genome wide distribution of GC content
d1=density(pCG_1)
```

```
d2=density(pCG_2)
d3=density(pCG_3)
d4=density(pCG_4)
d5=density(pCG_5)
plot(d5, lwd=2, main="Density Plot on GC Content in 1000 bp Window",
     xlab="Genomic GC Proportion (pCG)",xlim=c(0.3,0.55))
lines(d1, col="red",lwd=2)
lines(d2, col="blue",lwd=2)
lines(d3, col="green",lwd=2)
lines(d4, col="purple",lwd=2)
par(cex = 0.75)
legend("topright", legend=c("bat_SL_CoVZC45", "bat_SL_CoVZXC21","SARS_CoV","M
ERS_CoV","SARS_CoV2"),
       lwd=2, col=c("red","blue","green","purple","black"))

## compare with genome wide distribution of GC content
d1_ICpG=density(obsExp_1)
d2_ICpG=density(obsExp_2)
d3_ICpG=density(obsExp_3)
d4_ICpG=density(obsExp_4)
d5_ICpG=density(obsExp_5)
plot(d5_ICpG,col="black", lwd=2,ylim=c(0,3.0), main="Density Plot on Observed
-To-Expected CG Ratio in 1000 bp Window",
     xlab="CpG Deficiency (Observed CG/expected CG)")
lines(d1_ICpG, col="red",lwd=2)
lines(d2_ICpG, col="blue",lwd=2)
lines(d3_ICpG, col="green",lwd=2)
lines(d4_ICpG, col="purple",lwd=2)
par(cex = 0.75)
legend("topright", legend=c("bat_SL_CoVZC45", "bat_SL_CoVZXC21","SARS_CoV","M
ERS_CoV","SARS_CoV2"),
       lwd=2, col=c("red","blue","green","purple","black"))

#####################################
pCG <- c(mean(pCG_1), mean(pCG_2), mean(pCG_3), mean(pCG_4), mean(pCG_5))
obsExp <- c(mean(obsExp_1), mean(obsExp_2), mean(obsExp_3), mean(obsExp_4), m
ean(obsExp_5))

#########################################
plot(x=pCG, y=obsExp)

plot(x=pCG_1, y=obsExp_1, col="red", main="Scatter Plot on Viral Genomic GC C
ontent vs. CpG Deficiency in 1000 bp Window",
     xlab="Genomic GC Proportion (pCG)", ylab="CpG Deficiency (Observed CG/ex
pected CG)",xlim=c(0.3,0.5),ylim=c(0,1.0))
points(x=pCG_5, y=obsExp_5, col="black")
points(x=pCG_2, y=obsExp_2,col="blue")
points(x=pCG_3, y=obsExp_3,col="green")
points(x=pCG_4, y=obsExp_4,col="purple")
points(x=mean(pCG_1),y=mean(obsExp_1),col="red",pch=17,cex=2)
```

```
points(x=mean(pCG_4),y=mean(obsExp_4),col="purple",pch=17, cex=1.5)
points(x=mean(pCG_2),y=mean(obsExp_2),col="blue",pch=17, cex=1.5)
points(x=mean(pCG_3),y=mean(obsExp_3),col="green",pch=17, cex=1.5)
points(x=mean(pCG_5),y=mean(obsExp_5),col="black",pch=17, cex=1.5)
par(cex = 0.75)
legend("topleft", legend=c("bat_SL_CoVZC45", "bat_SL_CoVZXC21","SARS_CoV","ME
RS_CoV","SARS_CoV2"),
       pch=1, col=c("red","blue","green","purple","black"),bg="transparent")
```