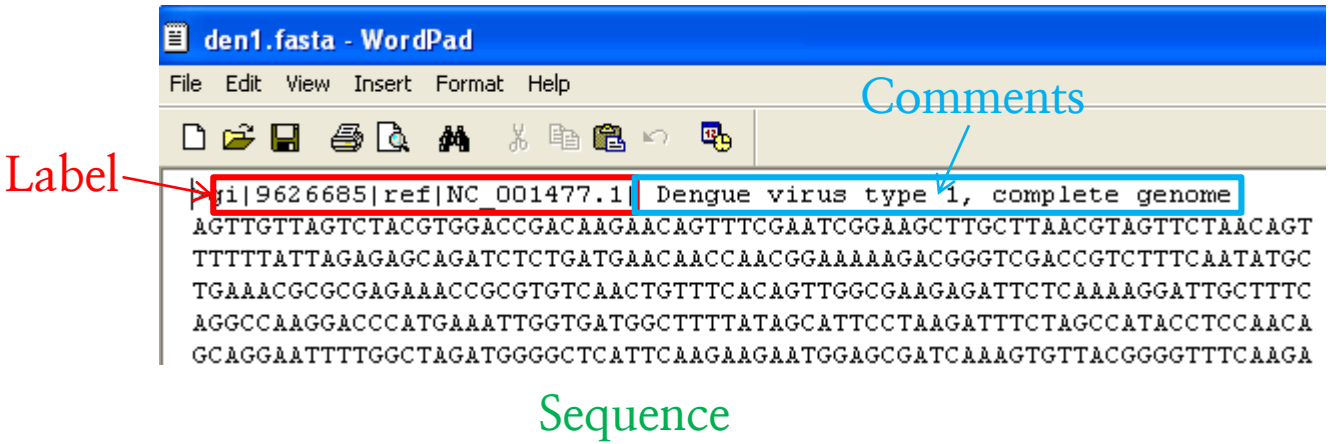
DNA Sequence Statistics (2)

UT Health Science Center 진학예정 /Yunju Yang
2016.08

5.2 Reading Sequence Data with SeqinR

NCBI offers the sequence data in FASTA format!
(i.e. file name – "den1.fasta")




```
> library("seqinr")
> dengue <- read.fasta(file="den1.fasta")
> dengueseq <- dengue[[1]]
> dengueseq[425:535]
```

```
# Load the SeqinR package.
# Read in the file "den1.fasta".
# Put the sequence in a vector called "dengueseq".
# Extract a piece of a sequence
```

5.3 Local Variation in GC Content

```
> GC(dengueseq)                    # Calculate GC proportion
[1] 0.4666977
```

There is probably local variation in GC content within the genome. Local fluctuations in GC content within the genome sequence can provide different interesting information.

Horizontal transfer!?? Biases in mutation !!??

1) Low GC content 를 가지는 종의 DNA 일부가 high GC content를 갖는 종으로 horizontal transfer로 옮겨간다면?? 그 high GC content 종에서 유독 GC content가 낮은 부분이 바로 transfer 된 부분이 될 거다!
2) 유독 Low GC content인 부분이 있다면, 그 부분에서 mutation에서 편향이 발생했다고 볼 수 있다. GC-> AT 로 바뀌는 mutation이 유전체 내의 다른 부분보다 더 자주 일어나겠지!

5.4 A Sliding Window Analysis of GC Content

구간별로 GC content 를 구해보자.

```
> GC(dengueseq[1:2000])
[1] 0.465
> GC(dengueseq[2001:4000])
[1] 0.4525
> GC(dengueseq[4001:6000])
[1] 0.4705
> GC(dengueseq[6001:8000])
[1] 0.479
> GC(dengueseq[8001:10000])
[1] 0.4545
> GC(dengueseq[10001:10735])
[1] 0.4993197
```

```
> starts <- seq(1, length(dengueseq)-2000,
by=2000)
> starts
[1] 1 2001 4001 6001 8001
> n <- length(starts)
> for (i in 1:n) {
    chunk <- dengueseq[starts[i]:(starts[i]+1999)]
    chunkGC <-GC(chunk)
    print (chunkGC)
     }
[1] 0.465
[1] 0.4525
[1] 0.4705
[1] 0.479
[1] 0.4545
```

There seems to be some local variation!!

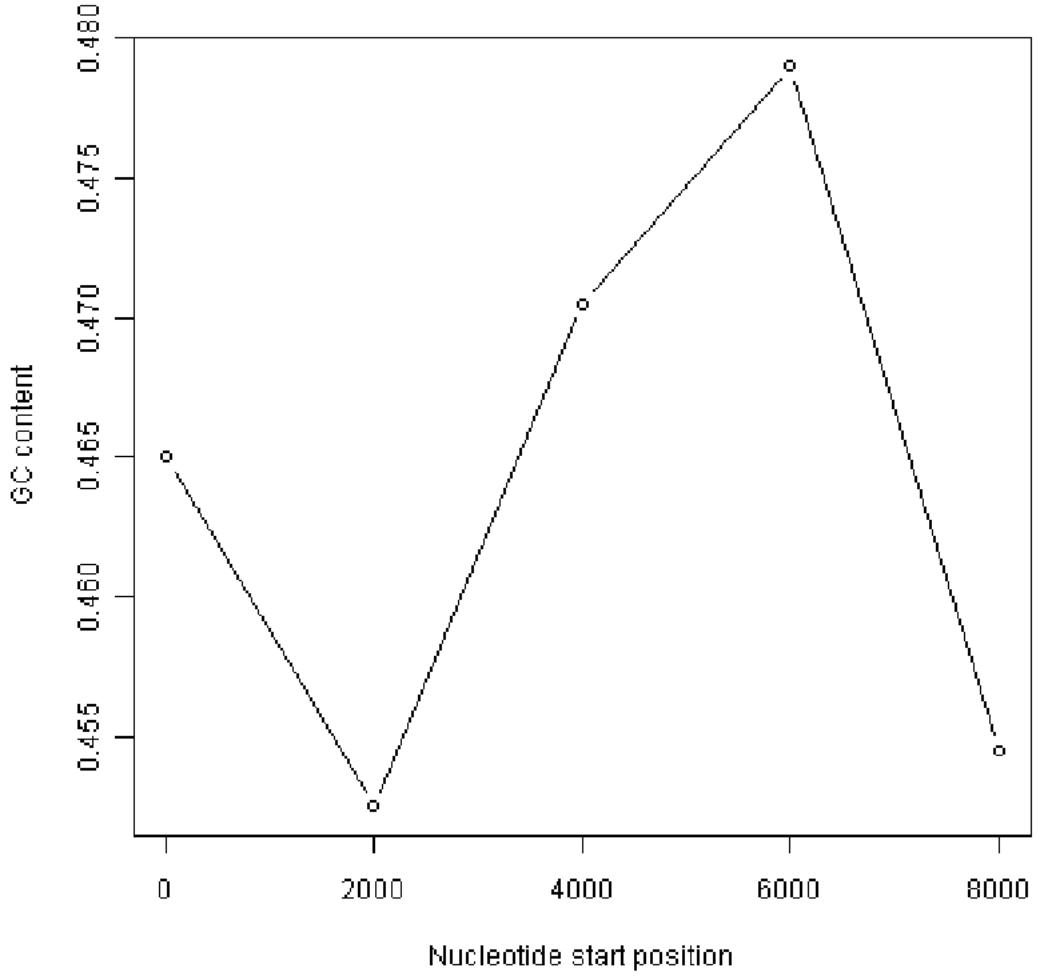
5.5 A Sliding Window Plot of GC Content

구간별로 GC content 를 구해보자.

```
> starts <- seq(1, length(dengueseq)-2000, by=2000)
> starts
[1] 1 2001 4001 6001 8001
> n <- length(starts)
> for (i in 1:n) {
    chunk <- dengueseq[starts[i]:(starts[i]+1999)]
    chunkGC <-GC(chunk)
    print (chunkGC)
    chunkGCs[i]  <- chunkGC
     }
> Plot (starts, chunkGCs, type='b', xlab="Nucleotide start position", ylab="GC content")
```

There seems to be some local variation!!

5.5 A Sliding Window Plot of GC Content



5.5 A Sliding Window Plot of GC Content – Function으로 만들어 자동화하기

```
> slidingwindowplot <- function(windowsize, inputseq)
{
   starts <- seq(1, length(inputseq)-windowsize, by=windowsize)
   n <- length(starts)
   chunkGCs <-numeric(n)
> for (i in 1:n) {
     chunk <- inputseq[starts[i]:(starts[i]+windowsize-1)]
     chunkGC <-GC(chunk)
     print (chunkGC)
     chunkGCs[i]  <- chunkGC
      }
> plot (starts, chunkGCs, type='b', xlab="Nucleotide start position", ylab="GC content")

> slidingwindowplot(windowsize, denguueseq)  짠!
```

5.6 Over-represented and Under-represented DNA Words

> count (dengueseq, 2) ## 특정길이의 sequence가 몇 개나 등장하는지 세기!

$$\rho\,(xy) = fxy/(fx*fy)$$

Over- or Under-represented 인지 판단하는 Statistics!

```
> count(dengueseq, 1) # Get the number of occurrences of 1-nucleotide DNA words
   a    c    g    t
3426 2240 2770 2299
> 2770/(3426+2240+2770+2299) # Get fG
[1] 0.2580345
> 2240/(3426+2240+2770+2299) # Get fC
[1] 0.2086633
> count(dengueseq, 2) # Get the number of occurrences of 2-nucleotide DNA words
  aa  ac  ag  at  ca  cc  cg  ct  ga  gc  gg  gt  ta  tc  tg  tt
1108 720 890 708 901 523 261 555 976 500 787 507 440 497 832 529
> 500/(1108+720+890+708+901+523+261+555+976+500+787+507+440+497+832+529) # Get fGC
[1] 0.04658096
> 0.04658096/(0.2580345*0.2086633) # Get rho(GC)
[1] 0.8651364
```

감사합니다.