Tracking Measles D4

J.A.Hartigan Yale University September 2013

1 Recent Measles History

from
http://en.wikipedia.org/wiki/Measles
http://www.iayork.com/MysteryRays/2009/09/02/measles-deaths-pre-vaccine

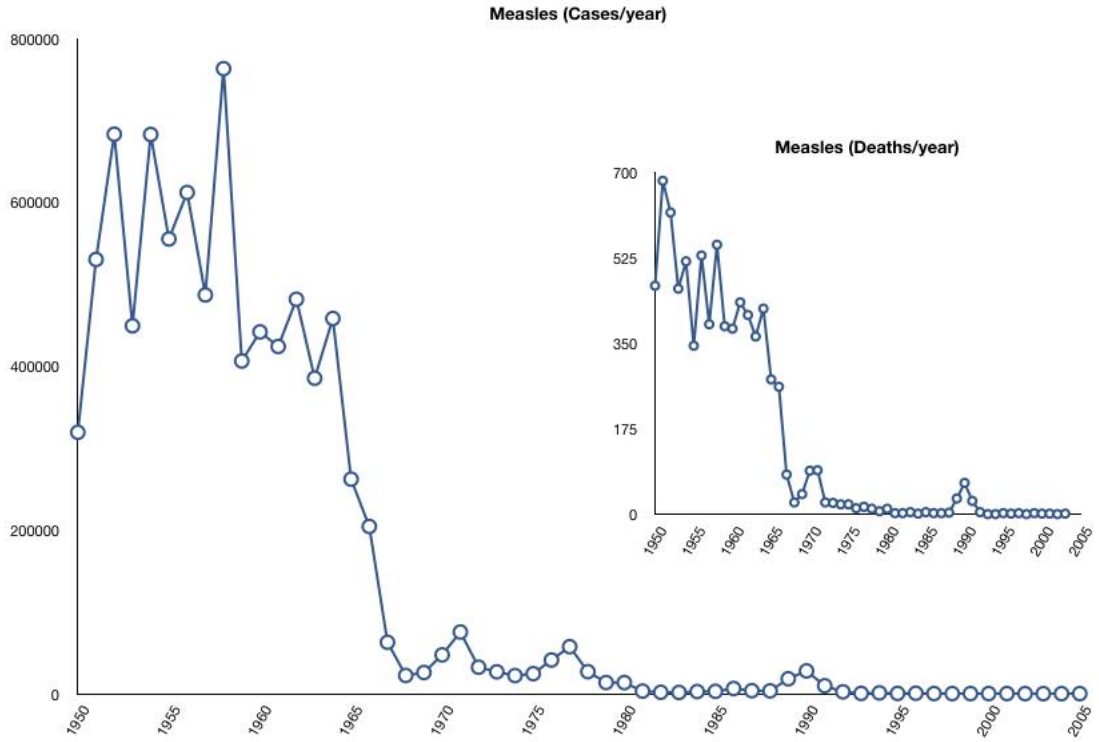
"Measles is an infection of the respiratory system caused by a single-stranded, negative-sense enveloped RNA virus. Symptoms include fever, cough, runny nose, red eyes and a generalized, maculopapular, erythematous rash. Measles is spread through respiration (contact with fluids from an infected person's nose and mouth, either directly or through aerosol transmission), and is highly contagious—90% of people without immunity sharing living space with an infected person will catch it.

The measles virus evolved from the then-widespread rinderpest virus, most probably in the 11th and 12th centuries. The earliest likely origin is within the seventh century: for this earlier origin there is some linguistic evidence. The current epidemic strain evolved at the beginning of the 20th century – most probably between 1908 and 1943. In 1954, the virus causing the disease was isolated from an 11-year old boy from the United States, David Edmonston, and adapted and propagated on chick embryo tissue culture. To date, 21 strains of the measles virus have been identified. While at Merck, Maurice Hilleman developed the first successful vaccine[46]. Licensed vaccines to prevent the disease became available in 1963.

In 2000 the WHO estimated that there were 45 million cases of measles worldwide with 800,000 deaths from it. Mortality in developed countries is ~1/1000. In sub-Saharan Africa, mortality is 10%.
The WHO currently recognises 8 clades of measles (A - H). Subtypes are designed with numerals - A1, D2 etc. Currently a total of 23 subtypes are recognised. The sequencing of the 450 nucleotides that code for the C-terminal 150 amino acids of N are the minimum amount of sequence data required for genotyping a measles virus isolate.
The major genotypes differ between countries and regions: D strains are the most common in Europe, the Middle East, India and Japan; B strains in Africa; and H in China. Type C occur in North Africa and southern Europe. Several types appear to have become extinct - D1, E, F, G1 - not having been isolated in many years. The measles vaccine strains are representatives of genotype A.

Incidence of Measles in United States after Vaccine introduction, 1963:

```
picture("pictures/Measles.jpg")
```



We see the huge drop in incidence of measles and deaths from measles between 1963 and 1968 due to vaccination. Andrew Wakefield, (1998, Lancet) claimed that the vaccine caused autism. Vaccination rates declined in the UK after the claims, which have now been repudiated in several clinical trials.

2. D4 Hamburg

"A new strain of measles virus, D4-Hamburg, was imported from London to Hamburg in December 2008 and subsequently spread to Bulgaria, where an outbreak of 24,300 cases was observed. We analyzed spread of the virus to demonstrate the importance of addressing hard-to-reach communities within the World Health Organization European Region regarding access to medical care and vaccination campaigns. The D4-Hamburg strain appeared during 2009–2011 in Poland, Ireland, Northern Ireland, Austria, Greece, Romania, Turkey, Macedonia, Serbia, Switzerland, and Belgium and was repeatedly reimported to Germany. The strain was present in Europe for 27 months and led to 25,000 cases in 12 countries. Spread of the virus was prevalently but not exclusively associated with travel by persons in the Roma ethnic group; because this travel extends beyond the borders of any European country, measures to prevent the spread of measles should be implemented by the region as a whole."

3. D4 data in in Genbank

Data is retrieved in Genbank format, a text format accessible to scanning in R, and stored in the file " data/MeaslesGenbank".
(To repeat the process, go to the web page indicated, click on “send to”, choose destination “file”, and choose format “Genbank”.)

Selections are made from the Genbank data set in the data preparation section and saved in :
"data/MeasleExtract.csv"

```
MeasleExtract <-
        read.csv("data/MeasleExtract.csv",as.is=T)
head(MeasleExtract, 2)
```

```
X cases       years          city
1 DEFINITION  Measles virus strain MVs/Barcelona.SPA/07.06/2
[D4] nucleoprotein     1 2006.083333 Barcelona.SPA
2      DEFINITION  Measles virus strain Mvi/Sydney.Aus/1.06
nucleoprotein gene,     3 2006.250000    Sydney.Aus

data
1
aaggtcagttccacattggcatctgaactcggtatcactgctgaggatgcaaggcttgt
ttcagagattgcaatgcatactactgaggacaggatcagtagagctgttggacctagac
aagcccaagtgtcatttatacacggtgatcaaagtgagaatgagctaccaggattgggg
ggcaaggaagataggagggtcaaacagagtcggggagaggccagggagagccacagaga
agccgggtccagcagagcaagtgatgcgagagctgcccattttccaaccaacacacccc
tagacattgacactgcatcagagtcgggccaagatccgcaggacagtcgaaggtcagct
gacgccctgctcaggttgcaggccatggcagggatcttggaagaacaaggctcagacac
ggacatccctagggtgtataatgacagagatcttctggactag
2
aaggtcagttccacattggcatctgaactcggtatcactgctgaggatgcaaggcttgt
ttcagagattgcaatgcatactactgaggacaggatcagtagagcggttggacccagac
aagcccaagtgtcatttatacacggtgatcaaagtgagaatgagctaccaggattgggg
ggcaaggaagataggagggtcaaacagggtcggggagaggccagggagagctacagaga
agccgggtccagcagagcaagtgatgcaagagctgtccctcttccaaccagcacacccc
tagacattgacactgcatcagagtcgggccaagatccgcaggacagtcgaaggtcagct
```

```
gacgccctgctcaggttgcaggccatggcaggaatcctggaagaacaaggctcagacac
ggacatctccagggtgtataatgacagagatcttctggactag
```

We have the dates (in fractional years), the cities, the number of cases, and the 456 nucleotide base pairs, for each of 283 distinct measles virus specimens.

```
names(MeasleExtract)
```

```
[1] "X"     "cases" "years" "city"  "data"
```

```
years <- MeasleExtract$years
city <- MeasleExtract$city
cases <- MeasleExtract$cases
data <- MeasleExtract$data
```

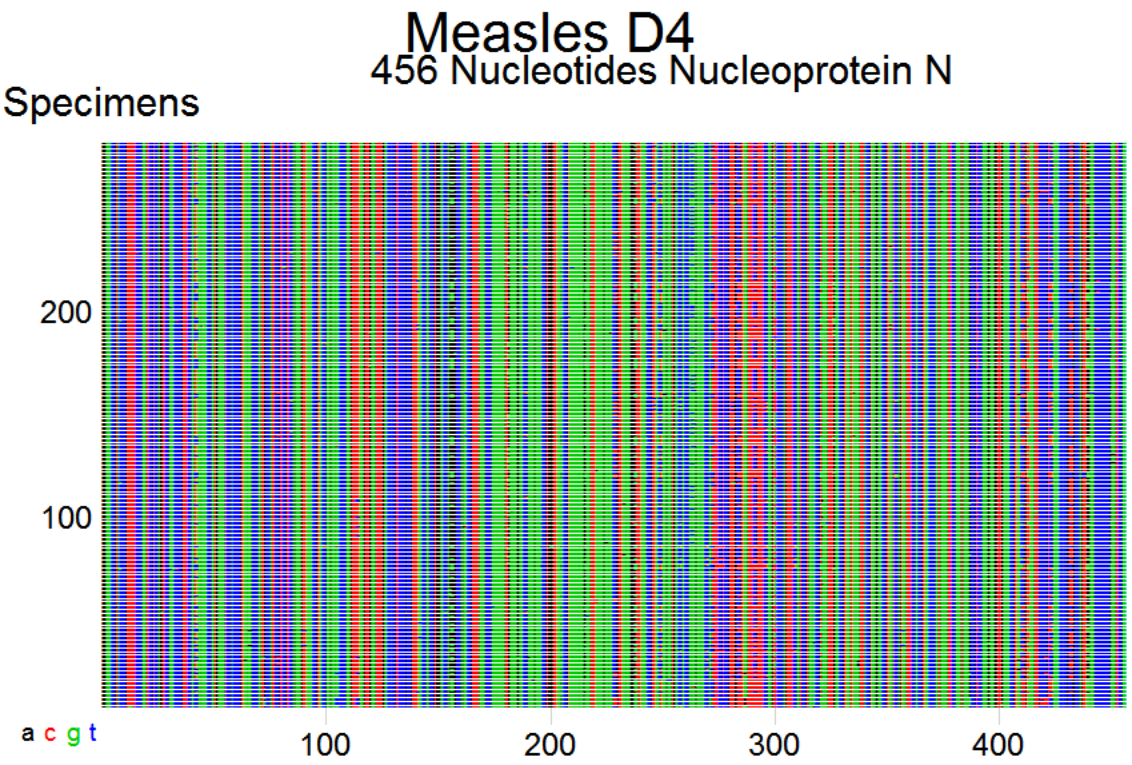
The nucleotide data is changed to a vector of characters using Split:

```
MeaslesNucleotides <- Split(data)

tiff("pictures/MeaslesNucleotides.tif",w=900,
h=600)

PlotCharMatrix(MeaslesNucleotides, at=c(200,250, 0),
ylab="Measles D4/456 Nucleotides Nucleoprotein
N/Specimens",
chr = c("a", "c", "g", "t"), col = 1:4)

dev.off()
```



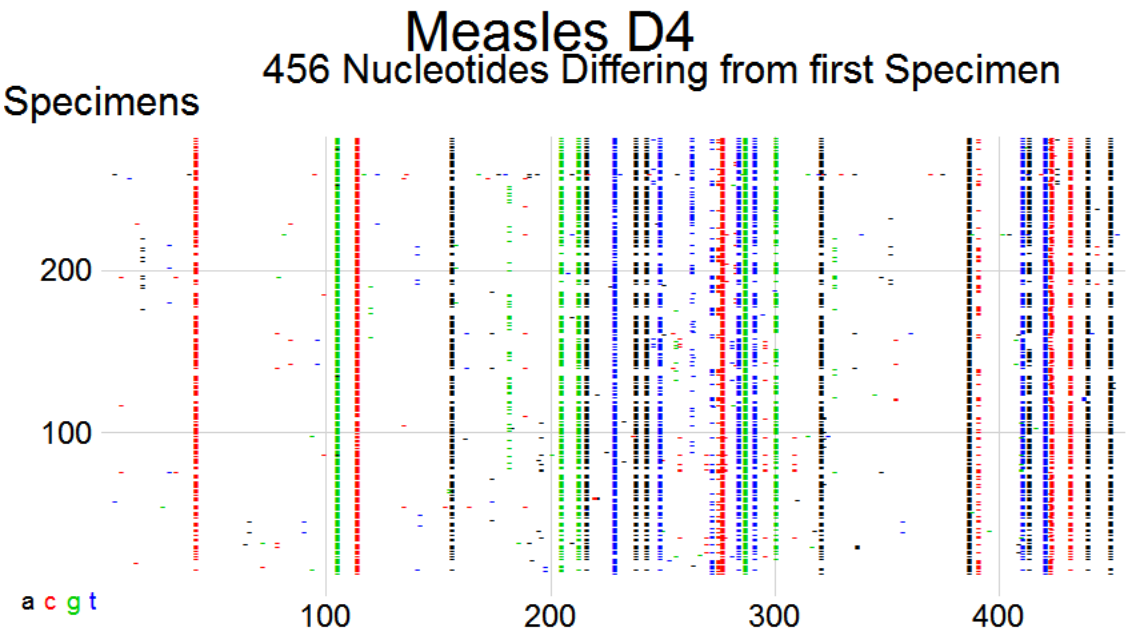
We ignore the 3 characters in the data set that are not "a","c","g","t", in positions where the nucleotide is not completely known. One can see that most columns are constant. A few positions have nucleotides changing as we move from the oldest to the most recent viruses. To examine them, we need to blank out the non-discriminatory identical positions.

5 Plot differences from earliest specimen

We plot only those nucleotides that vary from the nucleotide in the same position in the earliest specimen.

```
MND <- MeaslesNucleotides
for( row in 2:dim(MND)[1])
MND[row, MND[row, ] == MND[1, ]] <- "."

tiff("pictures/MeaslesNucleotidesDiffering.tif",w=9
00, h=500)
PlotCharMatrix(MND, at=c(200,250, 0),
ylab="Measles D4/456 Nucleotides Differing from first
Specimen/Specimens",
chr = c("a", "c", "g", "t"), col = 1:4)
dev.off()
```



We see 25 "variant" positions where the nucleotide changed value from the first specimen in a majority of later specimens. In 7 positions almost all the specimens have changed values. In other positions, there are substantial fractions that agree with the original specimen, and suggest different clades of viruses. We need to order the viruses by descent rather than year of submission to explore further.

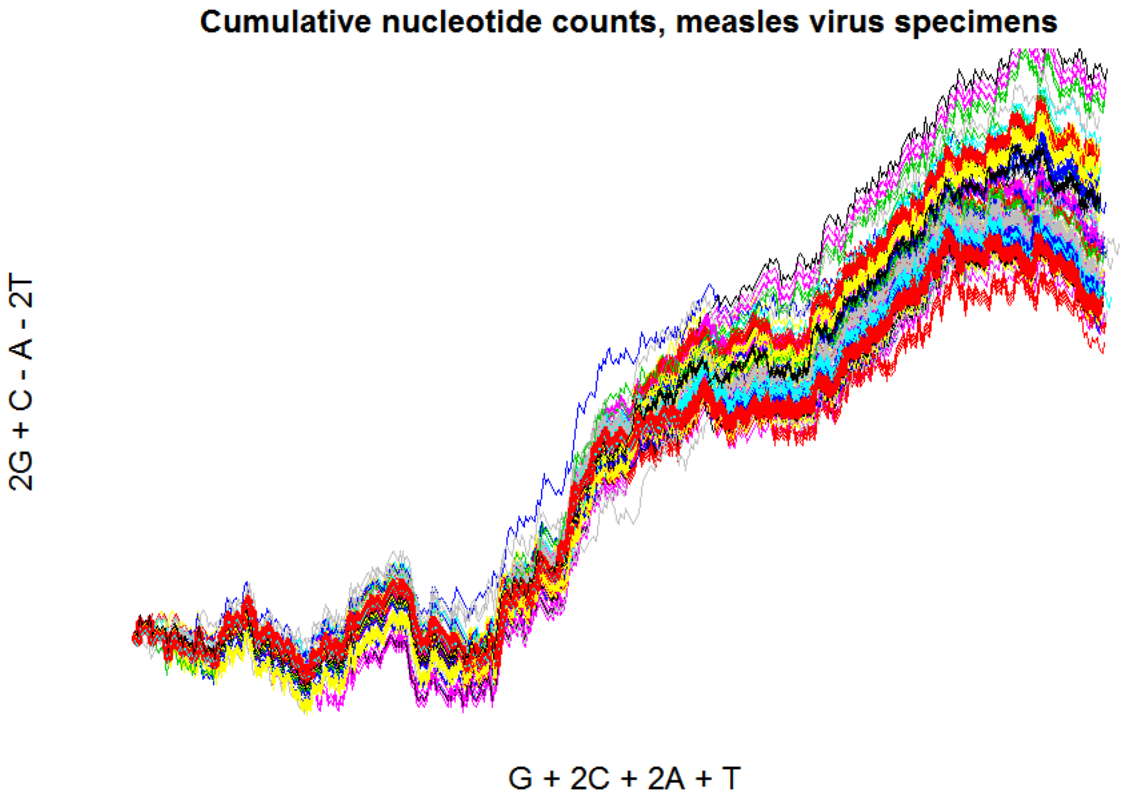
4 Plot cumulative GCAT occurrences

The 456 nucleotide values are plotted for each virus specimen. Cumulatively sum the value of G+2A+2C+T on the x axis, and the value of 2G−A+C-2T on the y axis. The change in x,y value between successive nucleotides is (1, ±2) or (2,±1), always a line of the same length, and always increasing in x. The curve uniquely characterizes the sample sequence.

Most changes are transitions within purines (G<->A) or within pyrimidines (C<->T), so a similar plot is obtained plotting purines (G+A) versus pyrimidines (C+T) but that plot does not characterize the sequence .

```
tiff("pictures/CumulativeNucleotide.tif",w=900,
h=600)
par(mar=c(4,5,3,2))

DragonPlot(data, cases)
title(" Cumulative nucleotide counts, measles virus
specimens",cex.main=2)
dev.off()
```

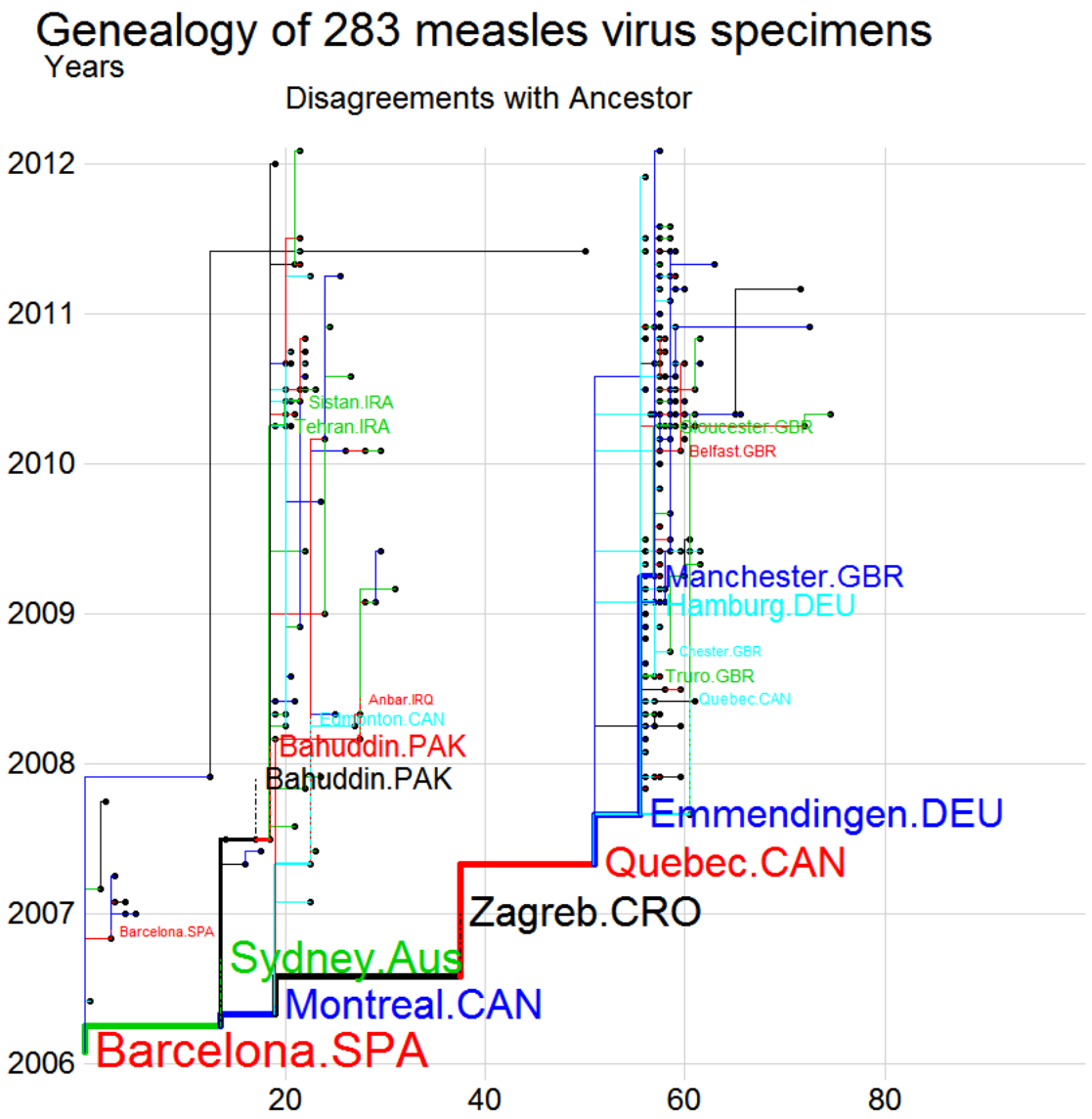


We see two distinct bands, the red and yellow through all positions, and some suggested lines blue, red, purple, for the higher positions..maybe another 3 or 4. Each band represent measles strains going through a similar evolution over time.

5 Genealogy

We find the closest earlier specimen, the ancestor, to each specimen, as measured by the number of positions in which they disagree. Each point is plotted at y-value equal to year, and at x-value distant from its ancestor by the number of positions in which they differ. The lines connect each sample to its computed ancestor.

```
tiff("pictures/Genealogy.tif", w=900, h=900)
a <- Ancestor(data, years, city,
ylab ="Genealogy of 283 measles virus specimens/Years/
Disagreements with Ancestor",at=c(40,0,40))
dev.off()
```



6 Conclusions

We see two distinct main lines, the Sydney group (2006.5) and the Emmindingen group(2007.7). The Sydney group is mainly Asian and Middle East, the Emmendingen group is European. The two groups differ in about 40 base pairs. The Sydney group differs from the earliest Barcelona sequence in about 20 positions.

7 Data Preparation

The file" MeaslesGenbank" copied from the Genbank depository is located in the data subdirectory of the present directory.
To check file is in that subdirectory:

```
list.files("data/")
```

```
[1] "MeasleExtract.csv" "MeaslesGenbank"
```

```
g <- scan("data/MeaslesGenbank", what="",sep="\n")
```

```
Read 50134 items
```

7.1 Submissions

```
head(g,53)
```

```
[1] "LOCUS       JQ809708                 453 bp    cRNA
linear   VRL 11-APR-2012"
 [2] "DEFINITION  Measles virus genotype D4 strain
MVs/Alberta.CAN/09.12[D4]"
 [3] "            nucleoprotein (N) gene, partial cds."
 [4] "ACCESSION   JQ809708"
 [5] "VERSION     JQ809708.1  GI:383478576"
 [6] "KEYWORDS    ."
 [7] "SOURCE      Measles virus genotype D4"
 [8] "  ORGANISM  Measles virus genotype D4"
 [9] "            Viruses; ssRNA negative-strand viruses;
Mononegavirales;"
[10] "            Paramyxoviridae; Paramyxovirinae;
Morbillivirus."
[11] "REFERENCE   1  (bases 1 to 453)"
[12] "  AUTHORS   Mendoza,L., Hiebert,J. and Severini,A."
```

```
[13] "  TITLE     Measles genotype surveillance in Canada"
[14] "  JOURNAL   Unpublished"
[15] "REFERENCE   2  (bases 1 to 453)"
[16] "  AUTHORS   Mendoza,L., Hiebert,J. and Severini,A."
[17] "  TITLE     Direct Submission"
[18] "  JOURNAL   Submitted (22-MAR-2012) Public Health Agency
of Canada, National"
[19] "            Microbiology Laboratory, 1015 Arlington
Street, Winnipeg, Manitoba"
[20] "            R3E 3R2, Canada"
[21] "FEATURES             Location/Qualifiers"
[22] "     source          1..453"
[23] "                     /organism=\"Measles virus genotype
D4\""
[24] "                     /mol_type=\"viral cRNA\""
[25] "
/strain=\"MVs/Alberta.CAN/09.12[D4]\""
[26] "                     /isolate=\"VE12-0265\""
[27] "                     /isolation_source=\"urine\""
[28] "                     /host=\"Homo sapiens\""
[29] "                     /db_xref=\"taxon:170525\""
[30] "                     /country=\"India\""
[31] "                     /collection_date=\"15-Mar-2012\""
[32] "                     /note=\"genotype: D4\""
[33] "     gene            <1..>453"
[34] "                     /gene=\"N\""
[35] "     CDS             <1..>453"
[36] "                     /gene=\"N\""
[37] "                     /codon_start=1"
[38] "                     /product=\"nucleoprotein\""
[39] "                     /protein_id=\"AFH36327.1\""
[40] "                     /db_xref=\"GI:383478577\""
[41] "
/translation=\"KVSSTLASELGITAEDARLVSEIAMHTTEDRISRAVGPRQAQVS
"
[42] "
FIHGDQSENELPGLGGKEDRRVKQGRGEARESYRETGSNRASDARAAHLPISTPLDID"
[43] "
TASESGQDPQDSRRSADALLRLQAMAGILEEQGSDTDISRVYNDKDLLD\""
[44] "ORIGIN      "
```

```
[45] "        1 aaggtcagtt ccacattggc atctgaactc ggtatcactg
ccgaggatgc aaggcttgtt"
[46] "       61 tcagagattg caatgcatac taccgaggac aggatcagta
gagcggttgg acccagacaa"
[47] "      121 gctcaagtgt catttataca cggtgatcaa agtgaaaatg
agctaccagg attgggggc"
[48] "      181 aaggaagata ggagggtcaa acagggtcgg ggagaagcca
gggagagcta cagagaaacc"
[49] "      241 ggatccaata gagcaagtga tgcgagagct gcccatcttc
caatcagcac acccctagac"
[50] "      301 attgacactg catcagagtc aggccaagat ccgcaggaca
gtcgaaggtc agctgacgcc"
[51] "      361 ctgctcaggt tgcaggccat ggcaggaatc ttggaagaac
aaggctcaga tacggacatc"
[52] "      421 tctagggtgt acaatgacaa agatcttcta gac"
[53] "//"
```

This is one of the measles sequences submitted to Genbank. We need to pick out the city and country, the collection date, and the nucleotide sequence.

7.2 Select markers, names and sequences

The submission name is located after "VRL":

```
first <- grep("VRL",g)
vname <- g[first+1]
head(vname, 2)
```

```
[1] "DEFINITION  Measles virus genotype D4 strain
MVs/Alberta.CAN/09.12[D4]" "DEFINITION  Measles virus
genotype D4 strain MVs/Ilfov.ROU/01.12/[D4]"
```

The nucleotide sequence appears between "ORIGIN" and "//":

```
origin <- grep("ORIGIN",g)
end <- grep("//",g)
c(length(first), length(origin), length(end))
```

```
[1] 990 990 990
```

```
rbind(first, origin, end)[ , 1:9]
```

```
       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
first     1   54  103  159  215  270  328  386  444
origin   44   93  147  203  259  316  374  432  490
end      53  102  158  214  269  327  385  443  501
```

```
data <- rep("",length(vname))
for (d in 1:length(vname)){
 for ( s in (origin[d] + 1):(end[d] -1 ))
data[d] <-paste(data[d], substr(g[s],10,79), sep="")
}
data<- gsub(" ","",data)
head(data, 2)
```

```
[1]
"aaggtcagttccacattggcatctgaactcggtatcactgccgaggatgcaaggcttg
tttcagagattgcaatgcatactaccgaggacaggatcagtagagcggttggacccaga
caagctcaagtgtcatttatacacggtgatcaaagtgaaaatgagctaccaggattggg
gggcaaggaagataggagggtcaaacagggtcggggagaagccagggagagctacagag
aaaccggatccaatagagcaagtgatgcgagagctgcccatcttccaatcagcacaccc
ctagacattgacactgcatcagagtcaggccaagatccgcaggacagtcgaaggtcagc
tgacgccctgctcaggttgcaggccatggcaggaatcttggaagaacaaggctcagata
cggacatctctagggtgtacaatgacaaagatcttctagac"
[2]
"aaggtcagttccacattggcatctgaactcggtatcactgccgaggatgcaaggcttg
tttcagagattgcaatgcatactactgaggacaggatcagtagagcggttggacccaga
caagcccaagtgtcatttatacacggtgatcaaagtgaaaatgagctaccaggattggg
gggcaaggaagataggagggtcaaacagggtcgggggaagccagggagagctacagag
aaaccggatccagtagagcaagtgatgtgagagctgcccatcttccaatcagcactccc
ctagacgttgacactgcatcagagtcaggccaagatccgcaggacagtcgaaggtcagc
```

```
tgacgccctgctcaggttgcaggccatggcaggaatcttggaagaacaaggctcagata
cagacatctctcgggtgtacaatgacaaagatcttctagactag"
```

7.3 Dates when speciments collected

Use "collection" for the collection date:

```
collect <- rep(NA, length(vname))
for ( v in 1 : length(vname)){
u <- grep('collection.+[A-Za-za-z].[0-9]{4}',
g[first[v] : origin[v]])
  if (length(u)==1) collect[v] <- u + first[v] - 1
}
```

Frequently check data lengths to make sure all the submissions have the variables of interest:

```
c(length(vname), length(data), length(collect))
```

```
[1] 990 990 990
```

Some dates cannot be found:

```
head(collect)
```

```
[1]  31  83  NA  NA  NA 302
```

Use only submissions with 456 bp to avoid alignment issues:

```
use456 <- grepl("456 bp    cRNA",g[first])
use <- use456 & !is.na(collect)
vname <- vname[use]
data <- data[use]
collect <- collect[use]
```

Extract months and years from collection date:

```
cr <- regexpr('[A-Z][a-z][a-z].[0-9]{4}',g[collect])
date <- substr(g[collect],cr,cr+7)
months <- c("Jan","Feb","Mar","Apr","May","Jun",
```

```
            "Jul","Aug","Sep","Oct","Nov","Dec")
years <- rep(0,length(date))
for (d in 1:length(date)){
  mnth <- substr(date[d], 1, 3)
  years[d] <- as.numeric(substr(date[d], 5, 8)) +
              which(mnth == months)/12
}
head(date, 5)
```

```
[1] "Jan-2012" "Apr-2006" "Apr-2007" "May-2007" "Feb-2011"
```

```
head(years, 5)
```

```
[1] 2012.083333 2006.333333 2007.333333 2007.416667
2011.166667
```

```
data <- data[order(years)]
vname <- vname[order(years)]
years <- years[order(years)]
head(years)
```

```
[1] 2006.083333 2006.250000 2006.250000 2006.250000
2006.333333 2006.416667
```

7.4 Use only unique specimens

Select one specimen from each set of specimens with the same DNA and collection date:

```
mm <- rep(T,length(data))

for ( d in 2 : length(data))
mm[d] <-!(data[d] %in%
data[d-1]&years[d]==years[d-1])

cases <- rep(1,length(data))
```

```
for ( d in 1:(length(data)-1) ){
 use <- which(mm[(d + 1) : length(data)])
 if(length(use)>0) cases[d] <- min(use)
}

data <- data[mm]
head(mm)
```

```
[1]  TRUE  TRUE FALSE FALSE  TRUE  TRUE
```

The third and fourth specimens are identical to the second.

```
vname <- vname[mm]
years <- years[mm]
cases <- cases[mm]
tail(cases,15)
```

```
[1]  1  6  1 45  1  2  1  4  5  1  1  1  1  1  1
```

One of the specimens appeared 45 times.

7.5 Identify city where collected

The city appears in a certain format in the submission name:

```
use <- grepl("[a-zA-Z]+\.[A-Za-z]+", vname)
print(sum(use))
```

```
[1] 283
```

```
data <- data[use]
years <- years[use]
vname <- vname[use]
cases <- cases[use]
city <-regmatches(vname,
regexpr("[a-zA-Z]+\.[A-Za-z]+",vname))
head(city)
```

```
[1] "Barcelona.SPA" "Sydney.Aus"    "Montreal.CAN"
"Valencia.SPA"  "Zagreb.CRO"    "Barcelona.SPA"
```

```
length(data)
```

```
[1] 283
```

7.6 Save data in convenient form for R

```
MeasleExtract <- data.frame( cases=cases,
years=years,
city=city,data=data,stringsAsFactors=F,row.names=vn
ame)
dim(MeasleExtract)
```

```
[1] 283   4
```

```
head(MeasleExtract, 1)
```

```
cases       years          city
DEFINITION  Measles virus strain MVs/Barcelona.SPA/07.06/2
[D4] nucleoprotein     1 2006.083333 Barcelona.SPA
```

```
data
DEFINITION  Measles virus strain MVs/Barcelona.SPA/07.06/2
[D4] nucleoprotein
aaggtcagttccacattggcatctgaactcggtatcactgctgaggatgcaaggcttgt
ttcagagattgcaatgcatactactgaggacaggatcagtagagctgttggacctagac
aagcccaagtgtcatttatacacggtgatcaaagtgagaatgagctaccaggattgggg
ggcaaggaagataggagggtcaaacagagtcggggagaggccagggagagccacagaga
agccgggtccagcagagcaagtgatgcgagagctgcccattttccaaccaacacacccc
tagacattgacactgcatcagagtcgggccaagatccgcaggacagtcgaaggtcagct
gacgccctgctcaggttgcaggccatggcagggatcttggaagaacaaggctcagacac
ggacatccctagggtgtataatgacagagatcttctggactag
```

```
write.csv(MeasleExtract,
file="data/MeasleExtract.csv",
          row.names=T)
```

8 Functions

```
DragonPlot <- function(str,cases){
# plot series of nucleotides cumulatively

ls <- length(str)

# find  range for initial plot
rg <- rep(0,ls)
rc <- rg
ra <- rg
rt <- rg
for(v in 1:ls){
 rg[v] <- sum(unlist(strsplit(str[v],""))=="g")
 rc[v] <- sum(unlist(strsplit(str[v],""))=="c")
 ra[v] <- sum(unlist(strsplit(str[v],""))=="a")
 rt[v] <- sum(unlist(strsplit(str[v],""))=="t")
}

# Define x and y variables to identify nucleotide
sequence
yend <- 2*rg + rc - ra - 2*rt
xend <- rg + 2*rc+ 2*ra + rt
plot(c(0,max(xend)),c(-10,max(yend)),
     pch="",axes=F, cex.lab=2,
     xlab="G + 2C + 2A + T", ylab="2G + C - A - 2T")

for(v in 1:ls){
 cg <- cumsum(unlist(strsplit(str[v],""))=="g")
 cc <- cumsum(unlist(strsplit(str[v],""))=="c")
 ca <- cumsum(unlist(strsplit(str[v],""))=="a")
 ct <- cumsum(unlist(strsplit(str[v],""))=="t")
 color <- max(2*cg + cc - ca - 2*ct)

# make a small random change in y so that slightly
different series will show
lines(cg+2*cc+2*ca+ct, 2*cg+cc-ca-2*ct+2*runif(1),
col=color,lwd=sqrt(cases[v]))
}
```

```
invisible()
}

Split <- function(data){
# split vector of strings into character matrix
dd <- strsplit(data, split="")
dd <- t(matrix(unlist(dd),nrow
<-nchar(data[1]),ncol=length(data)))
return(dd)
}

Grid <- function(xticks, yticks, ylab="",
       at=(min(xticks)+ mean(xticks))/2, cex=2){
# background for plot using grid of light grey lines

par(mar=c(3,3,6,2))
plot(1, 1,  xlim=range(xticks),  ylim =
range(yticks),
          xlab="", ylab="", axes=F, pch="")

# use only interior values of tick ranges in plots
usey <- rep( T, length(yticks) )
usey[c( 1, length(yticks) )] <- F
usex <- rep( T, length(xticks) )
usex[c( 1, length(xticks) )] <- F

# grey lines in both directions
for ( row in yticks[usey] )
lines(range(xticks), c(row, row), col="light grey")
for ( col in xticks[usex] )
lines(c(col, col), range(yticks), col="light grey")

# put ylab on left top, using / to split long
expressions
ylabs <- unlist(strsplit(ylab,"/"))

# identify tick marks on both axes
if (length(yticks) > 2)
 text(pos=2, rep(min(xticks), length(yticks)-2 ),
```

```
      yticks[usey], yticks[usey], cex=2, xpd=T)
if (length(xticks)>2)
 text(pos=1, xticks[usex],  rep(min(yticks),
     length(xticks)-2), xticks[usex], cex=2, xpd=T)

# do top texts
lylabs <- min(5, length(ylabs))
cex <- rep(cex, length(ylabs))
cex[1] <- cex[1]+1

if(lylabs > 0){
mtext(ylabs, side=3,line = (5/lylabs)*(lylabs-1):0,
     at = at, cex=cex)
}

par(mar=c(5, 4, 4, 2))

invisible()
}
```

```
PlotCharMatrix <- function(data, chr="", col="",
ylab="", at=""){
# plot character matrix

# define unassigned variables
if(!is.matrix(data)) return(" data not matrix")
if(length(chr) == 1) chr <- unique(as.vector(data))
if(length(col) == 1) col <- rainbow( length(chr) )
if(ylab=="") ylab <- deparse(substitute(data))

# set up background using Grid
nrow <- dim(data)[1]
ncol <- dim(data)[2]
xticks <- 1:ncol
yticks <- 1:nrow
yinc <- 10^(ceiling(log(nrow/10, base=10)))
xinc <- 10^(ceiling(log(ncol/10, base=10)))
yticks <- c(seq(0, nrow, yinc), nrow)
xticks <- c(seq(0, ncol, xinc), ncol)

Grid(xticks, yticks, ylab=ylab, at=at, cex=2.5)

# plot each character
for ( ch in 1:length(chr)){
where <- which(data==chr[ch])
y <- (where-1) %% nrow + 1
x <- (where-1) %/% nrow + 1
points( x, y, col=col[ch], pch="-")
}
```

```
# plot color legend
text(pos=4, ((1:length(chr))-5)* xinc/10,
rep(-4, length(chr)), chr, col=col, cex=1.6, xpd=T)

invisible()
}
```

```
NotAgree <- function(str,strvec){
# number of agreement of vector of strings y with x

notagree <- rep(0, length(strvec))
cstr <- unlist(strsplit(str, ""))

for( s in 1:length(strvec)) {
  cstrvec <- unlist(strsplit(strvec[s], ""))
  notagree[s] <- sum(cstrvec != cstr)
}
return(notagree)
}
```

```
Ancestor <-
function(data, years, city, ylab="", at=0,
whenshow=5){
# find closest data point in previous years
# years and data assumed in order strictly of time

ancestor <- 1:length(data)
notagree <- rep(0,length(data))
for( d in 2:length(data)){
  dist <- NotAgree(data[d], data[1:(d-1)])
  ancestor[d] <- min(which(dist == min(dist)))
 notagree[d] <- min(dist)

}

# plot out distances
x <- rep(0,length(data))
x[1]<- 0
for ( d in 2:length(data)){
x[d] <- notagree[d] + x[ancestor[d]] + 0.5
}

# f is number of descendants, all generations
f <- rep(1, length(data))
```

```
for( d in length(data):1)
 f[ancestor[d]] <- f[d] + f[ancestor[d]]

# background grid
Grid(seq(0,100,20),c(2005.9,2006:2012,2012.1),
    ylab=ylab, at=at)

# points and lines according to ancestor function
points(x, years, pch=16)
for ( d in 2:length(data)){
lines(c(x[d],x[ancestor[d]],x[ancestor[d]]),
     c(years[d],years[d],years[ancestor[d]] ),
     col=1 + d%%5,lwd=log(f[d]))
}

# put in names separated
d <- (1:length(data))[f > whenshow]
SeparateText(x[d],years[d],city[d],
cex=log(f[d])/2,col=1+d%%5,xlim=c(0,110),ylim=c(200
6,2012),)

invisible()
}
```

```
SeparateText <- function(x,y,text,
cex=1,col=1,xlim=c(0,0),ylim=c(0,0), adj=c(1,1)){
# separate text placement to avoid overlap

# some needed functions
#-----------------------------------------------
----
# helical search sequence
helix <- function(r,R,nx=101,ny=101){

# helix of positions for rectangle r in rectangle R

# set limiting values and centre of r
x <- (r[1] + r[3])/2
y <- (r[2] + r[4])/2
dx <- x - r[1]
dy <- y - r[2]
xmin <- R[1] + dx
```

```
xmax <- R[3] - dx
ymin <- R[2] + dy
ymax <- R[4] - dy

if(xmin >= xmax | ymin >= ymax) return(NULL)
if(nx < 2 | ny < 2) return(NULL)

# make sequence of grid points in large rectangle
xg <- rep( c( xmin + (1:nx)*(x-xmin)/(nx-1),
              x + (1:nx)*(xmax-x)/(nx-1) ), 2*ny )
yg <- rep( c( ymin + (1:ny)*(y-ymin)/(ny-1),
              y + (1:ny)*(ymax-y)/(ny-1) ), 2*nx )
yg <- as.vector(t( matrix(yg, nrow=2*ny, ncol= 2*nx)
))

# put them in order of eudclidean distance from x,y
dist <- (x-xg)^2 + (y-yg)^2
xg <- xg[order(dist)]
yg <- yg[order(dist)]

# define shifted rectangles
rg <- matrix(0, nrow=length(xg), ncol=4)

rg[, 1] <- r[1] + xg - x
rg[, 2] <- r[2] + yg - y
rg[, 3] <- r[3] + xg - x
rg[, 4] <- r[4] + yg - y

return(rg)
}

#-----------------------------------------------
----
Overlap <- function(r,R){
# does rectangle r overlap any of the rectangles R
if(is.vector(R)) R <- matrix(R, nrow=1)
return(
sum(  (r[1] < R[,3]) & (R[,1] < r[3])  &
     (r[2] < R[,4]) & (R[,2] < r[4])   ) > 0
```

```
     )

}

#-----------------------------------------------
----

PlaceRectangle <- function(Rect, nx=101, ny=101,
xlim, ylim){

# place rectangles so they dont overlap
# search on nx by ny grid

# check Rect contains rectangles and set bigrect
if(!is.matrix(Rect)) return ("input not matrix")
if(dim(Rect)[2] != 4) return("input not rectangles")

minx <- min(Rect[,1])
miny <- min(Rect[,2])
maxx <- max(Rect[,3])
maxy <- max(Rect[,4])
rx <- maxx-minx
ry <- maxy-miny
Bigrect <- c(xlim[1], ylim[1], xlim[2], ylim[2])

# place each rectangle closest not overlapping
previous
for( rect in 2:dim(Rect)[1] ){
 h <- helix(Rect[rect, ], Bigrect, nx=nx, ny=ny)

 for( r in 1:dim(h)[1] ){
   Rect[rect, ] <- h[r,]

   if( !Overlap(h[r,], Rect[1:(rect-1),]) ) break
 }

}
return(Rect)
}
```

```
#-----------------------------------------------
----
# main function starts here
Rect <- matrix(0, nrow=length(x), ncol=4)

# set limits and half-ranges for x, y
if (xlim[2] - xlim[1] == 0) xlim <- c(min(x), max(x))
if (ylim[2] - ylim[1] == 0) ylim <- c(min(y), max(y))
xr <- 0.5 * adj[1] *(xlim[2] - xlim[1]) * cex/40
yr <- 0.5 * adj[2] *(ylim[2] - ylim[1]) * cex/60

# Define Rectangles and place them nonoverlapping
Rect[, 1] <- x
Rect[, 2] <- y - yr
Rect[, 3] <- x + 2*xr*nchar(text)
Rect[, 4] <- y + yr

# reorder all quantities by area of rectangles
ordR <- order( -(Rect[, 3]-Rect[, 1])*(Rect[,
2]-Rect[, 1]) )
Rect <- Rect[ordR, ]
col <- col[ordR]
cex <- cex[ordR]
text <- text[ordR]
yr <- yr[ordR]
xr <- xr[ordR]
xx <- x[ordR]
yy <- y[ordR]

# expand limits to include all rectangles
xlim[1] <- min( c(xlim[1], Rect[,1]) )
ylim[1] <- min( c(ylim[1], Rect[,2]) )
xlim[2] <- max( c(xlim[2], Rect[,3]) )
ylim[2] <- max( c(ylim[2], Rect[,4]) )

Rect <- PlaceRectangle(Rect, nx=101, ny=101, xlim,
ylim)
```

```
#for(i in 1:dim(Rect)[1]) rect(Rect[i,1], Rect[i,2],
Rect[i,3], Rect[i,4], border="red")

# Set points at left center of rectangle
x <- Rect[, 1]
y <- Rect[, 2] + yr
for( i in 1:dim(Rect)[1] ) lines(c(x[i],xx[i]),
c(y[i],yy[i]), col=col[i], lty=4)
text(x,y, text, cex=cex, col=col, pos=4)

invisible()
}
```