Indirect Selection

Jinliang Yang

02-24-2022

Path Normalization

Tajima's D

R function for Tajima'D calculation

$$D = \frac{\pi - \theta_W}{\sqrt{Var(\pi - \theta_W)}}$$

```r
TajimaD <- function(sfs){
    #' sfs (site frequency spectrum): number of singletons, doubletons, ..., etc
    n <- length(sfs) + 1 # number of chromosomes
    ss <- sum(sfs) # number of segregating sites

    a1 <- sum(1 / seq_len(n-1))
    a2 <- sum(1 / seq_len(n-1)^2)
    b1 <- (n + 1) / (3 * (n - 1))
    b2 <- 2 * (n^2 + n + 3)/(9 * n * (n - 1))
    c1 <- b1 - 1/a1
    c2 <- b2 - (n + 2)/(a1 * n) + a2 / a1^2
    e1 <- c1 / a1
    e2 <- c2 / (a1^2 + a2)

    Vd <- e1 * ss + e2 * ss * (ss - 1)
    theta_pi <- sum(2 * seq_len(n-1) * (n - seq_len(n-1)) * sfs)/(n*(n-1))
    theta_w <- ss / a1
    res <- (theta_pi - theta_w) / sqrt(Vd)
    return(res)
}
```

Site Freq Spectrum

Simulate one SFS

```
df <- data.frame(allele=c(1,2,3,4,5), frq=c(20,3,1,1,1))
TajimaD(sfs=df$frq)

#install.packages("ggplot2")
library(ggplot2)
fsize=18 #font size
p1 <- ggplot(data=df,aes(x=df$allele, y=df$frq)) +
        geom_bar(stat="identity", position=position_dodge()) +
        theme_bw() +
        theme(#axis.text.x=element_blank(), #axis.ticks.x=element_blank(),
          axis.text=element_text(size=fsize),
          axis.title=element_text(size=fsize, face="bold"),
          legend.title = element_text(size=fsize, face="bold"),
          legend.text = element_text(size=fsize)) +
      xlab("# of individuals with derived alleles") +
      ylab("Counts")
p1
```

Site Freq Spectrum

Simulate two SFS

```
df1 <- data.frame(allele=c(1,2,3,4,5), frq=c(20,2,1,1,1), sel="Sweep")
df2 <- data.frame(allele=c(1,2,3,4,5), frq=c(6,1,2,2,4), sel="Neutral")
df <- rbind(df1, df2)
TajimaD(sfs=df1$frq)
TajimaD(sfs=df2$frq)

p2 <- ggplot(df, aes(x=allele, y=frq, fill=sel)) +
    geom_bar(stat="identity", position=position_dodge()) +
    xlab("# of individuals with derived alleles") +
    ylab("Counts") +
    #scale_fill_manual(values=c("#E69F00","#56B4E9", "#009E73")) +
    #scale_x_discrete(labels=c("-log10(mu)","-log10(nu)","Ne*s")) +
    theme(legend.position = "top", legend.title = element_blank(), axis.text=element_text(size=10),
          axis.title=element_text(size=fsize, face="bold"),
                legend.text = element_text(size=fsize))

p2
```

Obtain SFS from the sequencing data

Now we switch from our local computer to HCC

```
ssh ID@crane.unl.edu
cd YOUR Git Repo
git pull
```

```
# request a computing node
srun --qos=short --nodes=1 --licenses=common --ntasks=4 --mem 8G --time 1:30:00 --pty bash

module load R
R
```

Obtain SFS from the sequencing data

```
geno <- read.table("data/geno.txt", header=FALSE)
dim(geno)
head(geno)

for(i in 5:24){
  # replace slash and everything after it as nothing
  geno$newcol <- gsub("/.*", "", geno[,i] )
  # extract the line name
  nm <- names(geno)[i]
  # assign name for this allele
  names(geno)[ncol(geno)] <- paste0(nm, sep="_a1")
  geno$newcol <- gsub(".*/", "", geno[,i] )
  names(geno)[ncol(geno)] <- paste0(nm, sep="_a2")
}
# count the number of derived allele
geno[, 25:64] <- apply(geno[, 25:64], 2, as.numeric)
geno$da <- apply(geno[, 25:64], 1, sum)
write.table(geno[, c("V1", "V2", "da")], "cache/Mt_derived_alleles.csv",
            sep=",", row.names = FALSE, quote=FALSE)
```

Obtain SFS from the sequencing data

```
df <- read.csv("cache/Mt_derived_alleles.csv")
sfs <- table(df$da)
TajimaD(sfs=sfs)
```

–

Calculate Tajima's D for windows (10 kb)

```
df <- read.csv("cache/Mt_derived_alleles.csv")
names(df)[1:2] <- c("chr", "pos")
winsize = 10000
df$win <- round(df$pos/winsize,0) + 1

res <- data.frame()
sfs0 <- data.frame(Var1=1:19, value=0)
```

```
for(i in 1:58){
  sub <- subset(df, win %in% i)
  tem <- as.data.frame(table(sub$da))
  if(nrow(tem) > 0){
    newsfs <- merge(sfs0, tem, by="Var1", all.x=TRUE)
    newsfs[is.na(newsfs$Freq),]$Freq <- 0
    out <- data.frame(win=i, tajimad = TajimaD(sfs=newsfs$Freq))
    res <- rbind(res, out)
  }
}
```

Visualize the Tajima'D results

Scatter plot

```
pdf("graphs/tajimad_res.pdf", height=5, width=10)
plot(x=res$win, y=res$tajimad, pch=16, col="red", xlab="Physical Position (10-kb)", ylab="Tajima D")
dev.off()
```

–

Histogram

```
pdf("graphs/hist_tajimad_res.pdf", height=5, width=5)
hist(res$tajimad, xlab="Tajima D", ylab="Frequency")
dev.off()
```

General feature format (GFF) from EnsemblPlants

Maize reference genome

change to `largedata\lab6` folder:

```
cd largedata
mkdir lab6
cd lab6
```

–

We will download and unzip the Mt GFF3 file

```
wget ftp://ftp.ensemblgenomes.org/pub/plants/release-46/gff3/zea_mays/Zea_mays.B73_RefGen_v4.46.chromose

### then unzip it
gunzip Zea_mays.B73_RefGen_v4.46.chromosome.Mt.gff3.gz
```

General feature format (GFF) version 3

```
  V1      V2         V3   V4     V5 V6 V7 V8
1 Mt Gramene chromosome    1 569630  .  .  .
2 Mt ensembl       gene 6391   6738  .  +  .
3 Mt    NCBI       mRNA 6391   6738  .  +  .
4 Mt    NCBI       exon 6391   6738  .  +  .
5 Mt    NCBI        CDS 6391   6738  .  +  0
6 Mt ensembl       gene 6951   8285  .  +  .
                  V9
1   ID=chromosome:Mt;Alias=AY506529.1,NC_007982.1;Is_circular=true
2   ID=gene:ZeamMp002;biotype=protein_coding;description=orf115-a1;
3   ID=transcript:ZeamMp002;Parent=gene:ZeamMp002;
4   Parent=transcript:ZeamMp002;Name=ZeamMp002.exon1;constitutive=1;ensembl_end_phase=0;
5   ID=CDS:ZeamMp002;Parent=transcript:ZeamMp002;
6   ID=gene:ZeamMp003;biotype=protein_coding;description=orf444
```

General feature format (GFF) version 3

```
  V1      V2         V3   V4     V5 V6 V7 V8
1 Mt Gramene chromosome    1 569630  .  .  .
2 Mt ensembl       gene 6391   6738  .  +  .
                  V9
1   ID=chromosome:Mt;Alias=AY506529.1,NC_007982.1;Is_circular=true
2   ID=gene:ZeamMp002;biotype=protein_coding;description=orf115-a1;
```

- 1 **sequence**: The name of the sequence where the feature is located.
- 2 **source**: Keyword identifying the source of the feature, like a program (e.g. RepeatMasker) or an organization (like ensembl).
- 3 **feature**: The feature type name, like “gene” or “exon”.
  - In a well structured GFF file, all the children features always follow their parents in a single block (so all exons of a transcript are put after their parent “transcript” feature line and before any other parent transcript line).
- 4 **start**: Genomic start of the feature, with a 1-base offset.
  - This is in contrast with other 0-offset half-open sequence formats, like BED.

General feature format (GFF) version 3

```
  V1      V2         V3   V4     V5 V6 V7 V8
1 Mt Gramene chromosome    1 569630  .  .  .
2 Mt ensembl       gene 6391   6738  .  +  .
                  V9
1   ID=chromosome:Mt;Alias=AY506529.1,NC_007982.1;Is_circular=true
2   ID=gene:ZeamMp002;biotype=protein_coding;description=orf115-a1;
```

- 5 **end**: Genomic end of the feature, with a 1-base offset.
  - This is the same end coordinate as it is in 0-offset half-open sequence formats, like BED.
- 6 **score**: Numeric value that generally indicates the confidence of the source in the annotated feature.
  - A value of "." (a dot) is used to define a null value.
- 7 **strand**: Single character that indicates the strand of the feature.
  - it can assume the values of "+" (positive, or 5' -> 3'),
  - "-", (negative, or 3' -> 5'), "." (undetermined).

General feature format (GFF) version 3

```
  V1      V2         V3   V4      V5 V6 V7 V8
1 Mt Gramene chromosome    1 569630  .  .  .
2 Mt ensembl       gene 6391   6738  .  +  .
                     V9
1   ID=chromosome:Mt;Alias=AY506529.1,NC_007982.1;Is_circular=true
2   ID=gene:ZeamMp002;biotype=protein_coding;description=orf115-a1;
```

- 8 **phase**: phase of CDS (**means CoDing Sequence**) features.
  - The phase indicates where the feature begins with reference to the reading frame.
  - The phase is one of the integers 0, 1, or 2, indicating the number of bases that should be removed from the beginning of this feature to reach the first base of the next codon.
- 9 **attributes**: All the other information pertaining to this feature.
  - The format, structure and content of this field is the one which varies the most between the three competing file formats.

Use R to process the GFF3 file

```
# install.package("data.table")
library("data.table")

## simply read in wouldn't work
gff <- fread("largedata/lab6/Zea_mays.B73_RefGen_v4.46.chromosome.Mt.gff3", skip="#", header=FALSE, data

## grep -v means select lines that not matching any of the specified patterns
gff <- fread(cmd='grep -v "#" largedata/lab6/Zea_mays.B73_RefGen_v4.46.chromosome.Mt.gff3', header=FALSE
```

rename each column

```
names(gff) <- c("seq", "source", "feature", "start", "end", "score", "strand", "phase", "att")
table(gff$feature)
```

Use R to process the GFF3 file

Get the start and end positions for each gene

```
gene <- subset(gff, feature %in% "gene")
gene$geneid <- gsub(".*gene:|;biotype.*", "", gene$att)
```

_

Calculate Tajima's D for each gene

```
df <- read.csv("cache/Mt_derived_alleles.csv")
names(df)[1:2] <- c("chr", "pos")

res <- data.frame()
sfs0 <- data.frame(Var1=1:19, value=0)
for(i in 1:nrow(gene)){
  sub <- subset(df, chr %in% gene$seq[i] & pos > gene$start[i] & pos < gene$end[i])
  tem <- as.data.frame(table(sub$da))
  if(nrow(tem) > 0){
    newsfs <- merge(sfs0, tem, by="Var1", all.x=TRUE)
    newsfs[is.na(newsfs$Freq),]$Freq <- 0
    out <- data.frame(win=i, gene=gene$geneid[i], tajimad = TajimaD(sfs=newsfs$Freq))
    res <- rbind(res, out)
  }
}
res <- res[order(res$tajimad),]
```