Direct Selection

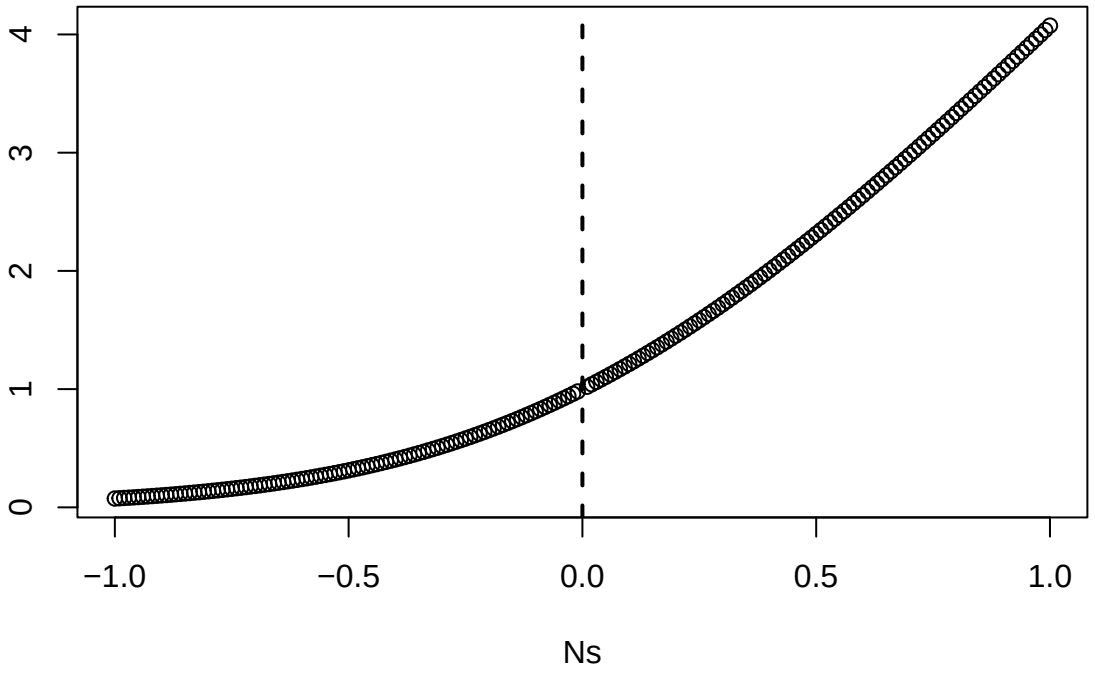
Jinliang Yang

02-16-2022

Path Normalization

Probability of fixation, relative to a neutral allele, of new, selected mutations:

```
ns <- seq(from = -1, to =1, by=0.01)
plot(ns, 4*ns/(1 - exp(-4*ns)), xlab="Ns", ylab="")
abline(v=0, lty=2, lwd=2)
```

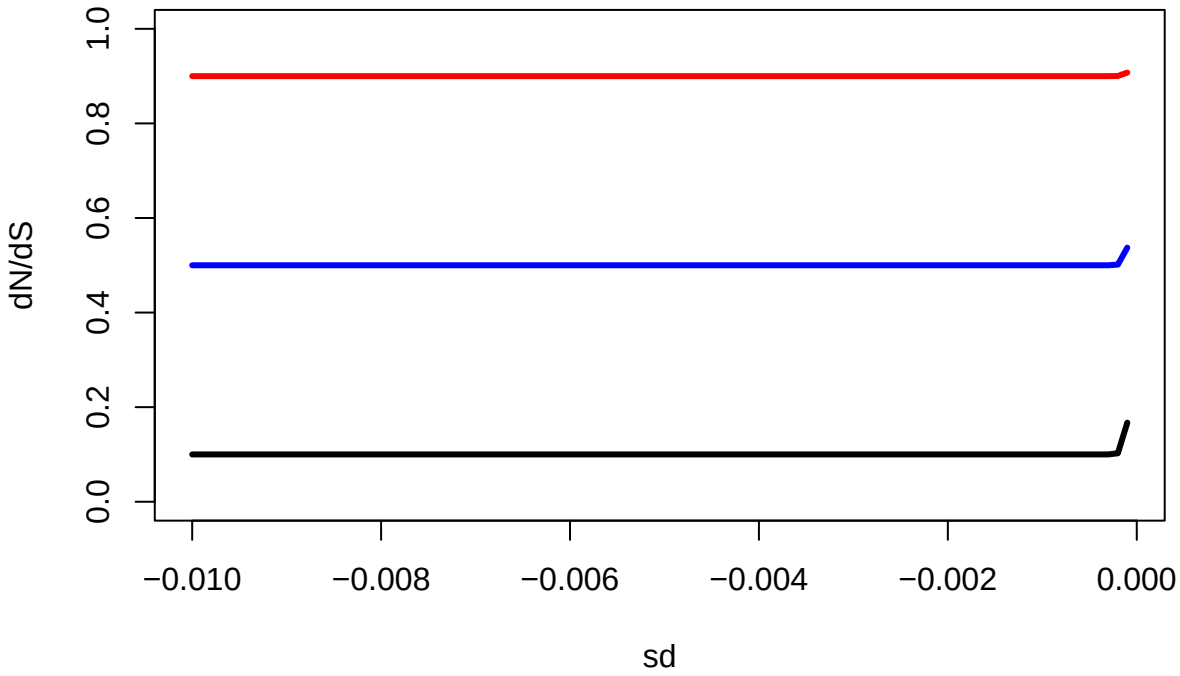


The ratio of nonsynonymous to synonymous divergence

```
dnds <- function(fa=0, fd=0, N=1000, sa=1, sd=0){
  f0 <- 1 - fa - fd
  r <- f0 + 4*N*fa*sa + (4*N*fd*sd)/(1 - exp(-4*N*sd))
  return(r)
}
```

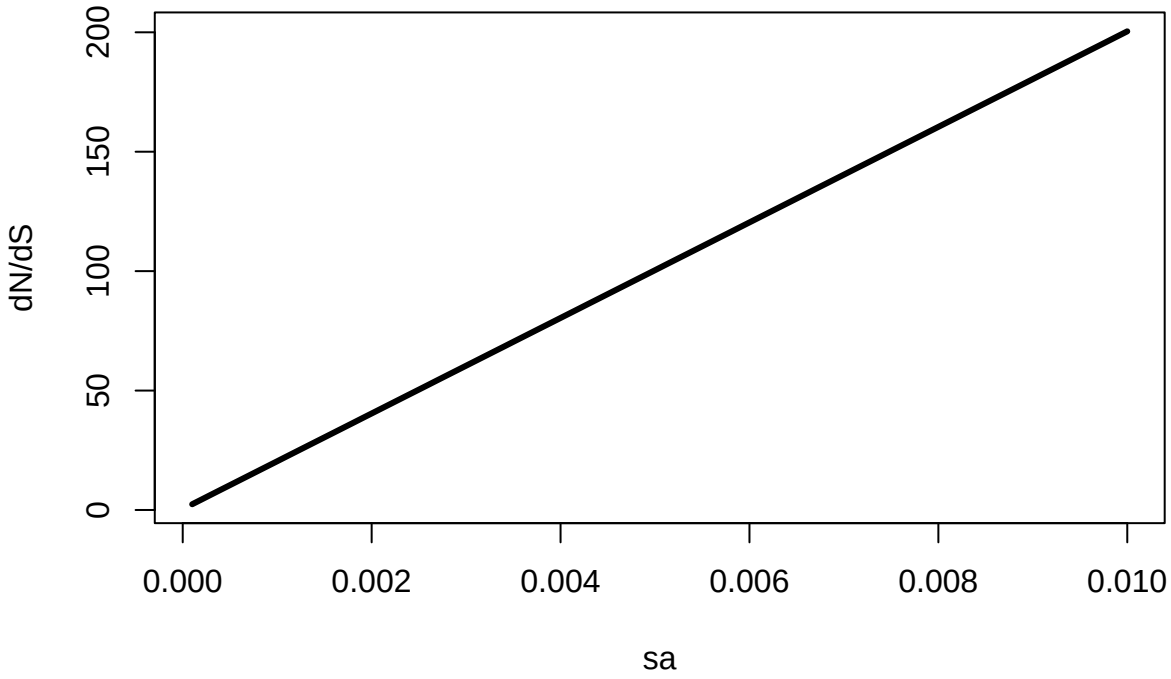
The vast majority of nonsynonymous mutations are deleterious, and negative (purifying) selection is predominant.

```
N=10000
Ns <- seq(from=1, to=100, by=1)
sd <- -Ns/N
plot(sd, dnds(fa=0, fd=0.9, N=N, sa=1, sd=sd), type="l", lty=1, lwd=3, xlab="sd", ylab="dN/dS", cex.lab=
lines(sd, dnds(fa=0, fd=0.5, N=N, sa=1, sd=sd), type="l", lty=1, lwd=3, col="blue")
lines(sd, dnds(fa=0, fd=0.1, N=N, sa=1, sd=sd), type="l", lty=1, lwd=3, col="red")
```

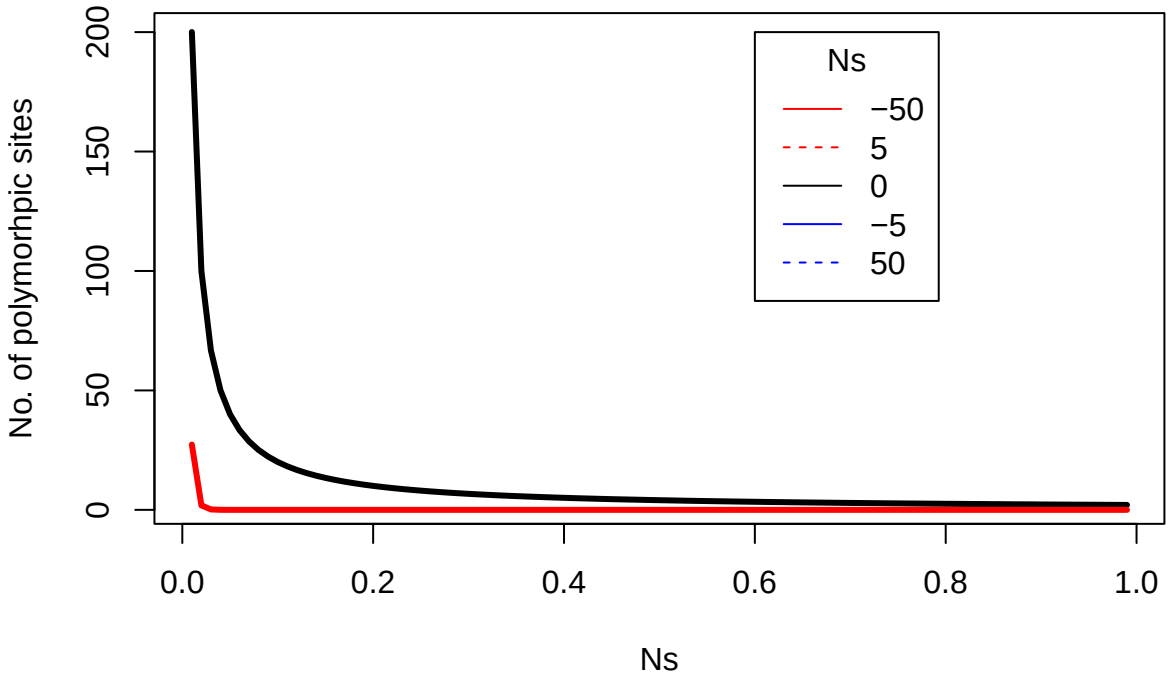


The majority of nonsynonymous mutations are deleterious, but here may be some unknown fraction of advantageous mutations.

```
N=10000
Ns <- seq(from=1, to=100, by=1)
sa <- Ns/N
plot(sa, dnds(fa=0.5, fd=0.1, N=N, sa=sa, sd=-1), type="l", lty=1, lwd=3, xlab="sa", ylab="dN/dS", cex.
```



```r
# expected freq spectra
f <- function(q, ns){
  frq = 2/(q*(1-q)) * (1 - exp(-4*ns*(1-q))) / (1 - exp(-4*ns))
  return(frq)}
q <- seq(from = 0.01, to =0.99, by=0.01)
## Ploting function
plot(q, f(q, ns=0.01), type="l", lty=1, lwd=3, xlab="Ns", ylab="No. of polymorhpic sites", cex.lab=1)
lines(q, f(q, ns=-50), type="l", lty=1, lwd=3, col="red")
#lines(q, f(q, ns=-5), type="l", lty=2, lwd=3, col="red")
#lines(q, f(q, ns=5), type="l", lty=1, lwd=3, col="blue")
#lines(q, f(q, ns=50), type="l", lty=2, lwd=3, col="blue")
legend(0.6, 200, title="Ns", legend=c("-50", "5", "0", "-5", "50"),
       col=c("red", "red", "black", "blue", "blue"),
       lty=c(1,2,1,1,2), cex=1, lwd=1)
```



- most strongly deleterious mutations are immediately removed from the population
- most strongly advantageous mutations fix very rapidly.

Work on the cluster

Login onto cluster

```
ssh USERID@crane.unl.edu # DUO to activate it
```

And then cd to your repo

```
cd $COMMON
# then cd to your repo
```

Then update your repo

```
git pull
```

If you made changes in your HCC repo, then add them, and sync with remote

```
git add --all
git commit -m "updates from cluster"
git push
```

---

submit slurm job

```
mkdir slurm-log
mkdir slurm-script
cd slurm-script
```

Use `vi` to create a slurm script

```
vi my_first_slurm_job.sh
i #insert text
#copy text
#type esc
:sq # save and quit
```

type `pwd` to find your current path

```
#!/bin/bash -l
#SBATCH -D /common/jyanglab/jyang21/courses/2022-agro932-lab
#SBATCH -o /common/jyanglab/jyang21/courses/2022-agro932-lab/slurm-log/steve-stdout-%j.txt
#SBATCH -e /common/jyanglab/jyang21/courses/2022-agro932-lab/slurm-log/steve-stderr-%j.txt
#SBATCH -J theta
#SBATCH -t 1:00:00
#SBATCH --mail-user=your_email_address@gmail.com
#SBATCH --mail-type=END #email if ends
#SBATCH --mail-type=FAIL #email if fails

set -e
set -u
# insert your script here
```

```
module load bwa samtools
mkdir largedata/lab5/
cp data/Zea_mays.B73_RefGen_v4.dna.chromosome.Mt.fa largedata/lab5

# simulate 20 individuals
cd largedata/lab5
for i in {1..20}
do
   wgsim Zea_mays.B73_RefGen_v4.dna.chromosome.Mt.fa -e 0 -d 500 -N 50000 -1 100 -2 100 -r 0.1  -R 0 -X
done

# alignment
module load bwa samtools bcftools
# index the reference genome
bwa index Zea_mays.B73_RefGen_v4.dna.chromosome.Mt.fa

# using bwa mem to align the reads to the reference genome
for i in {1..20}; do bwa mem Zea_mays.B73_RefGen_v4.dna.chromosome.Mt.fa l$i.read1.fq l$i.read2.fq | sa
# sort
for i in *.bam; do samtools sort $i -o sorted_$i; done
# index them
for i in sorted*.bam; do samtools index $i; done


### index the genome assembly
samtools faidx Zea_mays.B73_RefGen_v4.dna.chromosome.Mt.fa
### Run `mpileup` to generate VCF format
ls sorted_l*bam > bamlist.txt
samtools mpileup -g -f Zea_mays.B73_RefGen_v4.dna.chromosome.Mt.fa -b bamlist.txt > myraw.bcf
bcftools call myraw.bcf -cv -Ob -o snps.bcf
```

```
### Extract allele frequency at each position
bcftools query -f '%CHROM %POS %AF1\n' snps.bcf > frq.txt
bcftools query -f '%CHROM %POS %REF %ALT [\t%GT]\n' snps.bcf > geno.txt
```

```
cd ..
sbatch --qos=short --licenses=common --ntasks=5 --mem=10G slurm-script/my_first_slurm_job.sh

## check your job status
squeue | grep "YOUR USER ID"
```
