# Modeling the evolution of populations with multiple killer meiotic drivers

José Fabricio López Hernández, Boris Y Rubinstein, Robert L. Unckless, and Sarah E. Zanders

## Contents

## Figure 2

```
gamete.frequencies<-data.frame(x1_init=c(0.01  , 0.1 , 0.5   , 0      ,0     , 0),
                               x2_init=c(0     , 0   , 0     , 0.01   ,0.1   , 0.5),
                               x3_init=c(0     , 0   , 0     , 0      ,0     , 0),
                               x4_init=c(1-0.01,1-0.1, 1-0.5 , 1-0.01 ,1-0.1 ,1-0.5))

t.values<-create.transmission.values(tx.limits =c(0.5,0.5),ty.limits=c(0.5,0.5),step=0.5)

dplyr::tibble(gamete.frequencies)->gamete.frequencies
filter(t.values,tx==ty)->t.values
## Merge frequencies and transmission advantage values.
SameDriverAdvantage.freqs<-merging.frequencies.and.t.values(
  gamete.frequencies=gamete.frequencies,
  transmision.values=filter(t.values))

list.Frequencies<-apply(SameDriverAdvantage.freqs,MARGIN = 1,function(X){
  Simulate.TwoIdenticalDrivers(frequencies = X[c("x1_init","x2_init","x3_init","x4_init")],
                               recombi.freq = c(0),
                               t = X["tx"],generations = 500,
                               break.stable = FALSE)
})
SimulatedFrequencies<-do.call("rbind",list.Frequencies)


#### Plotting frequency of wtfA+ wtfB+ added to wtfA+ wtfB-. the othere
```

```
#### present genotype is wtfA+ wtfB+. Colors are done by distinct initial
SimulatedFrequencies%>%
  mutate(Generations=Generations+1)%>%
    ggplot()+
  ylim(0,1)+
  ylab("Genotype frequency")+
  xlab("Generations")+
  geom_line(aes(x=log10(Generations),y=F_xPlus_F_XPlus+F_xPlus_F_XMinus,
                group=id,color=id),linewidth=1.5)+
  scale_color_manual(values=c(rep("#E89524",3),
                              rep("#757E34",3)))+
  scale_x_continuous(breaks = seq(0,3,1),labels = c(0,10,100,1000),n.breaks = 4)+
  theme(axis.title.x = element_text(size=10),
        legend.justification=c(1,1),
        axis.text.x =  element_text(size=13),
        axis.title.y = element_text(size=14),
        axis.text.y  =  element_text(size=13),
        legend.text = element_text(size=10),
        legend.title = element_text(size=10),
        strip.text = element_blank(),
        legend.position= ("none"),
        strip.background = element_rect(fill="white"),
        panel.background =element_rect(fill="gray99"),
        panel.border = element_rect(color="black", size=0.5,fill=NA),
        panel.grid.major.y = element_line(color="gray75", size=0.25),
        panel.grid.major.x = element_line(color="gray75", size=0.25))+ggtitle("Figure 2B")
```

```
#### missing values due to the absence of one or the other genotype
```

```
gamete.frequencies<-data.frame(x1_init=c(0.03  , 0    ),
                               x2_init=c(0.09   , 0.09  ),
                               x3_init=c(0     , 0.03 ),
                               x4_init=c(1-0.12,1-0.12))

t.values<-create.transmission.values(tx.limits =c(1,1),ty.limits=c(1,1),step=0.05)

dplyr::tibble(gamete.frequencies)->gamete.frequencies
t.values<-t.values
## Merge frequencies and transmission advantage values.
SameDriverAdvantage.freqs<-merging.frequencies.and.t.values(
  gamete.frequencies=gamete.frequencies,
  transmision.values=filter(t.values))

list.Frequencies<-apply(SameDriverAdvantage.freqs,MARGIN = 1,function(X){
  Simulate.TwoIdenticalDrivers(frequencies = X[c("x1_init","x2_init","x3_init","x4_init")],
                               recombi.freq = c(0,0.5),
                               t = X["tx"],generations = 500,
                               break.stable = FALSE)
})
SimulatedFrequencies<-do.call("rbind",list.Frequencies)


SimulatedFrequencies$recombFreq<-paste("r=",SimulatedFrequencies$recombFreq,sep=" ")
SimulatedFrequencies%>%
```

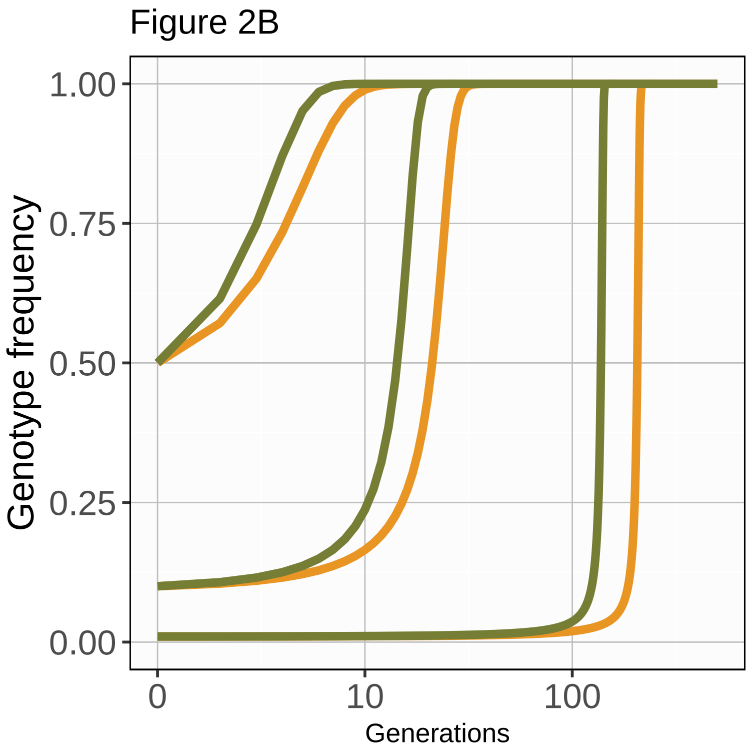

Figure 1: **Figure 2A.**Simulations of genotypes with one driver (wtfA+ wtfB-, orange) or two drivers absolutely linked in cis (wtfA+ wtfB+, black) spreading in a population where the alternate genotype lacks drivers (wtfA- wtfB-). The initial frequencies of the wtfA+ wtfB- and wtfA+ wtfB+ genotypes shown are 0.01, 0.1, and 0.5. The transmission advantage (t) for each driver is 0.5.

```
    mutate(Generations=Generations+1)%>%
    ggplot()+
  ylim(0,1)+
  ylab("Genotype frequency")+
  xlab("Generations")+
  geom_line(aes(x=log10(Generations),y=F_xPlus_F_XPlus,group=id,color=id),size=1,color="#757E34",linewi
  geom_line(aes(x=log10(Generations),y=F_xPlus_F_XMinus,group=id,color=id),size=1,color="#E89524",linew
  geom_line(aes(x=log10(Generations),y=F_xMinus_F_XPlus,group=id,color=id),size=1,color="#009262",linew
    scale_x_continuous(breaks = seq(0,3,1),labels = c(0,10,100,1000),n.breaks = 4,limits=c(0,2))+
  facet_wrap(recombFreq~x1_init)+
  theme(axis.title.x = element_text(size=10),
        legend.justification=c(1,1),
        axis.text.x =  element_text(size=13),
        axis.title.y = element_text(size=13),
        axis.text.y  =  element_text(size=13),
        legend.text = element_text(size=13),
        legend.key.size = unit(x = 2,units = "line"),
        legend.title = element_text(size=10),
        strip.background = element_rect(fill="white"),
        strip.text = element_text(size=14),
        panel.background =element_rect(fill="gray99"),
        panel.border = element_rect(color="black", size=0.5,fill=NA),
        panel.grid.major.y = element_line(color="gray75", size=0.25),
        panel.grid.major.x = element_line(color="gray75", size=0.25))+ggtitle("Figure 2C")
```

```
#### missing value are due to the absence of the other genotypes
```

# Figure 3

```
t<-seq(0,1,0.005)
tx_ty<-do.call("rbind",sapply(X = t,FUN = function(X){
  list(data.frame(tx=X,ty=t))
  }))
mutate(tx_ty,
       Sp.Surv.Db_Zer=(1/2)*(2+tx*ty-tx-ty),
       Sp.Surv.Single=1+(-tx-ty)/2,
       Tot.Surv  = (Sp.Surv.Db_Zer+Sp.Surv.Single)/2,
       Tot.Surv.Trans= (1-tx)*(1-ty))->tx_ty

tx_ty%>%
ggplot()+
  geom_tile(aes(x=tx,y=ty,fill=Tot.Surv,color=Tot.Surv))+
  xlab("Transmission advantage tA")+
  ylab("Transmission advantage tB")+
  coord_cartesian(expand = FALSE,clip="off")+
  scale_fill_gradient2(low ="#D7ECDB",mid ="#286894" ,high ="#000000",midpoint = 0.5)+
  scale_color_gradient2(low ="#D7ECDB",mid ="#286894" ,high ="#000000",midpoint = 0.5)+
  geom_line(data=filter(tx_ty,near(Tot.Surv,0.30,tol=0.0001)),aes(x=tx,y=ty),color="gray20",linewidth=1
  geom_line(data=filter(tx_ty,near(Tot.Surv,0.5,tol=0.0001)),aes(x=tx,y=ty),color="gray30",linewidth=1.
  geom_line(data=filter(tx_ty,near(Tot.Surv,0.70,tol=0.0001)),aes(x=tx,y=ty),color="gray50",linewidth=1
  geom_line(data=filter(tx_ty,near(Tot.Surv,0.90,tol=0.0001)),aes(x=tx,y=ty),color="white",linewidth=1.
```

Figure 2C

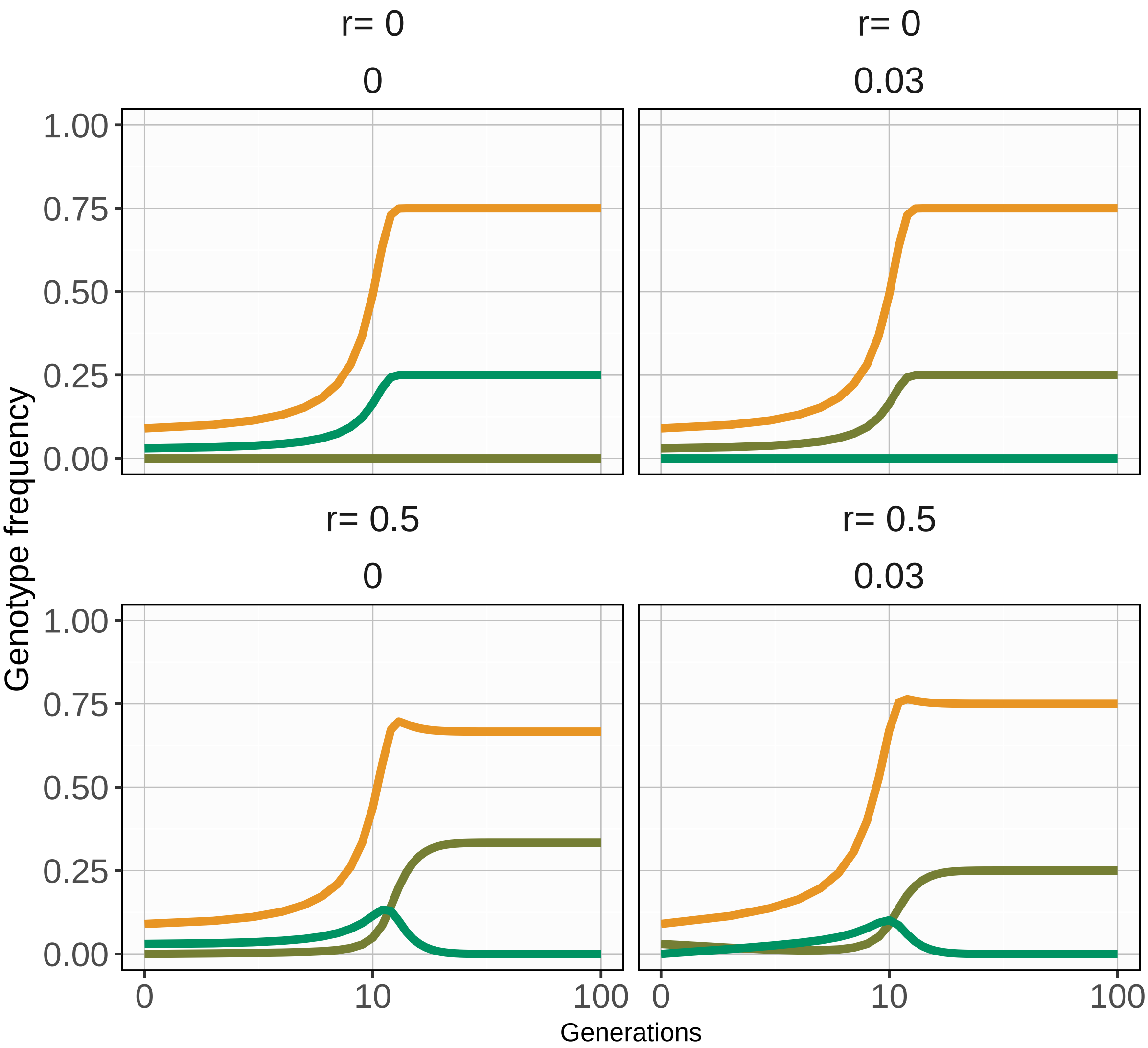

Figure 2: **Figure 2C.** Four distinct simulations in which a driver (wtfA+) makes an identical duplicate (wtfB+) in trans (on the homologous chromosome, left) or in cis (on the same chromosome, right). The transmission advantage (t) for each driver is 1. Simulations where the duplicate gene is absolutely linked (r=0, top) and unlinked (r=0.5, bottom) from the parent gene are shown. The starting frequency of the ancestral genotype (wtfA+ wtfB-, orange) is 0.1. The starting frequency of genotypes with a duplicated driver in cis (wtfA+ wtfB+, black) or in trans (wtfA- wtfB+, green) is 0.03. The remainder of each population is comprised of the wtfA- wtfB- genotype.

```
    theme(axis.title.x = element_text(size=10),
        legend.justification=c(1,1),
        axis.text.x =  element_text(size=13),
        axis.title.y = element_text(size=14),
        axis.text.y  =  element_text(size=13),
        legend.text = element_text(size=14),
        legend.title = element_text(size=10),
        legend.key.size = unit(x = 2,units = "line"),
        plot.title = element_text(size=11),
        panel.background =element_rect(fill="gray98"),
        panel.grid.major.y = element_blank(),
        panel.grid.major.x = element_blank())+
  ggtitle("Figure 3 B")
```

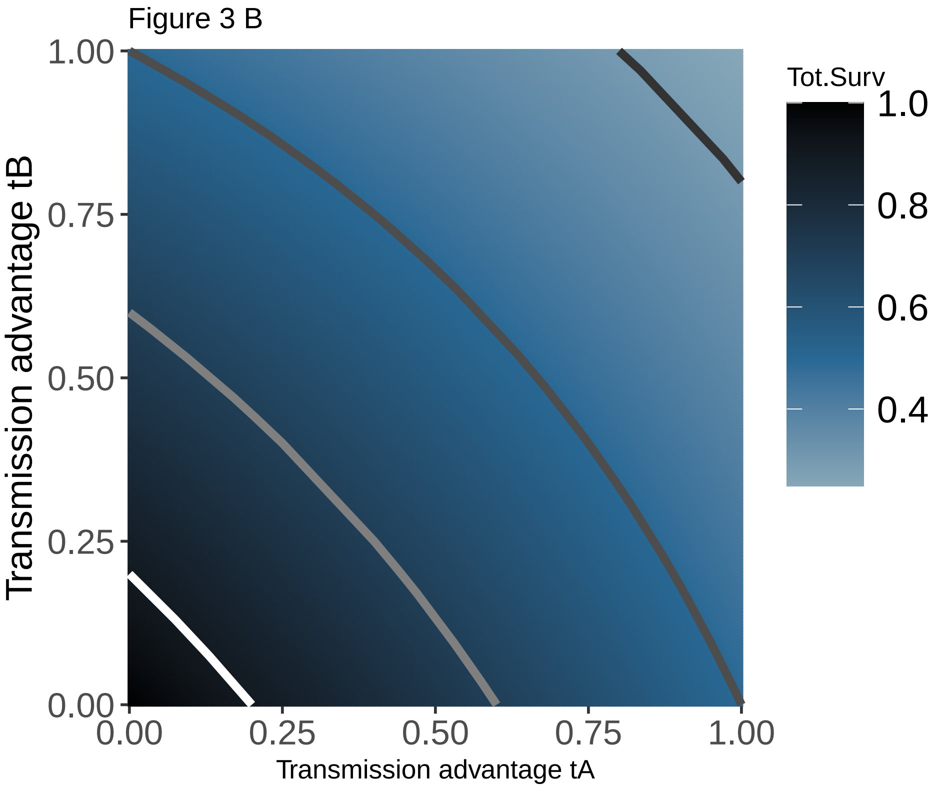

Figure 3: **Fig3B.** The fraction of spores, produced by diploids heterozygous for two unlinked drivers, expected to survive when considering varying drive strength.

## Figure 4

```
## [1] "Run simulations"
```

```
#### Picking the data to be plotted.
filter(SimulatedFrequencies,recombFreq==0)%>%
ggplot()+
  geom_line(aes(x=log10(Generations),
                y=F_xPlus_F_yPlus,group=id,color=x4_init),
            size=0.75,lty=1)+
    geom_line(aes(x=log10(Generations),
                y=F_xPlus_F_yMinus,group=id,color=x4_init),
            size=0.75,lty=5)+
facet_grid(~tx)+
  ylim(c(0,1))+
  ylab("Genotype Frequency")+
  xlab("Generations")+
      scale_x_continuous(breaks = seq(0,3,1),labels = c(0,10,100,1000),n.breaks = 4,limits=c(0,3))+
   scale_color_gradient(low ="#591145",high ="#F05934")+
    theme(axis.title.x = element_text(size=10),
          legend.justification=c(1,1),
          axis.text.x =  element_text(size=10),
          axis.title.y = element_text(size=10),
          axis.text.y  =  element_text(size=10),
          legend.text = element_text(size=10),
          legend.title = element_text(size=10),
          legend.key.size = unit(x = 2,units = "line"),
          panel.background =element_rect(fill="gray99"),
          panel.border = element_rect(color="black", size=0.5,fill=NA),
          panel.grid.major.y = element_line(color="gray75", size=0.25),
          panel.grid.major.x = element_line(color="gray75", size=0.25),
          strip.background = element_rect(fill="white"),
          strip.text = element_text(size=14))+
  ggtitle("Figure 4A")->figure.4a
figure.4a
```

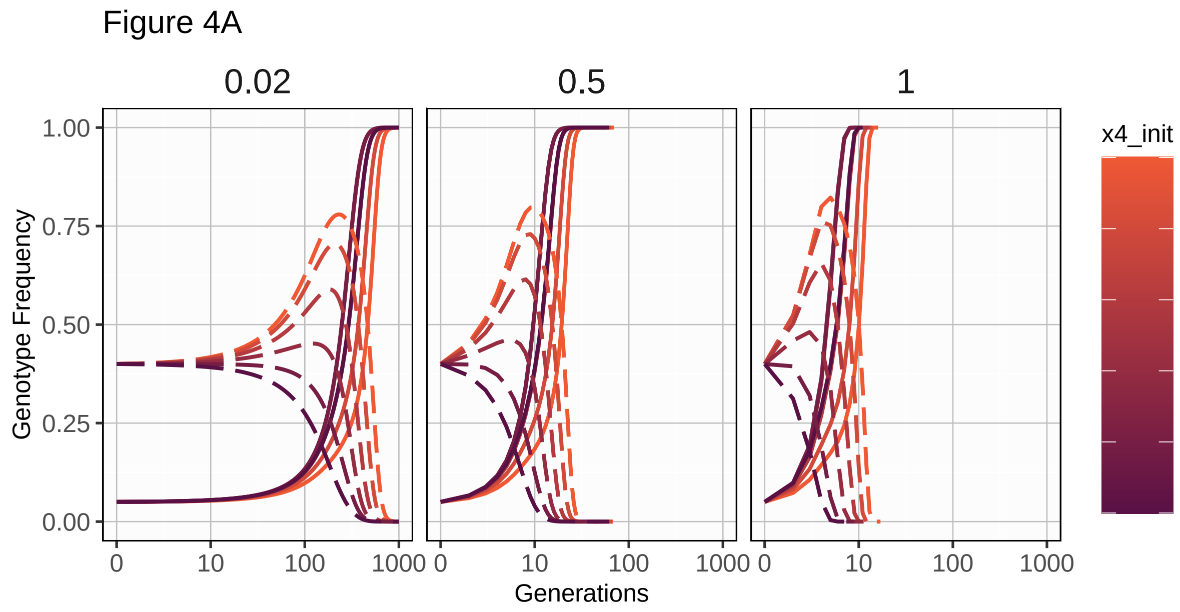

Figure 4: **Fig.4.** Change in driver genotype frequencies over time. The genotype frequencies of wtfA+ wtfB+ (solid, 0.05 initial frequency), wtfA+ wtfB- (dashed, 0.40 initial frequency) with varying wtfA- wtfB- initial frequencies with 0.1 steps. The remainder of each population is comprised of the wtfA- wtfB+ genotype. The genotype wtfA+ wtfB+ goes to fixation (See Supplemental materials) when present. Strong drivers (t=1, right) spread to fixation faster than weak drivers (t=0.2, left).

# Figure 5

```
#### Picking the data to be plotted.
filter(SimulatedFrequencies,recombFreq==0.5)%>%
ggplot()+
  geom_line(aes(x=log10(Generations),
                y=F_xPlus_F_yPlus,group=id,color=x4_init),
            size=0.75,lty=1)+
    geom_line(aes(x=log10(Generations),
                y=F_xPlus_F_yMinus,group=id,color=x4_init),
            size=0.75,lty=5)+
facet_grid(~tx)+
  ylim(c(0,1))+
    ylab("Genotype Frequency")+
  xlab("Generations")+
   scale_color_gradient(low ="#591145",high ="#F05934")+
        scale_x_continuous(breaks = seq(0,3,1),labels = c(0,10,100,1000),n.breaks = 4,limits=c(0,3))+
    theme(axis.title.x = element_text(size=10),
          legend.justification=c(1,1),
          axis.text.x =  element_text(size=10),
          axis.title.y = element_text(size=10),
          axis.text.y  =  element_text(size=10),
          legend.text = element_text(size=10),
          legend.title = element_text(size=10),
          legend.key.size = unit(x = 2,units = "line"),
          panel.background =element_rect(fill="gray99"),
          panel.border = element_rect(color="black", size=0.5,fill=NA),
          panel.grid.major.y = element_line(color="gray75", size=0.25),
          panel.grid.major.x = element_line(color="gray75", size=0.25),
          strip.background = element_rect(fill="white"),
          strip.text = element_text(size=14))+
  ggtitle("Figure 5A")->figure.5a
figure.5a
```

```
## [1] "Run simulations"
```

```
SimulatedFrequencies%>%
ggtern(
      aes(x=F_xPlus_F_yMinus+F_xMinus_F_yPlus,
                 y=F_xPlus_F_yPlus ,
                 z=F_xMinus_F_yMinus))+
  ylab("wtA+ wtfB+")+
  xlab("wtA+ wtfB- +
       wtA- wtfB+")+
  zlab("wtA- wtfB-") -> Plot

Plot+theme_classic(base_size = 20)+
  theme_showgrid_major()+
   theme_showgrid_minor()+
    theme_showarrows()->Plot
  ### Adding lines for the picked points
  Plot +  geom_path(data=SimulatedFrequencies,
                      aes(x=F_xPlus_F_yMinus+F_xMinus_F_yPlus,y=F_xPlus_F_yPlus,group=id),
                      lty=2,size=0.35,color="black",alpha=0.7) -> Plot
```

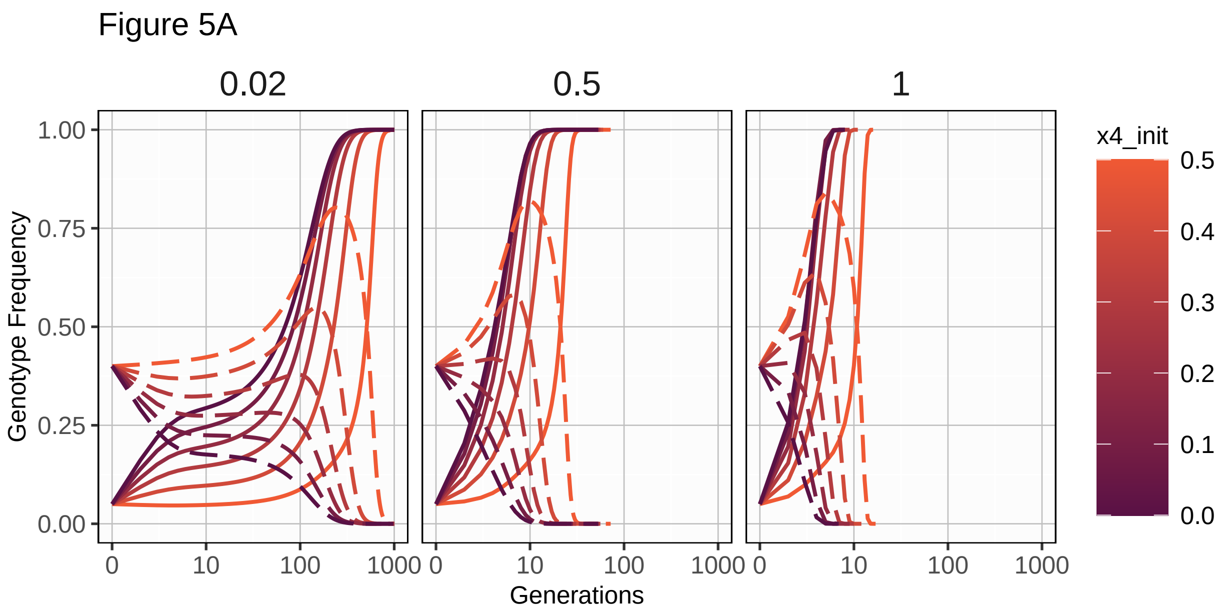

Figure 5: **Fig.5A** Changes in driver genotypes over time in the presence of recombination (r=0.5). The genotype frequencies of wtfA+ wtfB+ (solid, 0.1 initial frequency), wtfA+ wtfB- (dashed, 0.40 initial frequency) with varying wtfA- wtfB- initial frequencies with 0.1 steps. The remainder in each population is comprised of genotype wtfA- wtfB+. Strong drivers (t=1, right) spread to fixation faster than weak drivers (t=0.2, left).

```
  ### Adding arrows for the picked points
    Plot+ geom_path(data=dplyr::filter(SimulatedFrequencies,
                                       Generations>c(0L),
                                       Generations<c(6L)),
                    aes(x=F_xPlus_F_yMinus+F_xMinus_F_yPlus,y=F_xPlus_F_yPlus,group=id,color=tx),
                    size=0.45,alpha = 1,arrow=arrow(type = "closed",length = unit(.1,"cm")))->Plot
  Plot+
   # scale_fill_gradient2(low ="#D3BA07", mid ="#19827A" ,high ="#651D93", midpoint=color.midpoint)+
    scale_color_gradient2(low ="#D3BA07",mid ="#286894" ,high ="#B7002C", midpoint=0.5)+
    theme(axis.title.x = element_text(size=10),
          legend.justification=c(1,1),
          axis.text.x =  element_text(size=15),
          axis.title.y = element_text(size=10),
          axis.text.y  =  element_text(size=10),
          legend.text = element_text(size=10),
          legend.title = element_text(size=10),
          legend.key.size = unit(x = 2,units = "line"),
          panel.background =element_blank(),
          panel.grid.major = element_line(size=0.5,color ="gray20"),
         # tern.y.line   = element_line(color='black',size=1)
         )->Plot
      Plot+ ggtitle("Figure 5B")->Plot
Plot
```

```
gamete.frequencies<-create.gamete.frequencies(x1.limits = c(0,1),x2.limits = c(0,0),
                                                    x3.limits = c(0,0),x4.limits =c(0,1),step = 0.05)
t.values<-create.transmission.values(tx.limits =c(0,1),ty.limits=c(0,1),step=0.01)
```

# Figure 5B

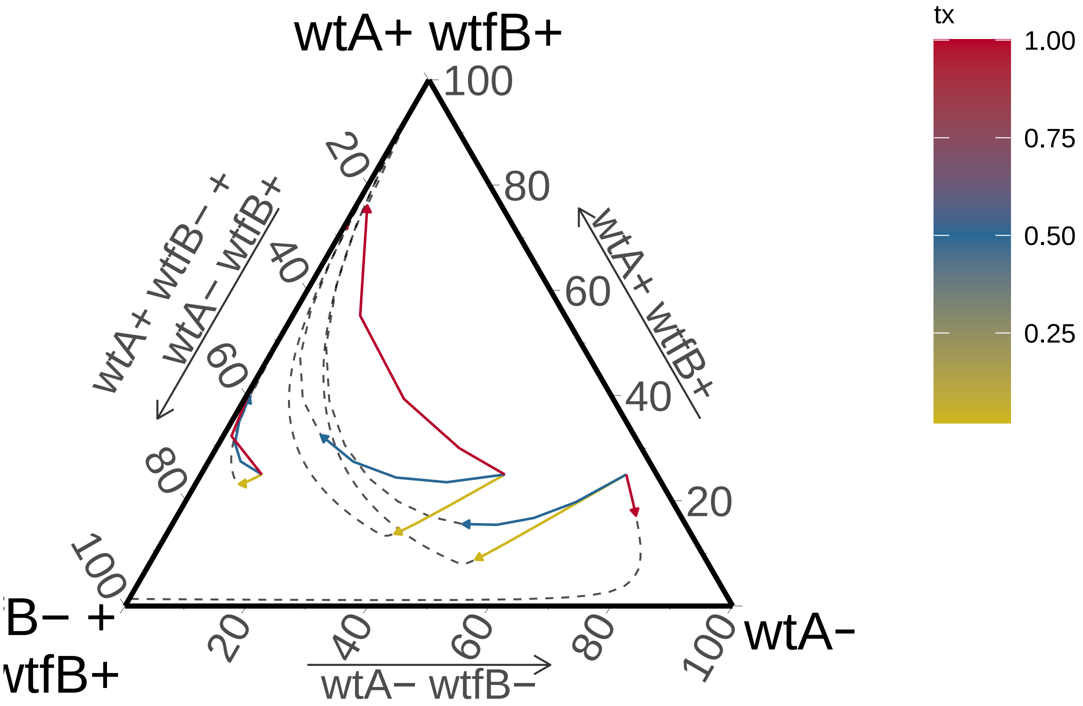

Figure 6: **Fig.5B.** The evolution of populations that initially lack the wtfA- wtfB+ genotype. The frequency of each genotype is shown on the three axes. The frequency wtfA- wtfB+ can be later generated by recombination. To read the frequency of the wtfA+ wtfB+ genotype, follow a horizontal line to the right axis. To read the frequency of the wtfA- wtfB- genotype, follow the diagonal down and to the left to the bottom axis. To read the combined frequency of the wtfA+ wtfB- and wtfA- wtfB+ genotypes, follow the diagonal up and to the left to the left axis. The two unlinked drivers have equal strength and three driver strengths (indicated by the different arrow colors as shown in the key) were considered. The point marked with an * represents the following frequencies: wtfA- wtfB- of 0.50, wtfA+ wtfB+ of 0.25, and wtfA+ wtfB- plus wtfA- wtfB+ of 0.25. The arrows depict allele frequency changes over four generations from that starting point and the dotted lines show subsequent frequency changes. Although the frequency of the wtfA+ wtfB+ genotype can initially decline (downward arrows), that genotype eventually spreads to fixation under all conditions illustrated.

```
dplyr::filter(t.values,tx%in%as.double(seq(0,1,0.1)))%>%
  dplyr::filter(ty%in%as.double(seq(0,1,0.1)))->t.values

set.seed(137)
## Merge frequencies and transmission advantage values.
SameDriverAdvantage.freqs<-merging.frequencies.and.t.values(
  gamete.frequencies=gamete.frequencies,
  transmision.values=dplyr::filter(t.values,tx==ty))
print("Run simulations")
```

```
## [1] "Run simulations"
```

```
cores<-2
cl <- makeCluster(cores) #make that many nodes
message(clusterEvalQ(cl, library("dplyr")))
SimulatedFrequencies.list.difRecfreq<-parLapply( X=1:cores,
                                      cl = cl,
                                      recombi.freq =seq(0,0.5,0.1),
                                      cores=cores,
                                      fun = Simulate.Evolved.PickedPoints.parallel,
                                      frequencies = SameDriverAdvantage.freqs,
                                      generations=10000,break.stable = TRUE,
                                      rounding.limit=13)

flush.console()
stopCluster(cl)
SimulatedFrequencies<-do.call("rbind",SimulatedFrequencies.list.difRecfreq)
```

```
dplyr::filter(SimulatedFrequencies,x3_init==0,x2_init==0,tx%in%c(0.2,1.0),recombFreq%in%c(0,0.5))%>%
  ggplot()+
  geom_path(aes(x=log10(Generations), y=F_xPlus_F_yPlus,color=x1_init,group=x1_init),
            size=.3,alpha = 1,arrow=arrow(type = "closed",length = unit(.05,"cm")))+
  facet_grid(recombFreq~tx)+
      scale_x_continuous(breaks = seq(0,3,1),labels = c(0,10,100,1000),n.breaks = 4,limits=c(0,3))+
    scale_color_gradient2(low ="#591145",mid ="#F05934" ,high ="#F7DBC4",midpoint = 0.5,limits=c(0,1))+
    theme(axis.title.x = element_text(size=10),
          legend.justification=c(1,1),
          axis.text.x =  element_text(size=10),
          axis.title.y = element_text(size=10),
          axis.text.y  =  element_text(size=10),
          legend.text = element_text(size=10),
          legend.title = element_text(size=10),
          panel.border = element_rect(color="black", size=0.15,fill=NA),
          legend.key.size = unit(x = 2,units = "line"),
          panel.background =element_rect(fill="gray99"),
          panel.grid.major.y = element_line(color="gray75", size=0.15),
          panel.grid.major.x = element_line(color="gray75", size=0.15),
          strip.background = element_rect(fill="white"),
          strip.text = element_text(size=14),
          legend.position="none",
        strip.placement = "outside")+
  xlab("Generations")+ylab("Frequency wtfA+ wtfB+")+
  ggtitle("Figure.5C-F")->Figure.5CF
Figure.5CF
```

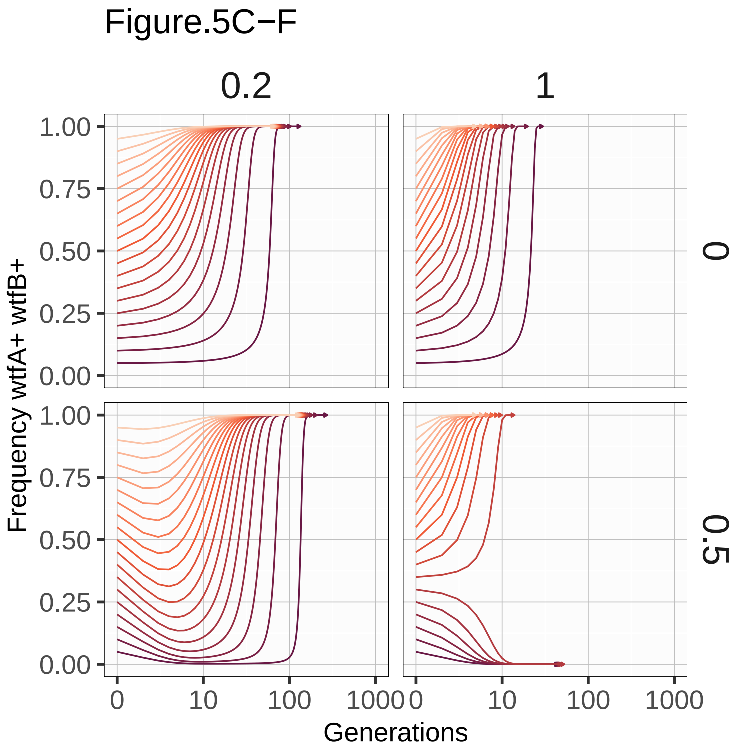

Figure 7: **Fig.5C-F.** Four simulated populations initially carry only two genotypes (wtfA+ wtfB+) and (wtfA- wtfB-). The initial frequencies for the genotype wtfA+ wtfB+ range from 0.05 to 0.95 with a 0.05 frequency step. Each simulation represents a population of two drivers that are absolutely linked (r=0, C and D) or unliked (r=0.5, E and F) and have a low (t=0.2, C and E) or high transmission bias (t=1, D and F). The spread of two drivers is delayed by recombination as the gametes carrying one driver can be destroyed by the alternate driver. Strong drivers can go extinct in the presence of recombination, particularly when the starting frequency of the wtfA+ wtfB+ genotype is low.

# Figure 6

```r
set.seed(394)
grid.freqs.ts<-expand.grid(0.1,0.40,0.20,0.30,seq(0.025,1,0.025),seq(0.025,1,0.025))
colnames(grid.freqs.ts)<-c("x1_init","x2_init","x3_init","x4_init" ,"tx" ,"ty")
dplyr::filter(grid.freqs.ts,tx!=ty)%>%sample_n(13)->DifferentDriverAdvantage.freqs_sampled

print("Run simulations")
```

```
## [1] "Run simulations"
```

```r
cores<-1
cl <- makeCluster(cores) #make that many nodes
message(clusterEvalQ(cl, library("dplyr"))) ## Loads package to each node
```

```
## c("dplyr", "stats", "graphics", "grDevices", "utils", "datasets", "methods", "base")
```

```r
set.seed(333)
Evolved.PickedPoints.list<-parLapply( X=1:cores,
                                      cl = cl,
                                      recombi.freq =seq(0,0.5,0.25),
                                      cores=cores,
                                      fun = Simulate.Evolved.PickedPoints.parallel,
                                      #frequencies = sample_n(DifferentDriverAdvantage.freqs_sampled,10
                                      frequencies = DifferentDriverAdvantage.freqs_sampled,
                                      generations=20000,break.stable = TRUE,
                                      rounding.limit=14)

flush.console()
stopCluster(cl)
SimulatedFrequencies<-do.call("rbind",Evolved.PickedPoints.list)
```

```r
SimulatedFrequencies%>%
ggplot()+
  geom_line(aes(x=log10(Generations),
                y=F_xPlus_F_yPlus,group=id,color=log10(tx/ty)),
            linewidth=0.75,lty=1)+
    geom_line(aes(x=log10(Generations),
                y=F_xPlus_F_yMinus,group=id,color=log10(tx/ty)),
            linewidth=0.75,lty=2)+
  ylim(c(0,1))+
    scale_color_gradient2(low ="#2D2D88",mid ="#CF407D" ,high ="#F3E305",midpoint = 0,
                          limits=c(log10(0.01),log10(1/0.01)))+
    facet_grid(~recombFreq)+
      scale_x_continuous(breaks = seq(0,3,1),labels = c(0,10,100,1000),n.breaks = 4,limits=c(0,3))+
    theme(axis.title.x = element_text(size=10),
          legend.justification=c(1,1),
          axis.text.x =  element_text(size=10),
          axis.title.y = element_text(size=10),
          axis.text.y  =  element_text(size=10),
          legend.text = element_text(size=10),
          legend.title = element_text(size=10),
          legend.key.size = unit(x = 2,units = "line"),
          panel.background =element_rect(fill="gray99"),
          panel.border = element_rect(color="black", size=0.5,fill=NA),
          panel.grid.major.y = element_line(color="gray75", size=0.25),
          panel.grid.major.x = element_line(color="gray75", size=0.25),
          strip.background = element_rect(fill="white"),
          strip.text = element_text(size=14))+
  ggtitle("Figure 6A")+ylab("Genotype frequencies")+
  xlab("Generations")->fig.6A
fig.6A
```

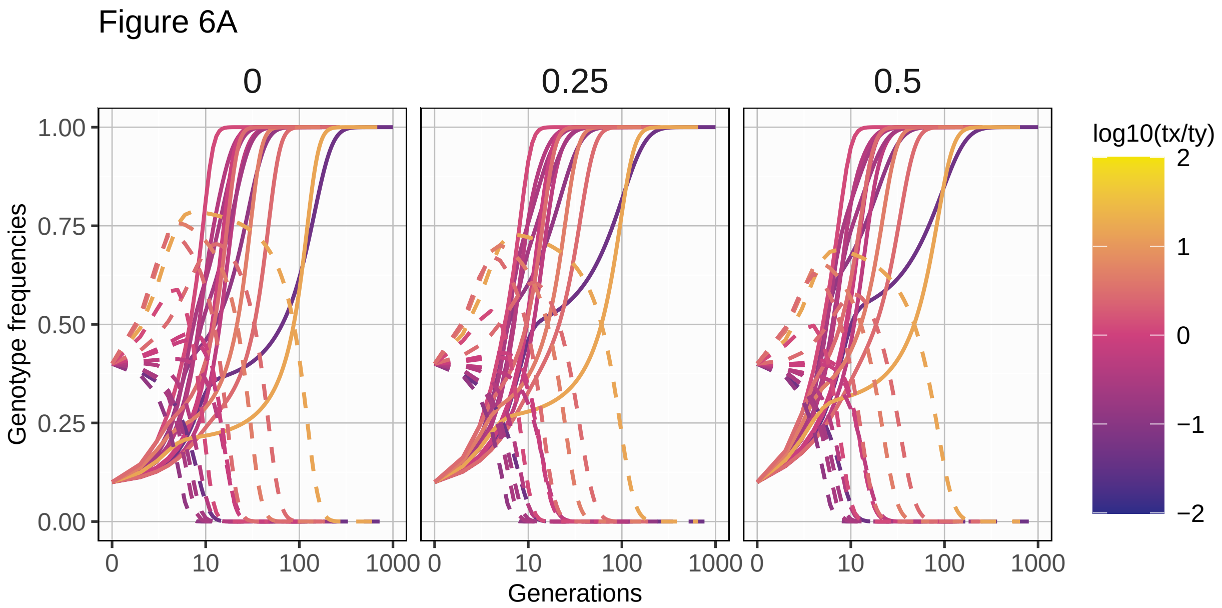

Figure 8: **Fig.6A.** Simulations with varying transmission advantages tA and tB for absolutely linked (r=0), mildly linked (r=0.25) or unlinked (r=0.5) loci. The genotype frequencies of wtfA+ wtfB+ (solid, 0.1), wtfA+ wtfB- (dashed, 0.40) with a wtfA- wtfB+ and wtfA- wtfB- initial frequencies with 0.2 and 0.3 respectively. The genotype wtfA+ wtfB+ goes to fixation when present (see suppl. for math proof). The genotype of the double driver can decrease in the presence of recombination, but it eventually spreads to fixation when present (see Supplemental material for mathematical proof).

```
set.seed(1678)
gamete.frequencies<-create.gamete.frequencies(x1.limits = c(0,1),x2.limits = c(0,1),
                                                    x3.limits = c(0,1),x4.limits =c(0,1),step = 0.01)
gamete.frequencies%>%filter( x1_init!=1,x2_init!=1,x3_init!=1,x4_init!=1)%>%
  filter((x1_init+x2_init)!=0,(x1_init+x2_init)!=0)%>%
  dplyr::sample_n(35355,replace=FALSE)->gamete.frequencies.samp

 create.transmission.values(tx.limits =c(0.01,1),ty.limits=c(0.01,1),step=0.01)%>%
   filter(tx!=ty)%>%
   dplyr::sample_n(5000,replace=FALSE)->t.values.samp

set.seed(137)
DifferentDriverAdvantage.freqs_sampled<-merging.frequencies.and.t.values(
  gamete.frequencies=gamete.frequencies.samp,
  transmision.values=dplyr::filter(t.values.samp,tx!=0,ty!=0))

print("Total possible conditions")
nrow(DifferentDriverAdvantage.freqs_sampled)
```

```
print("Run in parallel picked points 10000 distinct conditions")
cores<-20
cl <- makeCluster(cores) #make that many nodes
message(clusterEvalQ(cl, library("dplyr"))) ## Loads package to each node
set.seed(333)
Evolved.PickedPoints.list.r0<-parLapply( X=1:cores,
                                      cl = cl,
                                      recombi.freq =seq(0,0.5,0.1),
                                      cores=cores,
                                      fun = Simulate.Evolved.PickedPoints.parallel,
                                      frequencies = sample_n(DifferentDriverAdvantage.freqs_sampled,1000
                                      generations=10000,break.stable = TRUE,
                                      rounding.limit=13)
flush.console()
stopCluster(cl)
gc()
SimulatedFrequencies<-do.call("rbind",Evolved.PickedPoints.list.r0)

########Categorizing fate of drivers

filter(SimulatedFrequencies,
       x1_init!=1,x2_init!=1,x3_init!=1,x4_init!=1)->
  All.Present.Evolved

All.Present.Evolved%>%
  mutate(DriverXFreq=F_xPlus_F_yPlus+F_xPlus_F_yMinus,
         DriverYFreq=F_xPlus_F_yPlus+F_xMinus_F_yPlus)->All.Present.Evolved

All.Present.Evolved%>%
  mutate(DriverXFreq.round=round(DriverXFreq,digits = 13),
         DriverYFreq.round=round(DriverYFreq,digits = 13))->All.Present.Evolved

#### filter only the fixed ones
dplyr::filter(All.Present.Evolved,(DriverYFreq.round==1) | (DriverXFreq.round==1))->All.Present.Evolved

dplyr::select(All.Present.Evolved.Fixed,id)%>%unique()%>%sample_n(10000)%>%unlist()%>%unname()->sampled

FirstFixedRecord<-c()
for(k.ids in 1:10){
  #print(k.ids)
  chunk.ids<-sampled.ids[((k.ids-1)*1000+1):((k.ids*1)*1000)]
  filter(All.Present.Evolved.Fixed,id%in%chunk.ids)->All.Present.Evolved.id.Selected
  FirstFixedRecord.temp<-sapply(chunk.ids,function(X){
      selected.id<-All.Present.Evolved.id.Selected[All.Present.Evolved.id.Selected$id==X,]
      MinX<-selected.id[which(near(selected.id$DriverXFreq,1,tol = 10e-3))[1],"Generations"]
      MinY<-selected.id[which(near(selected.id$DriverYFreq,1,tol = 10e-3))[1],"Generations"]
      list(cbind(data.frame(min.Gen.X.Fixed=MinX,min.Gen.Y.Fixed=MinY),
                 selected.id[1,c("x1_init","x2_init","x3_init","x4_init","recombFreq","tx","ty")]))
    })
  FirstFixedRecord<-c(FirstFixedRecord,FirstFixedRecord.temp)
}
do.call("rbind",FirstFixedRecord)->FirstFixedRecord.DF
save(file = "./data/SimulatedFrequenciesFig6B.RData",SimulatedFrequencies,FirstFixedRecord.DF)
```

```
load(file = "./data/SimulatedFrequenciesFig6B.RData")

FirstFixedRecord.DF%>%
ggplot(aes(x=log10((x1_init+x2_init)/(x1_init+x3_init)),y=log10(min.Gen.X.Fixed/min.Gen.Y.Fixed),color=
  geom_point(size=0.3)+
    ylim(-2.1,2.1)+
  ylab("log10 (gens. to fix wtfA / gens. to fix wtfB)")+
  xlab("log10 (initial freq. wtfA / initial freq. wtfB)")+
    geom_smooth( color = "black",size=0.7)+
    scale_color_gradient2(low ="#2D2D88",mid ="#CF407D" ,high ="#F3E305",midpoint = 0,limits=c(log10(0.
    theme(axis.title.x = element_text(size=10),
          legend.justification=c(1,1),
          axis.text.x =  element_text(size=10),
          axis.title.y = element_text(size=10),
          axis.text.y  =  element_text(size=10),
          legend.text = element_text(size=10),
          legend.title = element_text(size=10),
        legend.key.size = unit(x = 2,units = "line"),
        panel.background =element_rect(fill="gray99"),
        panel.grid.major.y = element_line(color="gray85"),
        panel.grid.major.x = element_line(color="gray85"),
        strip.background = element_blank(),
        panel.border = element_rect(color="black", size=0.5,fill=NA),
        strip.text = element_text(size=14))+ggtitle("Figure 6B")->Figure6.B
Figure6.B
```

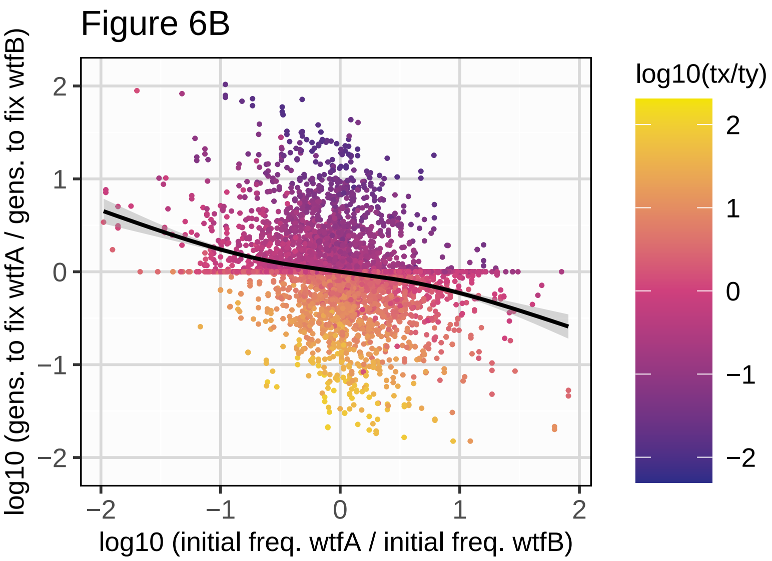

Figure 9: Fig.6B. 10,000 initial populations were simulated with multiple recombination frequencies ( r=0, 0.1, 0.2, 0.3, 0.4 and 0.5). The number of generations to fix a driver allele (i.e., wtfA) were compared to generations required to fix a second driver allele (i.e., wtfB). The stronger driver (larger t) tends to fix faster than a weaker driver, except in some cases when the weaker driver is initially more prevalent in a population. The black line is a local regression between X and Y axis. The shaded area is the standard error in the regression.

## Figure 7

```
set.seed(1678)
gamete.frequencies<-create.gamete.frequencies(x1.limits = c(0,0),x2.limits = c(0,1),
                                                    x3.limits = c(0,1),x4.limits =c(0,0),step = 0.01)
t.values<-create.transmission.values(tx.limits =c(0,1),ty.limits=c(0,1),step=0.001)


create.transmission.values(tx.limits =c(0,1),ty.limits=c(0,1),step=0.0025)%>%
  dplyr::sample_n(10000,replace=FALSE)->t.values


set.seed(137)
## Merge frequencies and transmission advantage values.
DifferentDriverAdvantage.freqs<-merging.frequencies.and.t.values(
  gamete.frequencies=gamete.frequencies,
  transmision.values=dplyr::filter(t.values,tx!=ty))
print("Run simulations")
cores<-20
cl <- makeCluster(cores) #make that many nodes
message(clusterEvalQ(cl, library("dplyr"))) ## Loads package to each node
Evolved.PickedPoints.list<-parLapply( X=1:cores,
                                      cl = cl,
                                      recombi.freq =0,
                                      cores=cores,
                                      fun = Simulate.Evolved.PickedPoints.parallel,
                                      frequencies = sample_n(DifferentDriverAdvantage.freqs,100000),
                                      generations=10000,break.stable = FALSE,
                                      rounding.limit=13)
flush.console()
stopCluster(cl)
Evolved.PickedPoints<-do.call("rbind",Evolved.PickedPoints.list)
save(file = "./data/SimulatedFrequenciesFig7.RData",Evolved.PickedPoints)
```

```
load(file = "./data/SimulatedFrequenciesFig7.RData")

print("Graph drawn from simulations")
```

```
## [1] "Graph drawn from simulations"
```

```
new.gamete.frequencies<-data.frame(x1_init=0, x2_init=seq(0,1,0.0025), x3_init=1-seq(0,1,0.0025), x4_ini
new.t.values<-data.frame(tx=sort(rep(seq(0,1,0.005),201)),ty=seq(0,1,0.005))
FrequenciesAntTValues<-merging.frequencies.and.t.values(
  gamete.frequencies=dplyr::filter(new.gamete.frequencies,x2_init!=1,x3_init!=1,x1_init==0,x4_init==0),
  transmision.values=dplyr::filter(new.t.values,tx!=0,ty!=0))
dplyr::filter(FrequenciesAntTValues,near(x2_init,(ty)/(tx+ty),tol=0.00000001),tx+ty!=0)->No.changeX2_X3

No.changeX2_X3.to.evolve%>%
  ggplot(aes(x=log10(tx/ty),y=x2_init))+
  geom_line(size=1)+
  geom_ribbon(ymin=-Inf, aes(ymax=x2_init), fill='#076844', alpha=0.2) +
  geom_ribbon(aes(ymin=x2_init), ymax=Inf, fill='#E89624', alpha=0.2) +
  xlab("log10(tx/ty)")+
  ylab("Min freq to spread wtfA2")+
  scale_color_gradient2(low ="#2D2D88",mid ="#CF407D" ,high ="#F3E305",midpoint =0,limits=c(log(1/200),
  scale_y_continuous(breaks = seq(0,1,0.25),labels =seq(0,1,0.25),n.breaks = 5)+
  coord_cartesian( ylim =c(0,1),
                    expand = FALSE) +
    theme(
      legend.justification=c(1,1),
        axis.text.x =  element_text(size=11),
        axis.text.y  =  element_text(size=11),
        axis.title.x = element_text(size=14),
        axis.title.y = element_text(size=14),
        legend.text = element_text(size=11),
        legend.title = element_text(size=11),
        legend.key.size = unit(x = 2,units = "line"),
        panel.background =element_rect(fill="gray99"),
        panel.grid.major.y = element_line(color="gray85"),
        panel.grid.major.x = element_line(color="gray85"),
        strip.background = element_blank(),
        panel.border = element_rect(color="black", size=0.5,fill=NA),
        strip.text = element_text(size=14),
        strip.placement = "outside")+
  ggtitle("Figure7A")->Figure.7A
Figure.7A
```

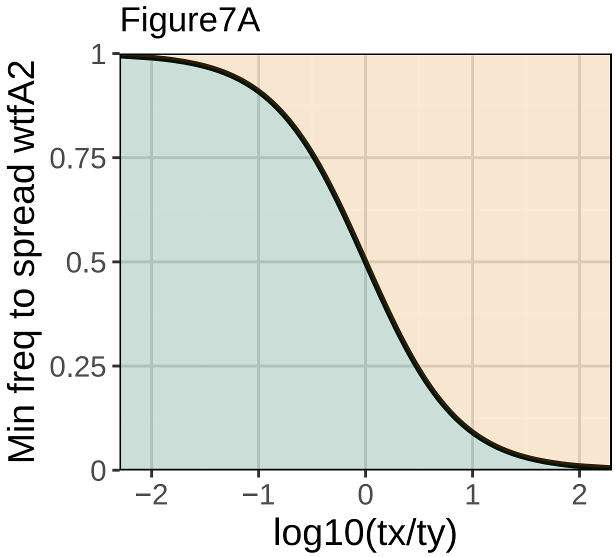

Figure 10: **Fig.7.** Populations with only wtfA1 and wtfA2 drivers are considered to represent two alternate driving alleles of varying relative strengths. The plotted line (black) represents a steady state where the driver frequencies remain constant. At points above the line, the wtfA1 spreads to fixation. At points below the line, the wtfA2 driver spreads. The weaker driver can spread to fixation if the weaker driver starts in excess.