

allFill_Kvary

December 20, 2017

1 Varying the carrying capacity across the landscape matrix

The results here relate to simulations that were run where the carrying capacity was varied across the landscape. The matrix was completely full from the outset and thus there are no colonization dynamics or founder effects. This was done at varying migration rates, as shown below. The carrying capacity was varied from 1000 at the "rural" end, declining linearly to a carrying capacity of 10 at the "urban" end. The other parameters in the simulations were as follows:

- Number of simulations: 1000
- Number of generations: 500
- Creation probability: NA
- Bottleneck proportion: NA
- pA and pB: 0.5
- Migration rate: varied from 0 to 1.0

```
In [2]: library(ggplot2)
library(grid)
library(data.table)
library(Rmisc)
library(dplyr)
```

Loading required package: lattice

Loading required package: plyr

data.table + dplyr code now lives in dtplyr.

Please library(dtplyr)!

Attaching package: dplyr

The following objects are masked from package:plyr:

```
arrange, count, desc, failwith, id, mutate, rename, summarise,
summarize
```

The following objects are masked from package:data.table:

```
between, first, last
```

The following objects are masked from package:stats:

```
filter, lag
```

The following objects are masked from package:base:

intersect, setdiff, setequal, union

```
In [3]: #Working directory for datasets varying migration rate and bottleneck proportion
setwd('/Users/jamessantangelo/Documents/Academia/Doctorate_PhD/Projects/SEC_Simulating.e
volutionary.clines/SEC_Data/summary-datasets/allFill_Kvary')

#Dataset with mean and proportions across all 1000 simulations
Kvary_Mig_Summary <- fread('20171106_Kvary-WithMig_SlopeSummaryGen_distRev.csv', header
= T)
Kvary_Mig_Summary$Mig_rate <- as.factor(as.character(Kvary_Mig_Summary$Mig_rate))

#Dataset with slope and P-value every generation every simulation
Kvary_Mig_Coefs <- fread('20171106_Kvary-WithMig_Coef_distRev.csv', header = T)
```

Read 19500000 rows and 6 (of 6) columns from 0.996 GB file in 00:00:12

```
In [152]: # Dataset used for plotting distribution of slopes in dominant allele clines. No
Migration. Generation 250
Coefs_pA_NoMig_Gen250 <- subset(Kvary_Mig_Coefs, id == "pA" & Mig_rate == 0 & Generation
== 250)

# Dataset used for plotting distribution of slopes HCN clines. No Migration. Generation
250
Coefs_Cyan_NoMig_Gen250 <- subset(Kvary_Mig_Coefs, id == "Cyan" & Mig_rate == 0 &
Generation == 250)

# Dataset used for plotting change in mean slope and proportions with increasing
generations
Kvary_Mig_Summary_Cyan <- subset(Kvary_Mig_Summary, id == "Cyan" & Mig_rate %in% c("0",
"0.001", "0.0025", "0.005", "0.01", "0.02", "0.035", "0.05"))
Kvary_Mig_Summary_Cyan$prop_sigPos[Kvary_Mig_Summary_Cyan$prop_sigPos == 1.000] <- 0.0

# Dataset used for plotting change in mean slope and proportions with varying migration
rate
Kvary_Mig_Summary_250 <- subset(Kvary_Mig_Summary, Generation == 250)
```

```
In [5]: str(Kvary_Mig_Coefs)
```

Classes data.table and 'data.frame': 19500000 obs. of 6 variables:

```
$ id      : chr  "Cyan" "Cyan" "Cyan" "Cyan" ...
$ Mig_rate : num  0 0 0 0 0 0 0 0 0 0 ...
$ Sim     : int  0 0 0 0 0 0 0 0 0 0 ...
$ Generation: int  1 2 3 4 5 6 7 8 9 10 ...
$ estimate : num  9.47e-05 1.30e-03 1.28e-03 6.23e-04 9.50e-04 ...
$ p.value  : num  0.00827 0.09663 0.12777 0.51763 0.37218 ...
- attr(*, ".internal.selfref")=<externalptr>
```

```
In [2]: #Themes used for plotting
ng1=theme(aspect.ratio=0.7,panel.background = element_blank(),
panel.grid.major = element_blank(),
panel.grid.minor = element_blank(),
panel.border=element_blank(),
axis.line.x = element_line(color="black",size=1),
axis.line.y = element_line(color="black",size=1),
axis.ticks=element_line(color="black"),
axis.text=element_text(color="black",size=15),
axis.title=element_text(color="black",size=1),
axis.title.y=element_text(vjust=2,face="bold",size=15),
axis.title.x=element_text(vjust=0.1,face="bold",size=15),
```

```

axis.text.x=element_text(size=13),
axis.text.y=element_text(size=13),
legend.position = "right", legend.direction="vertical",
legend.text=element_text(size=11), legend.key = element_rect(fill = "white"),
legend.title = element_text(size=13,face="bold"),legend.key.size = unit(0.5,
"cm"))

ng1.45=theme(aspect.ratio=0.7,panel.background = element_blank(),
panel.grid.major = element_blank(),
panel.grid.minor = element_blank(),
panel.border=element_blank(),
axis.line.x = element_line(color="black",size=1),
axis.line.y = element_line(color="black",size=1),
axis.ticks=element_line(color="black"),
axis.text=element_text(color="black",size=15),
axis.title=element_text(color="black",size=1),
axis.title.y=element_text(vjust=2,face="bold",size=15),
axis.title.x=element_text(vjust=0.1,face="bold",size=15),
axis.text.x=element_text(size=13,angle=45,hjust=1),
axis.text.y=element_text(size=13),
legend.position = "right", legend.direction="vertical",
legend.text=element_text(size=11), legend.key = element_rect(fill = "white"),
legend.title = element_text(size=13,face="bold"),legend.key.size = unit(0.5,
"cm"))

```

1.1 Distribution of slopes for allele and phenotype clines

For each simulation ($n = 1000$), I ran a linear model examining the change in either the frequency of either dominant allele with distance from the starting population or the frequency of HCN with distance. These models have different predictions: for clines in the dominant alleles at either locus, we expect on average half to go up and half to go down whereas for clines in HCN we expect the majority to go up (i.e. positive, loss of HCN across space) due to the duplicate recessivity underlying the phenotype.

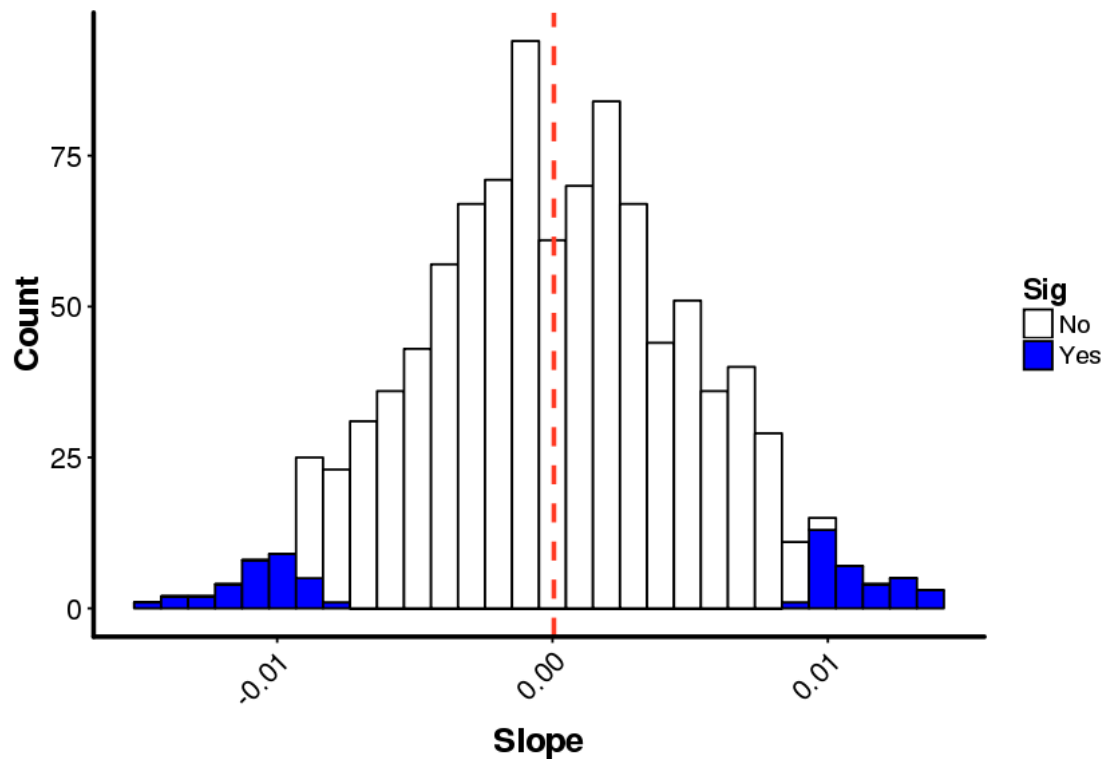
We can start to get at these predictions by looking at the distribution of the slopes from the different models. Note that the distribution of slopes is plotted when migration is 0. **First, we look at the distribution of slopes from models examining the change in the frequency of *Ac* with distance. As predicted, approximately half are positive and half are negative. Blue shading represents significant clines.**

```

In [25]: Coefs_pA_NoMig_Gen250 <- within(Coefs_pA_NoMig_Gen250, {
  Sig = ifelse(p.value < 0.05, "Yes", "No")
})
# Histogram of slopes for 'A' allele with no migration
HistSlopes_pA_NoMig_Gen250 <- ggplot(Coefs_pA_NoMig_Gen250, aes(x = estimate, fill =
Sig)) +
  ylab("Count") + xlab("Slope") +
  geom_histogram(color = "black") +
  geom_vline(data = Coefs_pA_NoMig_Gen250, aes(xintercept = mean(estimate)), size = 1,
colour = "#FF3721", linetype = "dashed") +
  scale_fill_manual(labels = c("No", "Yes"), values=c("white", "blue")) + ng1.45
HistSlopes_pA_NoMig_Gen250

```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

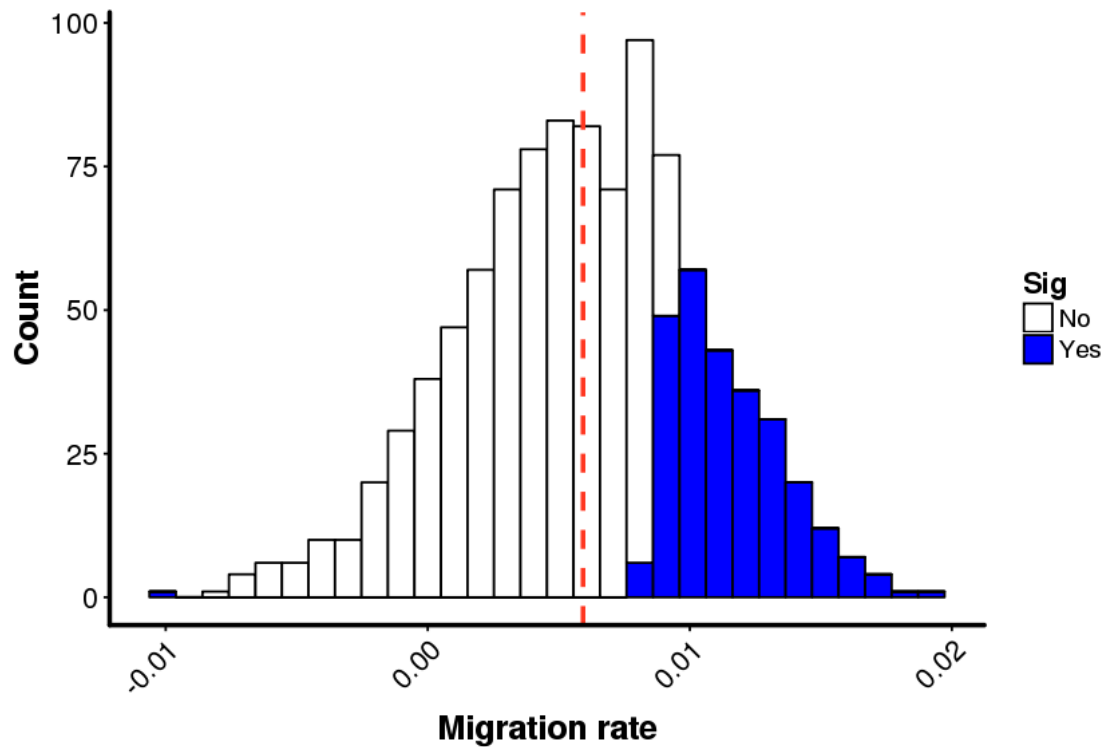


We find the same thing for the *Li* allele. Results not shown

For clines in the frequency of HCN, however, the majority of the slopes are negative and there are no significantly positive slopes. The mean slope is also negative

```
In [15]: Coefs_Cyan_NoMig_Gen250 <- within(Coefs_Cyan_NoMig_Gen250, {
  Sig = ifelse(p.value < 0.05, "Yes", "No")
})
# Histogram of slopes for 'Cyan' allele with no migration
HistSlopes_Cyan_NoMig_Gen250 <- ggplot(Coefs_Cyan_NoMig_Gen250, aes(x = estimate, fill =
Sig)) +
  ylab("Count") + xlab("Migration rate") +
  geom_histogram(color = "black") +
  geom_vline(data = Coefs_Cyan_NoMig_Gen250, aes(xintercept = mean(estimate)), size =
1, colour = "#FF3721", linetype = "dashed") +
  scale_fill_manual(labels = c("No", "Yes"), values=c("white", "blue")) + ng1.45
HistSlopes_Cyan_NoMig_Gen250
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



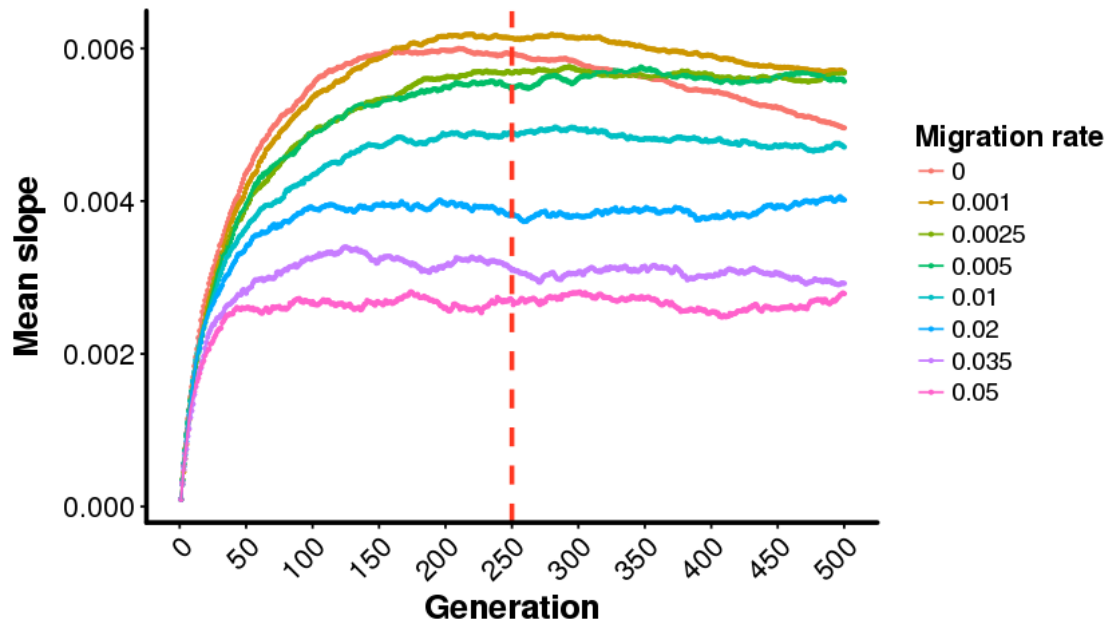
1.2 Mean slope with migration

Here I looked at how the mean slope of allele clines and HCN clines changed with migration.

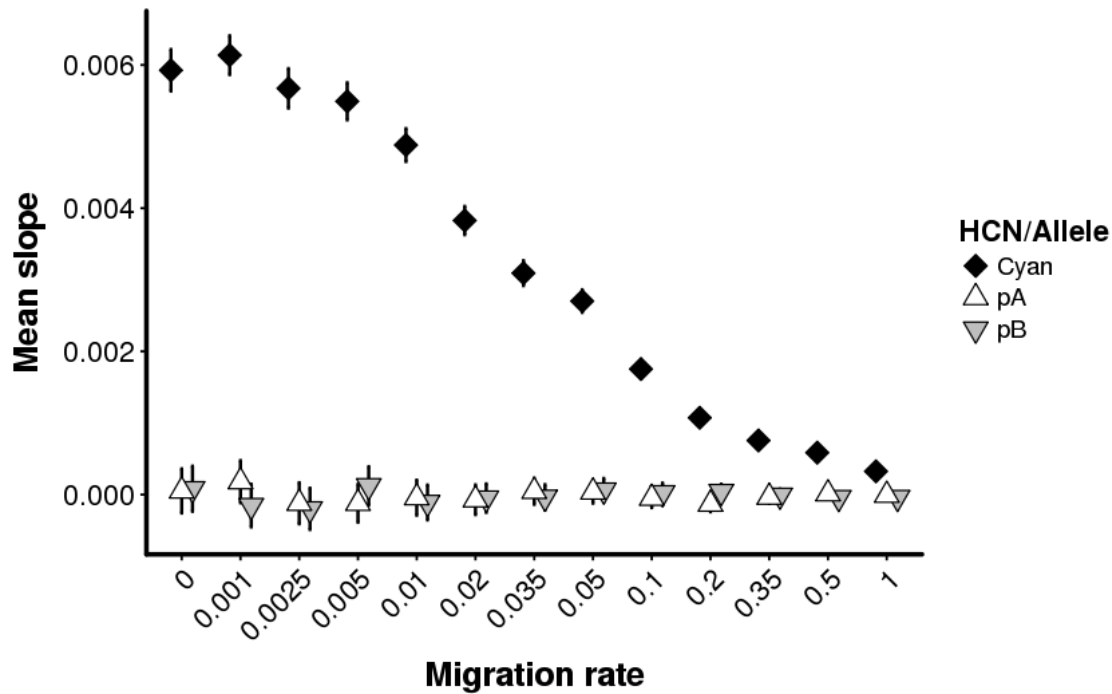
Migration had no effect on the mean slope of clines in *Ac* or *Li*. However, increasing migration resulted in HCN clines with slopes closer to 0. This is consistent with migration homogenizing alleles and flattening out clines.

```
In [153]: MeanSlope_Cyan_byGen <- ggplot(Kvary_Mig_Summary_Cyan, aes(x = Generation, y = mean,
group = Mig_rate, color = Mig_rate)) +
  ylab("Mean slope") + xlab("Generation") + geom_point(size = 0.5, aes(color =
Mig_rate)) +
  geom_line(size = 0.5, aes(color = Mig_rate)) +
  coord_cartesian(xlim = c(0, 500)) + scale_x_continuous(breaks = seq(from = 0, to =
500, by = 50)) +
  labs(color = "Migration rate") +
  geom_vline(data = Kvary_Mig_Summary_Cyan, xintercept = 250, size = 1, colour =
```

```
"#FF3721", linetype = "dashed") + ng1.45
MeanSlope_Cyan_byGen
```



```
In [40]: #Plot mean slope by migration rate, colored and shaped by phenotype or alleles
MeanSlope_Mig_Type <- ggplot(Kvary_Mig_Summary_250, aes(x = factor(Mig_rate), y = mean,
  shape = factor(id), fill = factor(id))) +
  geom_errorbar(aes(ymin = mean - ci_mean, ymax = mean + ci_mean), width=0.15, size =
0.7,
  position = position_dodge(width = 0.55)) +
  ylab("Mean slope") + xlab("Migration rate") + geom_point(size = 3.5, color =
"black",
  position = position_dodge(width = 0.55)) +
  scale_shape_manual(labels = c("Cyan", "pA", "pB"), values=c(23, 24, 25)) +
  scale_fill_manual(labels = c("Cyan", "pA", "pB"), values=c("black", "white",
"grey")) +
  labs(shape = "HCN/Allele", fill = "HCN/Allele") + ng1.45
MeanSlope_Mig_Type
```



1.3 Effects of migration on proportion of negative and positive clines

Here I look at how migration affects the proportion of significantly negative and positive clines in the *Ac* and *Li* alleles and in HCN clines.

Positive clines: Less HCN in urban environment (i.e. loss of HCN across space)

Negative clines: More HCN in urban environment (i.e. gain in HCN across space)

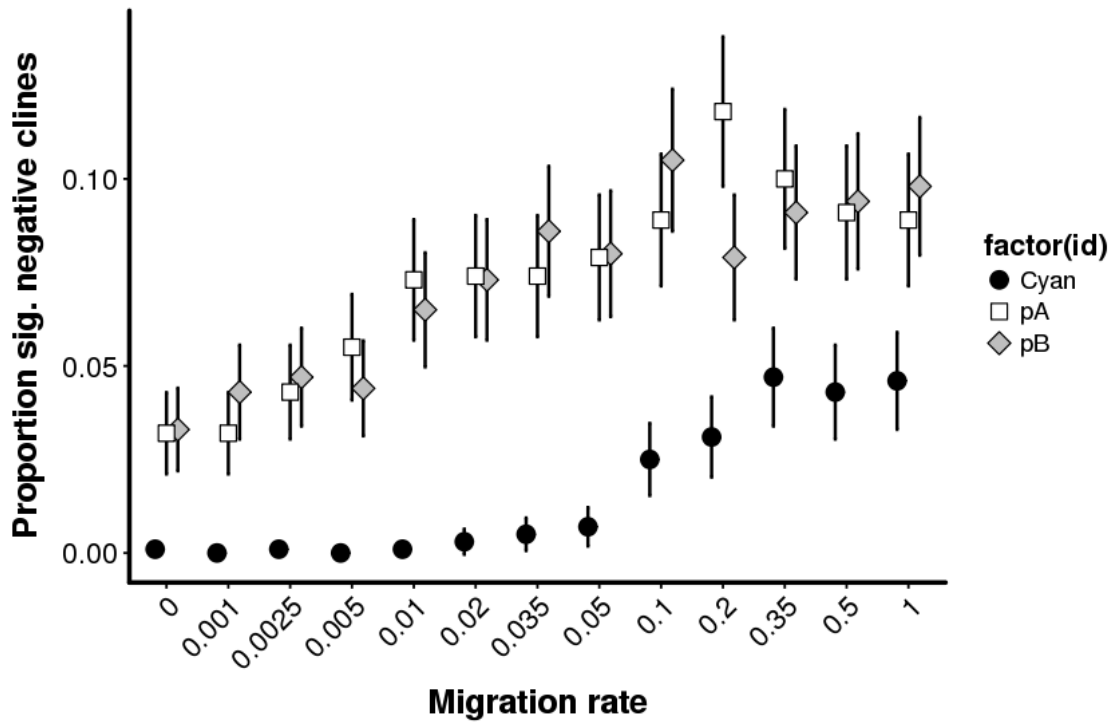
Increasing migration decreased the proportion of significantly positive clines in HCN, again consistent with the homogenizing effects of migration. In contrast, migration had little effect on the proportion of positive allele clines, although there seems to be a slight increase in the proportion with increasing migration.

```
In [158]: #Plot proportion of significantly negative clines by migration rate, colored and shaped
          by id
          NegClines_Mig_Type <- ggplot(Kvary_Mig_Summary_250, aes(x = factor(Mig_rate), y =
```

```

prop_sigNeg,
  shape = factor(id), fill = factor(id))) +
  geom_errorbar(aes(ymin = prop_sigNeg - ci_sigNeg, ymax = prop_sigNeg + ci_sigNeg),
width=0.15, size = 0.7,
  position = position_dodge(width = 0.55)) +
  ylab("Proportion sig. negative clines") + xlab("Migration rate") + geom_point(size =
3.5, color = "black",
  position = position_dodge(width = 0.55)) +
  scale_shape_manual(labels = c("Cyan", "pA", "pB"), values=c(21, 22, 23)) +
  scale_fill_manual(labels = c("Cyan", "pA", "pB"), values=c("black", "white",
"grey"))
NegClines_Mig_Type + ng1.45

```



```

In [60]: #Plot proportion of significantly positive clines by migration rate, colored and shaped
by id
PosClines_Mig_Type <- ggplot(Kvary_Mig_Summary_250, aes(x = factor(Mig_rate), y =
prop_sigPos,
  shape = factor(id), fill = factor(id))) +

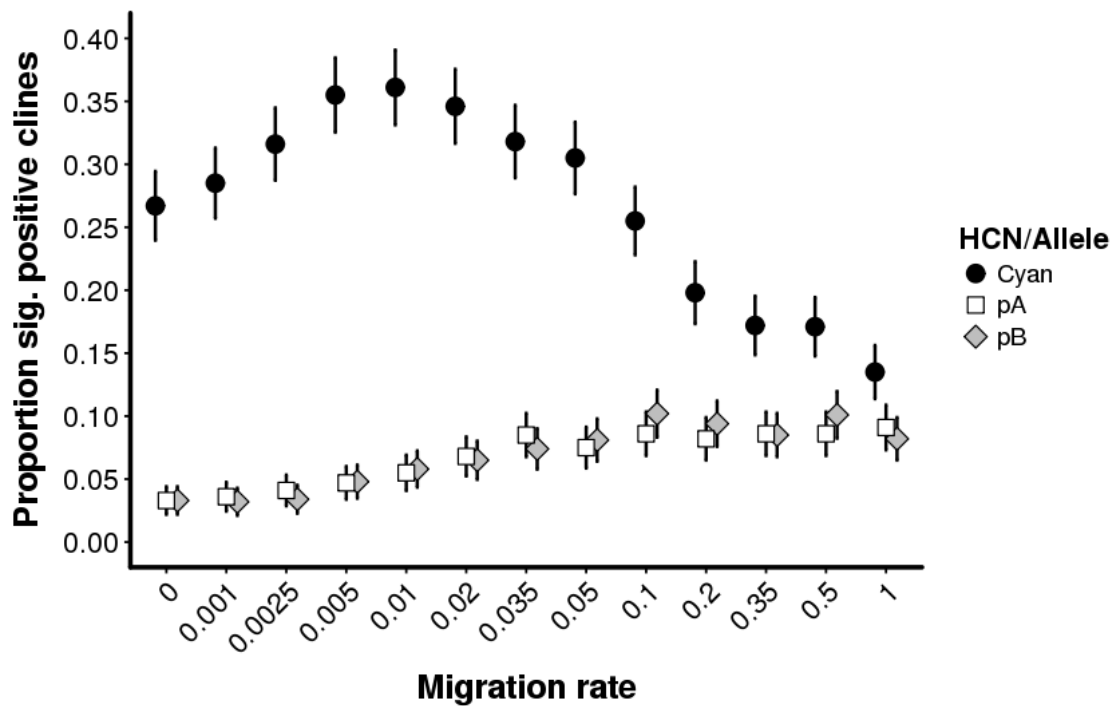
```



```

geom_errorbar(aes(ymin = prop_sigPos - ci_sigPos, ymax = prop_sigPos + ci_sigPos),
width=0.15, size = 0.7,
position = position_dodge(width = 0.55)) +
ylab("Proportion sig. positive clines") + xlab("Migration rate") + geom_point(size =
3.5, color = "black",
position = position_dodge(width = 0.55)) +
scale_shape_manual(labels = c("Cyan", "pA", "pB"), values=c(21, 22, 23)) +
scale_fill_manual(labels = c("Cyan", "pA", "pB"), values=c("black", "white",
"grey")) +
coord_cartesian(ylim = c(0, 0.4)) + scale_y_continuous(breaks = seq(from = 0, to =
0.4, by = 0.05)) +
labs(shape = "HCN/Allele", fill = "HCN/Allele") + ng1.45
PosClines_Mig_Type

```



```

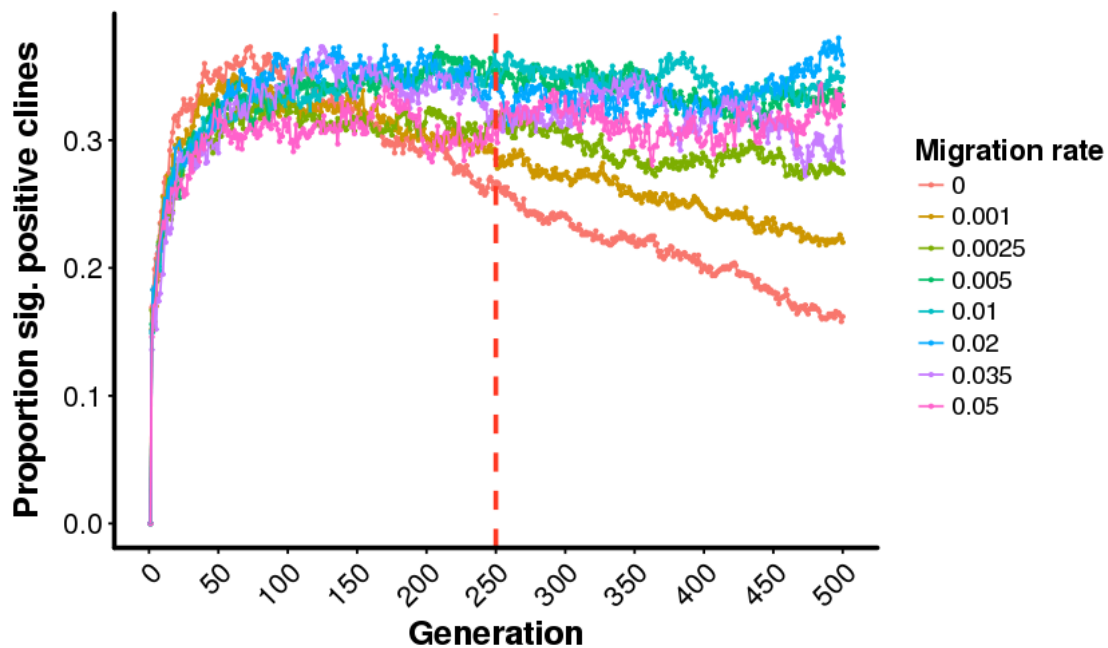
In [154]: PropSigPos_Cyan_byGen <- ggplot(Kvary_Mig_Summary_Cyan, aes(x = Generation, y =
prop_sigPos, group = Mig_rate, color = Mig_rate)) +
ylab("Proportion sig. positive clines") + xlab("Generation") + geom_point(size =
0.5, aes(color = Mig_rate)) +

```

```

geom_line(size = 0.5, aes(color = Mig_rate)) +
coord_cartesian(xlim = c(0, 500)) + scale_x_continuous(breaks = seq(from = 0, to =
500, by = 50)) +
labs(color = "Migration rate") +
geom_vline(data = Kvary_Mig_Summary_Cyan, xintercept = 250, size = 1, colour =
"#FF3721", linetype = "dashed") + ng1.45
PropSigPos_Cyan_byGen

```



```

In [157]: #Extra columns that will be used to create melting dataset for plotting proportion of
positive and negative slopes
SlopeSum_Melt_Mig <- dplyr::select(Kvary_Mig_Summary_250, id, Mig_rate, prop_sigPos,
ci_sigPos, prop_sigNeg,
ci_sigNeg)
SlopeSum_Melt_Mig <- SlopeSum_Melt_Mig %>%
filter(id == "Cyan")

#Melt dataframe
dm1 <- melt(SlopeSum_Melt_Mig[c("Mig_rate", "prop_sigPos", "ci_sigPos", "id")],

```

```

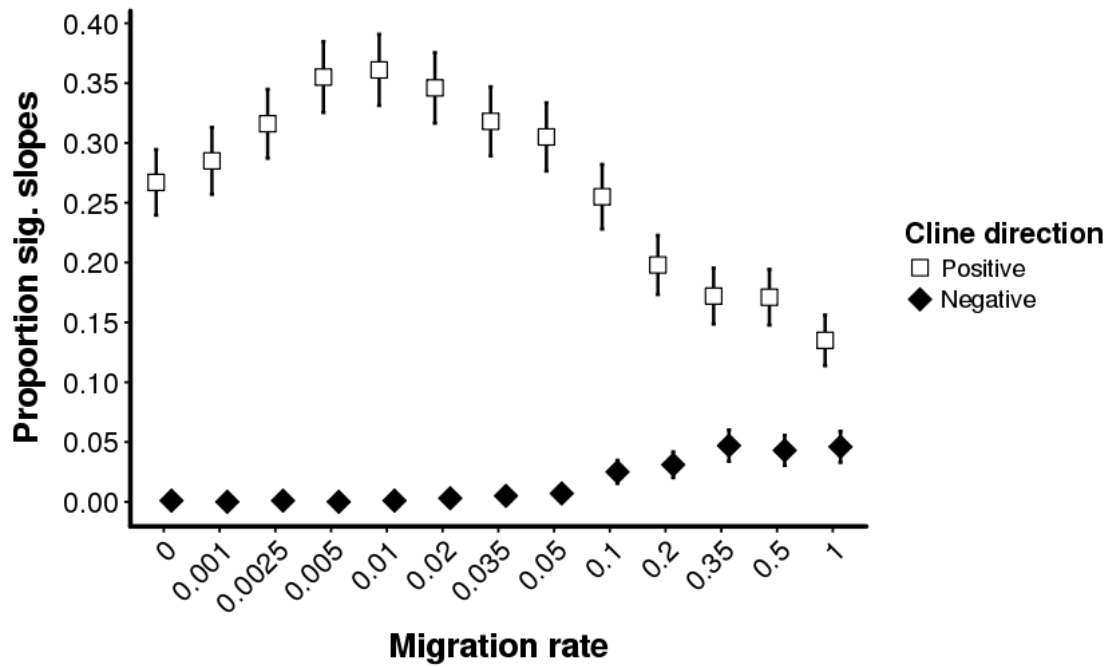
        id=c("Mig_rate", "ci_sigPos", "id"))
dm2 <- melt(SlopeSum_Melt_Mig[c("Mig_rate", "prop_sigNeg", "ci_sigNeg", "id")],
        id=c("Mig_rate", "ci_sigNeg", "id"))

#Rename columns
setnames(dm1, old = "ci_sigPos", new = "ci")
setnames(dm2, old = "ci_sigNeg", new = "ci")

#Merge melted dataframes
SlopeSum_Melt_Mig <- rbind(dm1, dm2)

In [156]: #Plot proportion of significant slopes by migration rate
PropSig_Mig <- ggplot(SlopeSum_Melt_Mig, aes(x = factor(Mig_rate), y = value,
        shape = variable, fill = variable)) +
        geom_errorbar(aes(ymin = value - ci, ymax = value + ci), width=0.15, size = 0.7,
        position = position_dodge(width = 0.55)) +
        ylab("Proportion sig. slopes") + xlab("Migration rate") + geom_point(size = 3.5,
        color = "black",
        position = position_dodge(width = 0.55)) +
        scale_shape_manual(labels = c("Positive", "Negative"),values=c(22, 23)) +
        scale_fill_manual(labels = c("Positive", "Negative"),values=c("white", "black")) +
        scale_y_continuous(breaks = seq(from = 0, to = 1.0, by = 0.05)) +
        labs(shape = 'Cline direction', fill = 'Cline direction')
PropSig_Mig + ng1.45

```



```
In [31]: path = "/scratch/research/projects/trifolium/SEC_Simulation.Evolutionary.Clines/SEC_Sync
/SEC_Figures/Drift.Migration/Mig_Bot_Vary"
```

```
# ggsave("PropSig_Mig.pdf", plot = PropSig_Mig, device = "pdf", width = 6.0, height =
6.0, path = path, dpi = 600)
# ggsave("HistSlopes_pA_NoMig.pdf", plot = HistSlopes_pA_NoMig, device = "pdf", width =
6.0, height = 6.0, path = path, dpi = 600)
# ggsave("HistSlopes_HCN_NoMig.pdf", plot = HistSlopes_Cyan_NoMig, device = "pdf", width
= 6.0, height = 6.0, path = path, dpi = 600)
# ggsave("MeanSlope_Mig_Type.pdf", plot = MeanSlope_Mig_Type, device = "pdf", width =
6.0, height = 6.0, path = path, dpi = 600)

ggsave("MeanSlope_Mig_Type.pdf", plot = MeanSlope_Mig_Type, device = "pdf", width = 6.0,
height = 6.0, path = path, dpi = 600)
ggsave("PropSig_Mig.pdf", plot = PropSig_Mig, device = "pdf", width = 6.0, height = 6.0,
path = path, dpi = 600)
```