

Supplementary Online Material for: Stable *Heliconius* butterfly hybrid zones are correlated with a local rainfall peak at the edge of the Amazon basin

Neil Rosser^{1,2*}, Kanchon Dasmahapatra² & James Mallet^{1,3}.

1. Department of Genetics, Evolution and Environment, University College London, Gower Street, London WC1E 6BT, UK.

2. Department of Biology, University of York, Wentworth Way, York YO10 5DD, UK

3. Department of Organismic and Evolutionary Biology, Harvard University, 16 Divinity Avenue, Cambridge, MA 02138, USA. Telephone: +(1)(617) 496 5350 Email: jmallet@oeb.harvard.edu

*Author for correspondence. Email: neil.rosser@york.ac.uk

S1: Genetics of colour patterns

Specimens were genotyped using their colour pattern phenotypes. The Mendelian genetics of colour patterns in the Amazonian and Andean races of *H. erato* and *H. melpomene* are described in more detail in Mallet (1989) and Mallet et al. (1990) and are merely outlined here.

In *H. erato*, three loci determine the major colour pattern elements. The *D* locus (referred to as D^{Ry}/d^{rY} in Mallet (1989)) determines the presence/absence of proximal forewing patch known as "dennis" as well as hindwing rays, and the yellow versus red colour of the forewing band. *DD* genotypes have dennis, rays and a yellow forewing band, whereas *dd* genotypes have no dennis or rays and a red forewing band. Heterozygotes are recognisable because the red colour elements are expressed; *Dd* heterozygotes have dennis and rays, with a red forewing band. A more or less dominant locus *Sd* determines the width of the forewing band. Recessive homozygotes (*sdsd*) have a broad forewing band, and dominant homozygotes (*SdSd*) and heterozygotes (*Sdsd*) have a narrow forewing band (Mallet 1989). The *Sd* and *Cr* loci interact to produce the yellow hindwing bar of the postman pattern race. Only in *sdsdcr cr* genotypes is the hindwing bar completely expressed. In *Sd- cr cr* individuals a weak hindwing bar is expressed (with a narrow forewing band), whereas in *sdsd Cr-* individuals only the tips of the hindwing bar are expressed (with a broad forewing band). *Sd- Cr-* individuals have no hindwing bar expression except for occasional "shadows" on the undersides of *Cr cr* genotypes.

In *H. melpomene*, four dominant loci determine colour pattern differences. The *D* locus (called D^R/d in Turner (1972) and Mallet (1989)) determines the presence (*D-*) or absence (*dd*) of the dennis and rays. The *Yb* locus determines the presence (*ybyb*) or absence (*Yb-*) of the yellow hind-wing bar. The *N* locus (N^N/N^B in Mallet (1989)), is putatively codominant (Turner 1972), but heterozygotes *Nn* are often hard to distinguish from *NN* homozygotes (Mallet 1989) and so *N* is treated as dominant here. The *N* locus interacts with another locus, *B* to determine the shape and colour of the forewing band; the *N* allele codes for a yellow forewing band, as well as narrowing any red forewing band present. The *B* allele codes for a red forewing band. *bbN-* individuals have a yellow forewing bar, whereas *B-nn* shows full red forewing bar expression. *B-N-* genotypes show a yellow bar, with a narrow red outer rim. Finally, double recessive homozygotes (*bbnn*) have melanic forewings with no band. The *N* and *Yb* loci are tightly linked (with recombination rate, $c \approx 0.005$), and the *B* and *D* loci are thought to be moderately ($c \approx 0.135$) or tightly linked (Sheppard et al. 1985; Mallet 1989; Baxter et al. 2010).

The *H. melpomene N* locus can be difficult to score due to epistatic effects of another locus, *M* (Mallet 1989). *M* has never been mapped but is almost certainly linked to the *B-D* region (Simon Baxter, pers. comm.), and its existence may explain curious features of (apparent) disequilibria measured among mapped loci (see results).

Recent studies confirm that the *N/Yb/Cr* and *B/D* loci are gene regulatory elements in homologous regions of the genomes of *H. erato* and *H. melpomene* (Joron et al. 2006, 2011; Reed et al. 2011). A third locus has now been characterized, which shows similar identity between the *H. erato Sd* and the *H. melpomene Ac* (Martin et al. 2012). However, the *Ac* locus appears to be involved here only in *erato*, and not in the *melpomene* hybrid zone. In spite of the homologous genomic locations of these three genes in the two species, one of the mimetic species, *H. melpomene*, has almost certainly adapted to and recreated the other's colour patterns, as the two

species are sufficiently divergent to prevent any possible hybridization and the allelic variants at these loci show no homologies other than genomic location (Supple et al. 2013).

S2: R code for iterative solution of multilocus clines and for cline fitting via likelihood

The model described in the methods section was iterated for 200 generations in a one-dimensional series of 200 demes, each connected to others by migration to obtain appropriate theoretical cline shapes for fitting to the data. This was sufficient as clines stabilized after a few tens of generations at the selection pressures used, and showed no discontinuities; see Fig. 3, top panels). Starting conditions did not affect cline shape; we initiated the model with "secondary contact" between the two pure genotypes, which meet in the middle of the series of demes in an initially abrupt step (see also Mallet and Barton (1989a). In each generation, the order of processes was migration, selection and then reproduction. The proportions of genotypes migrating to other demes was modelled as a binomial function to yield effectively continuous Gaussian migration across the discrete array of demes, as in Mallet and Barton (1989a). Because we know that selection in this system is liable to be high (Mallet and Barton 1989b), we implemented frequency dependent selection of $2s = 0.6$ on each locus.

For each cline, we optimised its position on the transect while simultaneously stretching / compressing the cline width to fit it to a kilometre scale, until we found the maximum likelihood position and width. For the codominant locus *erato D* we estimated the likelihood of the observed number of *D* alleles. Log likelihoods were calculated using the formula:

$$\log L = \sum i \ln p + (2n - i) \ln(1 - p) \quad \text{Equation 1a}$$

where p is the frequency of the dominant allele predicted from the theoretical cline, i is observed number of *D* alleles, and n is sample size of genotypes. Dominant loci *Cr* and *Sd* in *H. erato* as well as all loci in *H. melpomene* were fitted using the expected Hardy-Weinberg phenotype frequency obtained from the theoretical cline:

$$\log L = \sum (n - i) \ln(1 - p)^2 + i \ln(p^2 + 2p(1 - p)) \quad \text{Equation 1b}$$

where i is the observed number of dominant phenotypes. The assumption of Hardy-Weinberg equilibrium was tested and found to hold for the *D* locus in *H. melpomene* (Mallet et al. 1990: p. 926).

S3. Implementation of interaction of iterated warning colour cline with a density trough and exogenous cline

The dominant cline (*A*) was subject to a frequency-dependent selection of $2s = 0.5$, with a migration rate of $\sigma^2 = 30$. In the absence of any other force, the cline moves across the series of demes at a constant rate. To model a density trough across the demes, we used a cosine function in the interval 0 to 2π radians to generate a dip in which density was reduced maximally to 50% relative to the rest of the demes. To model an unlinked cline tied to the environment (i.e. one maintained by exogenous selection), we iterated a dominant allele *E*, favoured on one side of the demic array, relative to allele *e*. The exogenous gradient in selection was modelled using a logistic function:

$$f(x) = \frac{e^{b(x-a)}}{1+e^{b(x-a)}} \quad \text{Equation 2}$$

where a is the centre of the selection gradient, b is the slope and x is the deme number. Selection on locus E across the series of demes was given by a sigmoid with a centre of 50.5 and a slope - 1, with the asymptote rescaled to $s_E = 0.5$. The fitness of EE genotypes was $1 - s_E f(x)$, that of ee was $1 - s_E (1 - f(x))$; heterozygotes Ee had intermediate fitness.

References

- Baxter, S. W., N. J. Nadeau, L. S. Maroja, P. Wilkinson, B. A. Counterman, A. Dawson, M. Beltran, S. Perez-Espona, N. Chamberlain, L. Ferguson, R. Clark, C. Davidson, R. Glithero, J. Mallet, W. O. McMillan, M. R. Kronforst, M. Joron, R. H. ffrench-Constant, and C. D. Jiggins. 2010. Genomic hotspots for adaptation: the population genetics of Müllerian mimicry in the *Heliconius melpomene* Clade. *PLoS Genet.* 6:e1000794.
- Joron, M., L. Frezal, R. T. Jones, N. L. Chamberlain, S. F. Lee, C. R. Haag, A. Whibley, M. Becuwe, S. W. Baxter, L. Ferguson, P. A. Wilkinson, C. Salazar, C. Davidson, R. Clark, M. A. Quail, H. Beasley, R. Glithero, C. Lloyd, S. Sims, M. C. Jones, J. Rogers, C. D. Jiggins, and R. H. ffrench-Constant. 2011. Chromosomal rearrangements maintain a polymorphic supergene controlling butterfly mimicry. *Nature* 477:203–206.
- Joron, M., R. Papa, M. Beltrán, N. Chamberlain, J. Mavárez, S. Baxter, M. Abanto, E. Bermingham, S. J. Humphray, J. Rogers, H. Beasley, K. Barlow, R. H. ffrench-Constant, J. Mallet, W. O. McMillan, and C. D. Jiggins. 2006. A conserved supergene locus controls colour pattern diversity in *Heliconius* Butterflies. *PLoS Biol* 4:e303.
- Mallet, J. 1989. The genetics of warning colour in Peruvian hybrid zones of *Heliconius erato* and *H. melpomene*. *Proc. R. Soc. B Biol. Sci.* 236:163–185.
- Mallet, J., and N. Barton. 1989a. Inference from clines stabilized by frequency-dependent selection. *Genetics* 122:967–976.
- Mallet, J., and N. H. Barton. 1989b. Strong natural selection in a warning-color hybrid zone. *Evolution* 43:421–431.
- Mallet, J., N. H. Barton, G. Lamas, J. Santisteban, M. Muedas, and H. Eeley. 1990. Estimates of selection and gene flow from measures of cline width and linkage disequilibrium in *Heliconius* hybrid zones. *Genetics* 124:921–936.
- Martin, A., R. Papa, N. J. Nadeau, R. I. Hill, B. A. Counterman, G. Halder, C. D. Jiggins, M. R. Kronforst, A. D. Long, W. O. McMillan, and R. D. Reed. 2012. Diversification of complex butterfly wing patterns by repeated regulatory evolution of a *Wnt* ligand. *Proc. Natl. Acad. Sci. U. S. A.* 109:12632–12637.
- Reed, R. D., R. Papa, A. Martin, H. M. Hines, B. A. Counterman, C. Pardo-Diaz, C. D. Jiggins, N. L. Chamberlain, M. R. Kronforst, R. Chen, G. Halder, H. F. Nijhout, and W. O. McMillan.

2011. *optix* drives the repeated convergent evolution of butterfly wing pattern mimicry. *Science* 333:1137–1141.

Sheppard, P. M., J. R. G. Turner, K. S. Brown, W. W. Benson, and M. C. Singer. 1985. Genetics and the evolution of Müllerian mimicry in *Heliconius* butterflies. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.* 308:433–610.

Supple, M. A., H. M. Hines, K. K. Dasmahapatra, J. J. Lewis, D. M. Nielsen, C. Lavoie, D. A. Ray, C. Salazar, W. O. McMillan, and B. A. Counterman. 2013. Genomic architecture of adaptive color pattern divergence and convergence in *Heliconius* butterflies. *Genome Res.* 23:1248–1257.

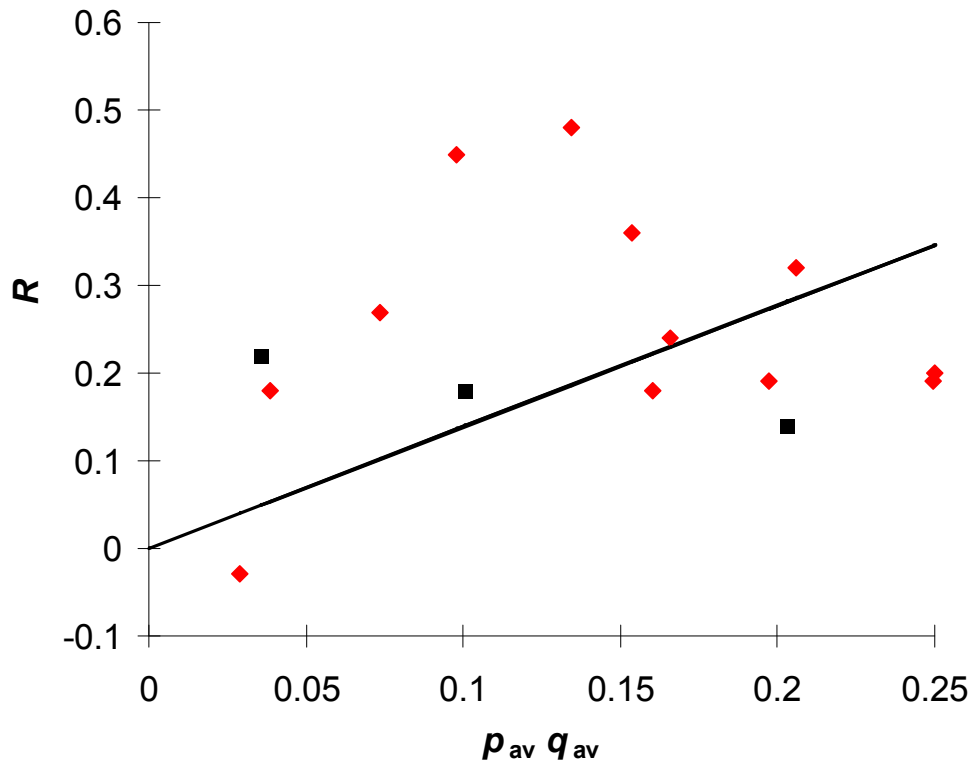


Figure S1. Least squares regression of linkage disequilibrium (R) for *H. erato* estimated from all three loci against the product of average allele frequencies, using only sample sizes greater than 20. 1986 values shown in red, 2011 values shown in black. The intercept was constrained to zero, with the slope then estimated as 1.38 (0.82-1.95), giving $R_{max} = 0.35$ (0.20, 0.49) when $p_{av}q_{av} = 0.25$ (the middle of the hybrid zone).

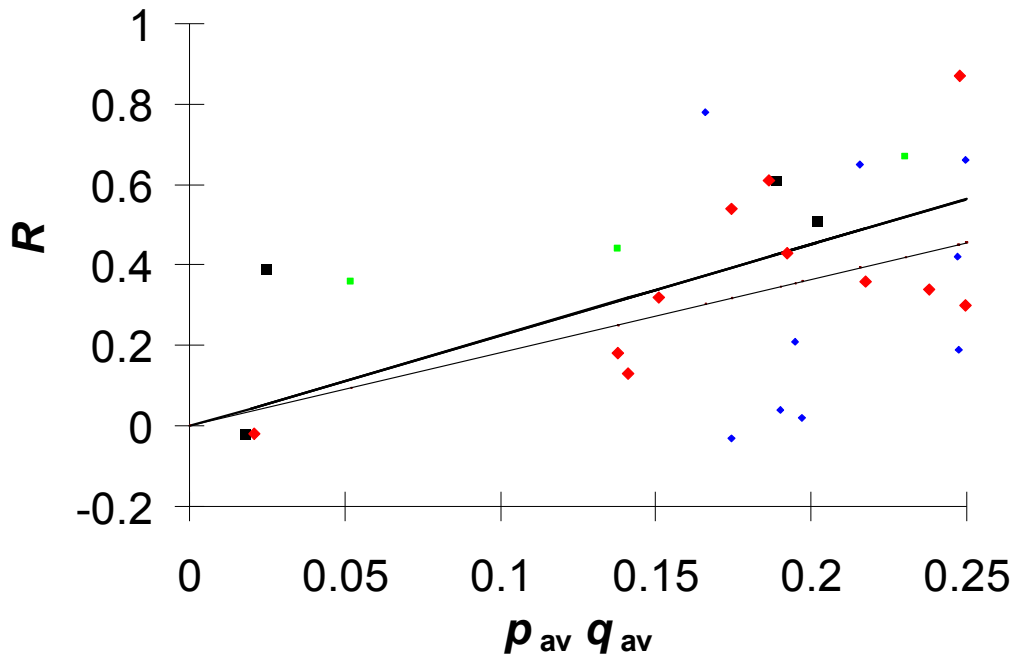


Figure S2. Least squares regression of linkage disequilibrium (R) for *H. melpomene* unlinked loci using sample sizes greater than 17. The estimates shown are for unlinked loci D and Yb , with 1986 results shown in red, 2011 results shown in black. The fitted regression (heavy line) gives $R_{max} = 0.56$ (0.42-0.71) when $p_{av}q_{av} = 0.25$. Also shown are disequilibria between unlinked loci b and Yb , with 1986 results in blue, 2011 results in green. Here regression (faint line) gives $R_{max} = 0.45$ (0.22-0.69) when $p_{av}q_{av} = 0.25$ (the middle of the hybrid zone).

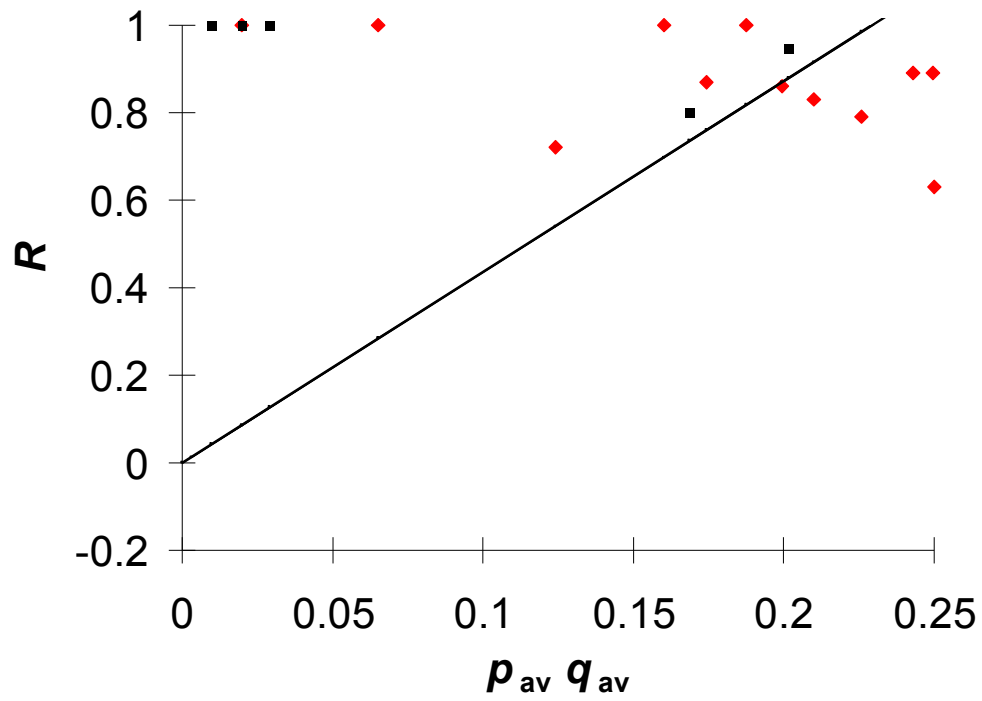


Figure S3. Linkage disequilibrium (R) for *H. melpomene* linked loci *N* and *Yb* plotted against the product of the average allele frequencies. 1986 values shown in red, 2011 values shown in black.

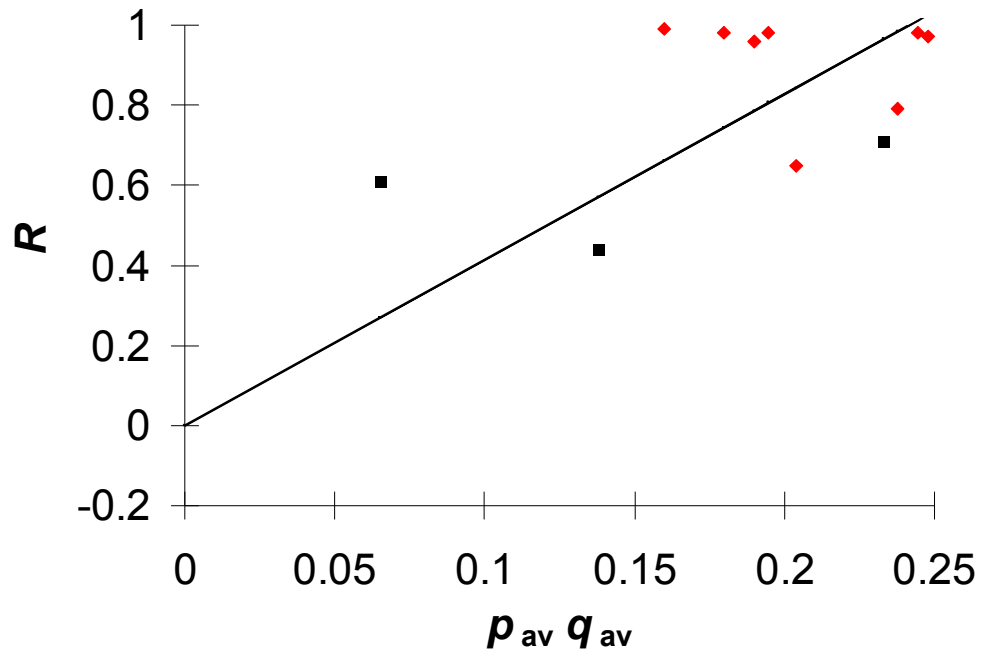


Figure S4. Linkage disequilibrium (R) for *H. melpomene* linked loci *b* and *D* plotted against the product of the average allele frequencies. 1986 values shown in red, 2011 values shown in black.

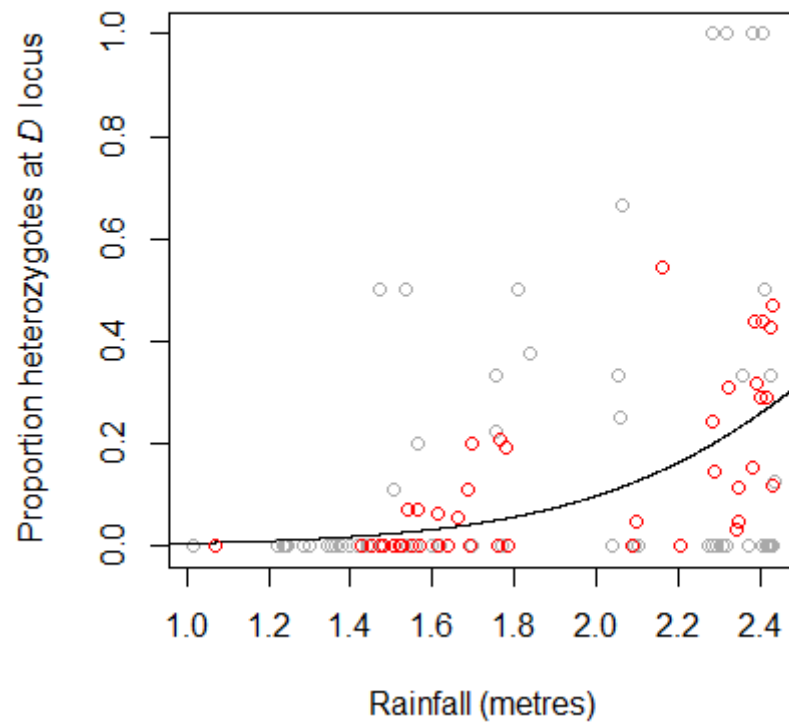


Figure S5. Rainfall plotted against proportions of heterozygotes at the *H. erato* *D* locus, with model predictions. Populations with sample sizes greater than 10 are shown in red.

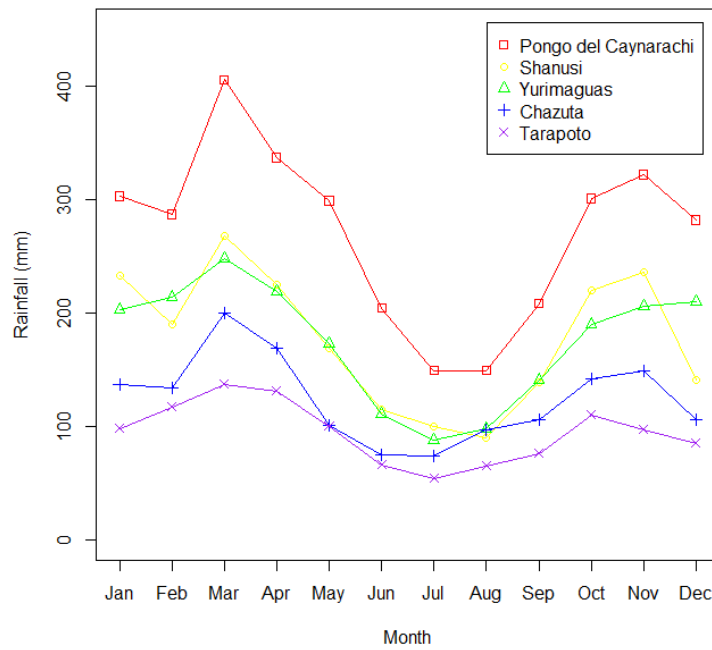


Figure S6. Monthly rainfall data at sites across the hybrid zone. Tarapoto and Chazuta fall on the Andean side of the hybrid zone, Shanusi and Yurimaguas on the Amazonian side and Pongo del Caynarachi falls in the centre of the hybrid zone.

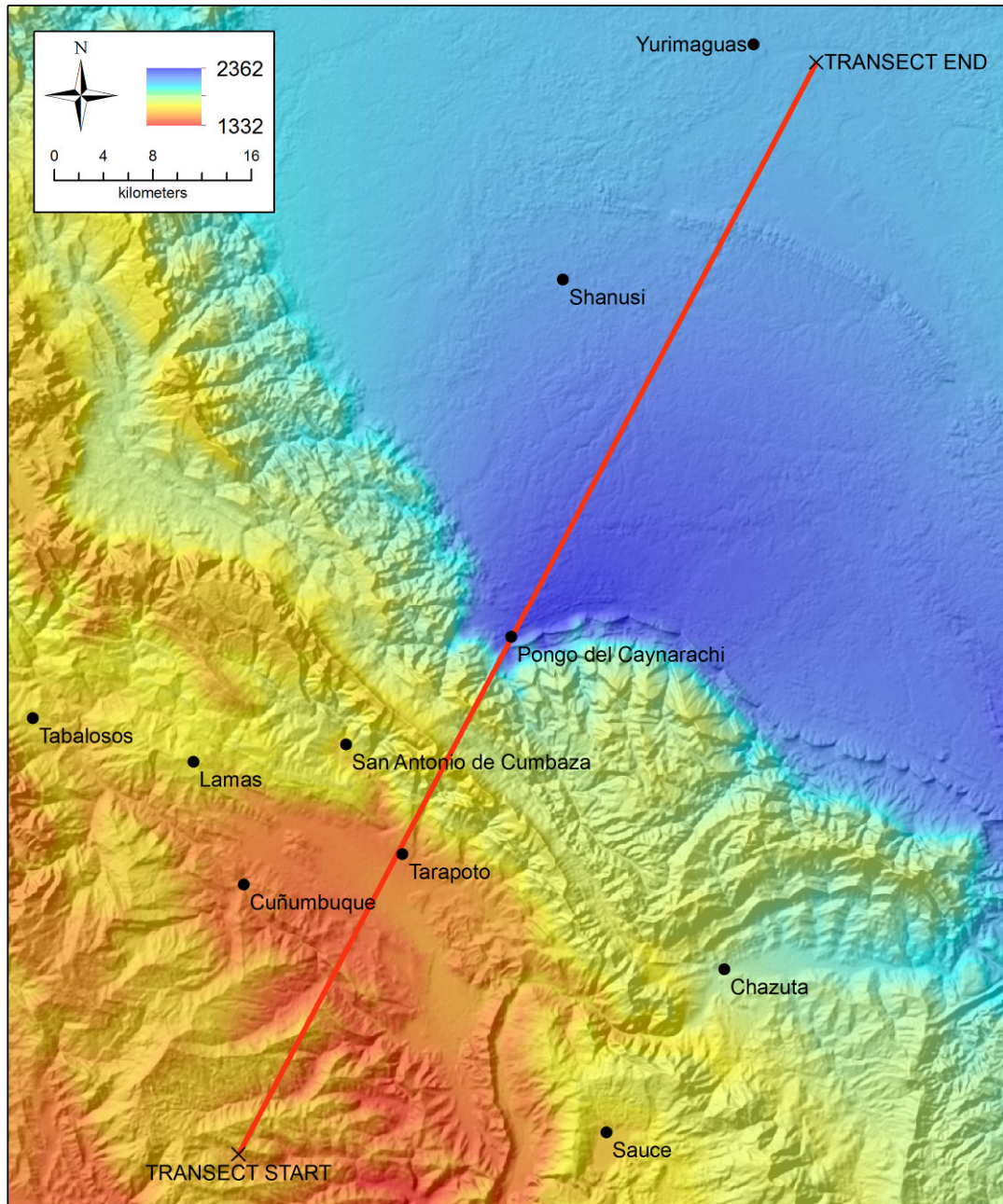


Figure S7. Gridded annual rainfall data overlaid onto a SRTM 90 metre resolution digital elevation model. The transect used in marked in red.

Table S1. Genotypic data for *Heliconius erato*. Exact coordinates: an individual GPS reading was taking for the specimen; approximate coordinate: the coordinates for the specimen were estimated, along with a measure of accuracy.

<u>General locality</u>	<u>Exact coordinates</u>	<u>Approximate coordinates</u>	<u>Accuracy</u>	<u>D</u>	<u>Cr</u>	<u>Sd</u>
Alianza		S6.126642 W76.267197	±50m	DD	C-	S-
Alto Shilcayo		S6.448943 W76.340899	±920m	dd	cc	ss
Alto Shilcayo		S6.448943 W76.340899	±920m	dd	cc	ss
Alto Shilcayo		S6.448943 W76.340899	±920m	dd	cc	ss
Bello Horizonte a Santa Elena	S6.52232 W76.28278			dd	cc	ss
Bello Horizonte a Santa Elena	S6.52232 W76.28278			dd	cc	ss
Bello Horizonte a Santa Elena	S6.52071 W76.28428			dd	cc	ss
Bello Horizonte a Santa Elena	S6.52071 W76.28428			dd	cc	ss
Bello Horizonte a Santa Elena	S6.52232 W76.28278			dd	cc	ss
Bello Horizonte a Santa Elena	S6.52232 W76.28278			dd	cc	ss
Bello Horizonte a Santa Elena	S6.52232 W76.28278			dd	cc	ss
Bello Horizonte a Santa Elena	S6.52232 W76.28278			dd	cc	ss
Bello Horizonte a Santa Elena	S6.52232 W76.28278			dd	cc	ss
Bello Horizonte a Santa Elena	S6.52232 W76.28278			dd	cc	ss
Bello Horizonte a Santa Elena	S6.52232 W76.28278			dd	cc	ss
Bello Horizonte a Santa Elena	S6.52232 W76.28278			dd	cc	ss
Bello Horizonte a Santa Elena	S6.52062 W76.28385			dd	cc	ss
Biodiversidad		S6.46041 W76.29039	±100m	dd	cc	Ss
Bonilla	S6.22317 W76.28631			Dd	C-	S-
Brisas del Paredon	S6.43048 W76.26252			Dd	cc	ss
Camino a Isla Bonita	S6.45929 W76.37193			dd	cc	ss
Camino a Isla Bonita	S6.45929 W76.37193			dd	cc	ss
Camino a Isla Bonita	S6.45929 W76.37193			dd	cc	ss
Casa de Orlando	S6.442639 W76.335470			dd	cc	ss
Casa de Orlando	S6.442639 W76.335470			dd	cc	ss
Casa de Orlando	S6.442639 W76.335470			dd	cc	ss
Casa de Orlando	S6.442639 W76.335470			dd	cc	ss
Casa de Orlando	S6.442639 W76.335470			dd	cc	ss
Casa de Orlando	S6.442639 W76.335470			dd	cc	ss
Casa de Orlando	S6.442639 W76.335470			dd	cc	ss
Casa de Orlando	S6.442639 W76.335470			dd	cc	ss
Casa de Orlando	S6.442639 W76.335470			dd	cc	ss
Cataratas de Ahuashiyacu	S6.45402 W76.31108			dd	cc	ss
Cataratas de Ahuashiyacu	S6.45672 W76.30921			dd	cc	ss
Cataratas de Ahuashiyacu	S6.45353 W76.31189			dd	cc	ss
Cataratas de Ahuashiyacu	S6.45353 W76.31189			dd	cc	ss
Cataratas de Ahuashiyacu	S6.45353 W76.31189			dd	cc	ss
Cataratas de Ahuashiyacu	S6.45353 W76.31189			dd	cc	ss
Cataratas de Ahuashiyacu	S6.45353 W76.31189			dd	cc	ss
Chazuta	S6.56654 W76.12738			dd	cc	ss
Chazuta	S6.56654 W76.12738			dd	cc	ss

BELOW IS R CODE FOR GENERATING THEORETICAL CLINES, FITTING THOSE CLINES TO OBSERVED DATA
USING LIKELIHOOD, AND FINDING SUPPORT LIMITS FOR DIFFERENCES IN WIDTH RATIOS BETWEEN
CLINES. THE CODE WAS NOT WRITTEN WITH OTHER USERS IN MIND, AND SO IS LIKELY TO BE HARD
TO DECIPHER. ANY QUERIES SHOULD BE MADE TO NEIL ROSSER.

```
#####  
####SIMULATE CLINES FOR HELICONIUS ERATO (3 LOCI)####  
#####
```

```
#!/usr/bin/env Rscript
```

```
library(data.table)
```

```
#STARTING CONDITIONS
```

```
#Vector of demes (should be an even number)
```

```
demes<-c(1:100)
```

```
#time steps
```

```
time_step<-c(1:100)
```

```
results<-matrix(ncol=6+length(demes))
```

```
colnames(results) <- c("locus","selectionD","selectionC","selectionS","migration","generation",demes)
```

```
results<-results[-1,]
```

```
#set up the loops for the varying levels of selection /migration you want to test
```

```
for (selectionD in c(0.1)){
```

```
for (selectionC in seq(0.1,0.8,0.1)){
```

```
for (selectionS in seq(0.1,0.8,0.1)){
```

```
for (n in c(10,20,30,40)){
```

```
print(paste(c("selection =",selectionD,"migration =",n)))
```

```
#Specify the loci, gametes and possible genotypes
```

```
locus1<-c("D","d")
```

```
locus2<-c("C","c")
```

```
locus3<-c("S","s")
```

```
gametes<-c()
```

```
for(i in locus1){
```

```
for (j in locus2){
```

```
for (k in locus3){
```

```
gametes<-append(gametes,paste(i,j,k,sep=""))
```

```
}}}
```

```
genotypes<-c()
```

```
for(i in gametes){
```

```
for (j in gametes){
```

```
genotypes<-append(genotypes,paste(i,j,sep=""))}}
```

```
#starting frequency of alleles
```

```
D<-c(rep(0,length(demes)/2),rep(1,length(demes)/2))
```

```
d<-1-D
```

```

C<-c(rep(0,length(demes)/2),rep(1,length(demes)/2))
c<-1-C
S<-c(rep(0,length(demes)/2),rep(1,length(demes)/2))
s<-1-S
#selection - see for loop below
#Migration function, for n see for loop below (n is the maximum distance (= deme) that genotypes will
migrate to. k (the number of trials), is 2*n)
a<-0.5
mig<-function(i)
{((factorial(2*n))/((factorial(n-i))*(factorial(n+i))))*(a^(n-i))*((1-a)^(n+i))}

demic_freqsD<-D
demic_freqsC<-C
demic_freqsS<-S

#FOR EACH DEME IN THE VECTOR CALCULATE GENOTYPE FREQUENCIES BASED ON THE ALLELES...
freq_genotypes_demes<-matrix(nrow=length(demes), ncol=length(genotypes))
for (j in demes){
#CALCULATE GENOTYPE FREQUENCIES
genotype_freqs<-c()
for (k in genotypes){
genotype<-unlist(strsplit(k, split=""))
prop_expression<-c()
for (l in 1:length(genotype)){
if (l<length(genotype)){
prop_expression<-paste(prop_expression,genotype[l],"[",j,"]", "*")
}else{
prop_expression<-paste(prop_expression,genotype[l],"[",j,"]")}}
genotype_freqs<-append(genotype_freqs,eval(parse(text=prop_expression)))}
freq_genotypes_demes[j,]<-(genotype_freqs)
}

#FOR EACH TIME STEP
for (time in time_step){
print(c("time step =",time))
#MIGRATION...
new_freq_genotypes_demes<-matrix(0,nrow=length(demes), ncol=length(genotypes))

for (i in 1:length(demes)){

#If the first deme in the array
if(i==1){
#Do the prop staying in the deme
for (j in 1:ncol(freq_genotypes_demes)){
new_freq_genotypes_demes[i,j]<-new_freq_genotypes_demes[i,j]+mig(0)*freq_genotypes_demes[i,j]}
#now do the prop migrating to the jth deme away
for (j in 1:ncol(freq_genotypes_demes)){
for (k in 1:n){

```

```

new_freq_genotypes_demes[i+k,j]<-
new_freq_genotypes_demes[i+k,j]+(mig(k)*freq_genotypes_demes[i,j])}

#Now do the props migrating in from the imaginary demes
for (j in 1:ncol(freq_genotypes_demes)){
  for (k in 1:n){
    for (l in k:n){
      new_freq_genotypes_demes[i+k-1,j]<-new_freq_genotypes_demes[i+k-
1,j]+(mig(l)*freq_genotypes_demes[i,j])
    }}}

#If the last deme in the array
else if(i==length(demes)){
  #Do the prop staying in the deme
  for (j in 1:ncol(freq_genotypes_demes)){
    new_freq_genotypes_demes[i,j]<-new_freq_genotypes_demes[i,j]+mig(0)*freq_genotypes_demes[i,j]
  }
  #now do the prop migrating to the jth deme away
  for (j in 1:ncol(freq_genotypes_demes)){
    for (k in 1:n){
      new_freq_genotypes_demes[i-k,j]<-new_freq_genotypes_demes[i-
k,j]+(mig(k)*freq_genotypes_demes[i,j])}
    }
  }
  #Now do the props migrating in from the imaginary demes
  for (j in 1:ncol(freq_genotypes_demes)){
    for (k in 1:n){
      for (l in k:n){
        new_freq_genotypes_demes[(i+1)-k,j]<-new_freq_genotypes_demes[(i+1)-
k,j]+(mig(l)*freq_genotypes_demes[i,j])
      }}}
  }

#Else all other demes...
}else{
  #prop staying in deme
  for (j in 1:ncol(freq_genotypes_demes)){
    new_freq_genotypes_demes[i,j]<-new_freq_genotypes_demes[i,j]+mig(0)*freq_genotypes_demes[i,j]
  }
  #now do the prop migrating to the jth deme away
  for (j in 1:ncol(freq_genotypes_demes)){
    for (k in 1:n){
      if(i+k<=length(demes)){
        new_freq_genotypes_demes[i+k,j]<-
        new_freq_genotypes_demes[i+k,j]+(mig(k)*freq_genotypes_demes[i,j])
      }
      if(i-k>=1){
        new_freq_genotypes_demes[i-k,j]<-new_freq_genotypes_demes[i-
k,j]+(mig(k)*freq_genotypes_demes[i,j])}
    }}}
  }

freq_genotypes_demes<-new_freq_genotypes_demes

#SELECTION

```

```
new_freq_genotypes_demes<-matrix(0,nrow=length(demes), ncol=length(genotypes))
```

```
#first calculate the fraction of each genotype that survive
```

```
for (j in 1:length(demes)){
```

```
  for (k in 1:length(genotypes)){
```

```
    #selection on the D locus
```

```
    genotype_D<-paste(substr(genotypes[k],1,1),substr(genotypes[k],4,4),sep="")
```

```
    D_DH_freqs<-c()
```

```
    D_HET_freqs<-c()
```

```
    D_RH_freqs<-c()
```

```
    for (l in 1:length(genotypes)){
```

```
      if (k != l){
```

```
        genotypes_D<-c(paste((substr(genotypes[l],1,1)),substr(genotypes[l],4,4),sep=""))
```

```
        if (genotypes_D=="DD"){
```

```
          D_DH_freqs<-append(D_DH_freqs,freq_genotypes_demes[j,l])
```

```
        }else if (genotypes_D=="Dd" | genotype_D=="dD"){
```

```
          D_HET_freqs<-append(D_HET_freqs,freq_genotypes_demes[j,l])
```

```
        }else if (genotypes_D=="dd"){
```

```
          D_RH_freqs<-append(D_RH_freqs,freq_genotypes_demes[j,l])
```

```
    D_DH_freqs<-sum(D_DH_freqs)
```

```
    D_HET_freqs<-sum(D_HET_freqs)
```

```
    D_RH_freqs<-sum(D_RH_freqs)
```

```
    if (genotype_D=="DD"){
```

```
      W_genD<-1-(selectionD*(D_RH_freqs+0.5*D_HET_freqs))
```

```
    }else if (genotype_D=="Dd" | genotype_D=="dD"){
```

```
      W_genD<-1-(selectionD*(0.5*D_DH_freqs+0.5*D_RH_freqs))
```

```
    }else if (genotype_D=="dd"){
```

```
      W_genD<-1-(selectionD*(D_DH_freqs+0.5*D_HET_freqs))
```

```
    }}
```

```
#selection on the C locus
```

```
genotype_C<-paste(substr(genotypes[k],2,2),substr(genotypes[k],5,5),sep="")
```

```
C_DH_freqs<-c()
```

```
C_HET_freqs<-c()
```

```
C_RH_freqs<-c()
```

```
for (l in 1:length(genotypes)){
```

```
  if (k != l){
```

```
    genotypes_C<-c(paste((substr(genotypes[l],2,2)),substr(genotypes[l],5,5),sep=""))
```

```
    if (genotypes_C=="CC"){
```

```
      C_DH_freqs<-append(C_DH_freqs,freq_genotypes_demes[j,l])
```

```
    }else if (genotypes_C=="Cc" | genotype_C=="cC"){
```

```
      C_HET_freqs<-append(C_HET_freqs,freq_genotypes_demes[j,l])
```

```
    }else if (genotypes_C=="cc"){
```

```
      C_RH_freqs<-append(C_RH_freqs,freq_genotypes_demes[j,l])
```

```

C_DH_freqs<-sum(C_DH_freqs)
C_HET_freqs<-sum(C_HET_freqs)
C_RH_freqs<-sum(C_RH_freqs)

if (genotype_C=="CC"){
W_genC<-1-(selectionC*(C_RH_freqs))
}else if (genotype_C=="Cc"|genotype_C=="cC"){
W_genC<-1-(selectionC*(C_RH_freqs))
}else if (genotype_C=="cc"){
W_genC<-1-(selectionC*(C_DH_freqs+C_HET_freqs))}
}}

#selection on the S locus
genotype_S<-paste(substr(genotypes[k],3,3),substr(genotypes[k],6,6),sep="")
S_DH_freqs<-c()
S_HET_freqs<-c()
S_RH_freqs<-c()
for (l in 1:length(genotypes)){
if (k != l){
genotypes_S<-c(paste((substr(genotypes[l],3,3)),substr(genotypes[l],6,6),sep=""))
if (genotypes_S=="SS"){
S_DH_freqs<-append(S_DH_freqs,freq_genotypes_demes[j,l])
}else if (genotypes_S=="Ss"|genotypes_S=="sS"){
S_HET_freqs<-append(S_HET_freqs,freq_genotypes_demes[j,l])
}else if (genotypes_S=="ss"){
S_RH_freqs<-append(S_RH_freqs,freq_genotypes_demes[j,l])}

S_DH_freqs<-sum(S_DH_freqs)
S_HET_freqs<-sum(S_HET_freqs)
S_RH_freqs<-sum(S_RH_freqs)

if (genotype_S=="SS"){
W_genS<-1-(selectionS*(S_RH_freqs))
}else if (genotype_S=="Ss"|genotype_S=="sS"){
W_genS<-1-(selectionS*(S_RH_freqs))
}else if (genotype_S=="ss"){
W_genS<-1-(selectionS*(S_DH_freqs+S_HET_freqs))}
}}

#now calculate the proportion surviving given the proportion surviving at each locus
new_freq_genotypes_demes[j,k]<-(W_genD*W_genC*W_genS)*freq_genotypes_demes[j,k]

}}

#Now calculate the new genotype frequencies for each deme
for (j in 1:length(demes)){
total<-sum(new_freq_genotypes_demes[j,])
for (k in 1:length(genotypes)){

```

```
new_freq_genotypes_demes[j,k]<-new_freq_genotypes_demes[j,k]/total
}}
```

```
freq_genotypes_demes<-new_freq_genotypes_demes
```

```
#calculate gametes frequencies for each deme:
new_freq_genotypes_demes<-matrix(0,nrow=length(demes), ncol=length(genotypes))
for (j in 1:length(demes)){
list_of_gametes<-c()
gamete_frequencies<-c()
for (k in 1:length(genotypes)){
genotype<-unlist(strsplit(genotypes[k],split=""))
#make a vector with all the possible gametes for that genotype..
gametes<-c()
gametes<-append(gametes,paste(genotype[1],genotype[2],genotype[3],sep=""))
gametes<-append(gametes,paste(genotype[1],genotype[2],genotype[6],sep=""))
gametes<-append(gametes,paste(genotype[1],genotype[5],genotype[3],sep=""))
gametes<-append(gametes,paste(genotype[1],genotype[5],genotype[6],sep=""))
gametes<-append(gametes,paste(genotype[4],genotype[2],genotype[3],sep=""))
gametes<-append(gametes,paste(genotype[4],genotype[2],genotype[6],sep=""))
gametes<-append(gametes,paste(genotype[4],genotype[5],genotype[3],sep=""))
gametes<-append(gametes,paste(genotype[4],genotype[5],genotype[6],sep=""))
#now make a list of the probabilities of each gametic type. These probabilities are equal with no linkage.
gametic_probs<-rep(0.125,8)
gametic_freqs<-freq_genotypes_demes[j,k]*gametic_probs
list_of_gametes<-append(list_of_gametes,gametes)
gamete_frequencies<-append(gamete_frequencies,gametic_freqs)}
#Use the data.table package to get the list of unique gametes and sum their frequencies...
gamete_table<-data.frame(list_of_gametes,gamete_frequencies)
gamete_table<-data.table(gamete_table)
gamete_table<-
gamete_table[,list(gamete_frequencies=sum(gamete_frequencies)),by='list_of_gametes']

#Now calculate the genotype frequencies generated by these gametes
punnet_square<-matrix(nrow=length(gamete_table$list_of_gametes),
ncol=length(gamete_table$list_of_gametes))
rownames(punnet_square) <- gamete_table$list_of_gametes
colnames(punnet_square) <- gamete_table$list_of_gametes

for (l in 1:length(gamete_table$list_of_gametes)){
for (m in 1:length(gamete_table$list_of_gametes)){
punnet_square[l,m]<-gamete_table$gamete_frequencies[l]*gamete_table$gamete_frequencies[m]}}

colnames(new_freq_genotypes_demes) <- genotypes

for (z in 1:length(genotypes)){
```

```

genotype_freq_from_Punnet<-
punnet_square[substr(genotypes[z],1,nchar(genotypes[z])/2),substr(genotypes[z],nchar(genotypes[z])/2
+1,nchar(genotypes[z]))]
new_freq_genotypes_demes[j,z]<-genotype_freq_from_Punnet
}}
freq_genotypes_demes<-new_freq_genotypes_demes

#Now save the allele frequencies
newD<-c()
newC<-c()
newS<-c()

for (j in 1:length(demes)){
demicD<-c()
demicC<-c()
demicS<-c()
for (k in 1:length(genotypes)){
genotypeD<-c(paste((substr(genotypes[k],1,1)),substr(genotypes[k],4,4),sep=""))
genotypeC<-c(paste((substr(genotypes[k],2,2)),substr(genotypes[k],5,5),sep=""))
genotypeS<-c(paste((substr(genotypes[k],3,3)),substr(genotypes[k],6,6),sep=""))

if(genotypeD=="DD"){
demicD<-append(demicD,new_freq_genotypes_demes[j,k])}
if(genotypeD=="Dd" | genotypeD=="dD"){
demicD<-append(demicD,new_freq_genotypes_demes[j,k]/2)}

if(genotypeS=="SS"){
demicS<-append(demicS,new_freq_genotypes_demes[j,k])}
if(genotypeS=="Ss" | genotypeS=="sS"){
demicS<-append(demicS,new_freq_genotypes_demes[j,k]/2)}

if(genotypeC=="CC"){
demicC<-append(demicC,new_freq_genotypes_demes[j,k])}
if(genotypeC=="Cc" | genotypeC=="cC"){
demicC<-append(demicC,new_freq_genotypes_demes[j,k]/2)}

}
newD<-append(newD,sum(demicD))
newC<-append(newC,sum(demicC))
newS<-append(newS,sum(demicS))

D<-newD
d<-1-D
C<-newC
c<-1-C
S<-newS
s<-1-S

```

```

demic_freqsD<-rbind(demic_freqsD,D)
demic_freqsC<-rbind(demic_freqsC,C)
demic_freqsS<-rbind(demic_freqsS,S)

}

results<-rbind(results,c("D",selectionD,selectionC,selectionS,n,max(time),demic_freqsD[max(time)+1,]))
results<-rbind(results,c("C",selectionD,selectionC,selectionS,n,max(time),demic_freqsC[max(time)+1,]))
results<-rbind(results,c("S",selectionD,selectionC,selectionS,n,max(time),demic_freqsS[max(time)+1,]))

}}}}

write.csv(results,paste("results_selectionD_",selectionD,".csv",sep=""))

#####
####SIMULATED CLINES FOR HELICONIUS MELPOMENE (4 LOCI)###
#####

#!/usr/bin/env Rscript

library(data.table)

#STARTING CONDITIONS
#Vector of demes (should be an even number)
demes<-c(1:10)
#time steps
time_step<-c(1:2)

results<-matrix(ncol=9+length(demes))
colnames(results) <-
c("locus","selectionB","selectionD","selectionN","selectionY","migration","recomb_r_BD","recomb_r_NY","n_generations",demes)
results<-results[-1,]

#set up the loops for the varying levels of selection /migration you want to test
for (selectionB in seq(0.6,0.7,0.1)){
  for (selectionD in seq(0.6,0.7,0.1)){
    for (selectionN in seq(0.6,0.7,0.1)){
      for (selectionY in seq(0.6,0.7,0.1)){
        for (n in c(1,2)){
          for (recomb_rate_BD in c(0.03,0.135)){
            for (recomb_rate_NY in c(0.03,0.135)){

              print(paste(c("selection =",selectionB)))

              #calculate the gamete weights for the specified recombination rates

```

```

recomb_at_niether<-((1-recomb_rate_BD)*(1-recomb_rate_NY))/4
recomb_at_BD<-(recomb_rate_BD*(1-recomb_rate_NY))/4
recomb_at_NY<-((1-recomb_rate_BD)*recomb_rate_NY)/4
recomb_at_both<-(recomb_rate_BD*recomb_rate_NY)/4

```

#Specify the loci, gametes and possible genotypes

```

locus1<-c("B","b")
locus2<-c("D","d")
locus3<-c("N","n")
locus4<-c("Y","y")
gametes<-c()
for(i in locus1){
  for(j in locus2){
    for(k in locus3){
      for(l in locus4){
        gametes<-append(gametes,paste(i,j,k,l,sep=""))
      }}}
genotypes<-c()
for(i in gametes){
  for(j in gametes){
    genotypes<-append(genotypes,paste(i,j,sep=""))}}
#starting frequency of alleles

```

```

B_all<-c(rep(1,length(demes)/2),rep(0,length(demes)/2))
b_all<-1-B_all
D_all<-c(rep(0,length(demes)/2),rep(1,length(demes)/2))
d_all<-1-D_all
N_all<-c(rep(0,length(demes)/2),rep(1,length(demes)/2))
n_all<-1-N_all
Y_all<-c(rep(0,length(demes)/2),rep(1,length(demes)/2))
y_all<-1-Y_all

```

#Migration function. n is the maximum distance (= deme) that genotypes will migrate to. k (the number of trials), is 2*n

```

a<-0.5
mig<-function(i)
{((factorial(2*n))/((factorial(n-i))*(factorial(n+i))))*(a^(n-i))*((1-a)^(n+i))}
#Find max slope of simulated cline function
max_slope <- function(x) {
  slopes<-c()
  for(i in 1:(length(x)-1)){
    slopes<-append(slopes,x[i+1]-x[i])}
  return(max(slopes))
}

```

```

demic_freqsB<-B_all
demic_freqsD<-D_all
demic_freqsN<-N_all

```

```
demic_freqsY<-Y_all
```

```
#FOR EACH DEME IN THE VECTOR CALCULATE GENOTYPE FREQUENCIES BASED ON THE ALLELES...
```

```
freq_genotypes_demes<-matrix(nrow=length(demes), ncol=length(genotypes))
for (j in demes){
#CALCULATE GENOTYPE FREQUENCIES
genotype_freqs<-c()
for (k in genotypes){
genotype<-unlist(strsplit(k, split=""))
prop_expression<-c()
for (l in 1:length(genotype)){
if (l<length(genotype)){
prop_expression<-paste(prop_expression,genotype[l], "_all", "[",j,"]", "*",sep="")
}else{
prop_expression<-paste(prop_expression,genotype[l], "_all", "[",j,"]",sep="")
}
genotype_freqs<-append(genotype_freqs,eval(parse(text=prop_expression)))
freq_genotypes_demes[j,]<-(genotype_freqs)
}
```

```
#FOR EACH TIME STEP
```

```
for (time in time_step){
print(c("time step =",time))
#MIGRATION...
new_freq_genotypes_demes<-matrix(0,nrow=length(demes), ncol=length(genotypes))
```

```
for (i in 1:length(demes)){
```

```
#If the first deme in the array
```

```
if(i==1){
#Do the prop staying in the deme
for (j in 1:ncol(freq_genotypes_demes)){
new_freq_genotypes_demes[i,j]<-new_freq_genotypes_demes[i,j]+mig(0)*freq_genotypes_demes[i,j]}
#now do the prop migrating to the jth deme away
for (j in 1:ncol(freq_genotypes_demes)){
for (k in 1:n){
new_freq_genotypes_demes[i+k,j]<-
new_freq_genotypes_demes[i+k,j]+(mig(k)*freq_genotypes_demes[i,j])}}
```

```
#Now do the props migrating in from the imaginary demes
```

```
for (j in 1:ncol(freq_genotypes_demes)){
for (k in 1:n){
for (l in k:n){
new_freq_genotypes_demes[i+k-1,j]<-new_freq_genotypes_demes[i+k-1,j]+(mig(l)*freq_genotypes_demes[i,j])
}}}}
```

```
#If the last deme in the array
```

```
else if(i==length(demes)){
```

```

#Do the prop staying in the deme
for (j in 1:ncol(freq_genotypes_demes)){
  new_freq_genotypes_demes[i,j]<-new_freq_genotypes_demes[i,j]+mig(0)*freq_genotypes_demes[i,j]}
#now do the prop migrating to the jth deme away
for (j in 1:ncol(freq_genotypes_demes)){
  for (k in 1:n){
    new_freq_genotypes_demes[i-k,j]<-new_freq_genotypes_demes[i-
    k,j]+(mig(k)*freq_genotypes_demes[i,j])}}
#Now do the props migrating in from the imaginary demes
for (j in 1:ncol(freq_genotypes_demes)){
  for (k in 1:n){
    for (l in k:n){
      new_freq_genotypes_demes[(i+1)-k,j]<-new_freq_genotypes_demes[(i+1)-
      k,j]+(mig(l)*freq_genotypes_demes[i,j])
    }}}

#Else all other demes...
}else{
  #prop staying in deme
  for (j in 1:ncol(freq_genotypes_demes)){
    new_freq_genotypes_demes[i,j]<-new_freq_genotypes_demes[i,j]+mig(0)*freq_genotypes_demes[i,j]}
  #now do the prop migrating to the jth deme away
  for (j in 1:ncol(freq_genotypes_demes)){
    for (k in 1:n){
      if(i+k<=length(demes)){
        new_freq_genotypes_demes[i+k,j]<-
        new_freq_genotypes_demes[i+k,j]+(mig(k)*freq_genotypes_demes[i,j])}
      if(i-k>=1){
        new_freq_genotypes_demes[i-k,j]<-new_freq_genotypes_demes[i-
        k,j]+(mig(k)*freq_genotypes_demes[i,j])}}}}

freq_genotypes_demes<-new_freq_genotypes_demes

#SELECTION

new_freq_genotypes_demes<-matrix(0,nrow=length(demes), ncol=length(genotypes))

#first calculate the fraction of each genotype that survive
for (j in 1:length(demes)){
  for (k in 1:length(genotypes)){

#selection on the B locus
genotype_B<-paste(substr(genotypes[k],1,1),substr(genotypes[k],5,5),sep="")
B_DH_freqs<-c()
B_HET_freqs<-c()
B_RH_freqs<-c()
for (l in 1:length(genotypes)){
  if (k != l){

```

```

genotypes_B<-c(paste((substr(genotypes[l],1,1)),substr(genotypes[l],5,5),sep=""))
if (genotypes_B=="BB"){
  B_DH_freqs<-append(B_DH_freqs,freq_genotypes_demes[j,l])
}else if (genotypes_B=="Bb"|genotypes_B=="bB"){
  B_HET_freqs<-append(B_HET_freqs,freq_genotypes_demes[j,l])
}else if (genotypes_B=="bb"){
  B_RH_freqs<-append(B_RH_freqs,freq_genotypes_demes[j,l])}

B_DH_freqs<-sum(B_DH_freqs)
B_HET_freqs<-sum(B_HET_freqs)
B_RH_freqs<-sum(B_RH_freqs)

if (genotype_B=="BB"){
  W_genB<-1-(selectionB*(B_RH_freqs))
}else if (genotype_B=="Bb"|genotype_B=="bB"){
  W_genB<-1-(selectionB*(B_RH_freqs))
}else if (genotype_B=="bb"){
  W_genB<-1-(selectionB*(B_DH_freqs+B_HET_freqs))}
}}

#selection on the D locus
genotype_D<-paste(substr(genotypes[k],2,2),substr(genotypes[k],6,6),sep="")
D_DH_freqs<-c()
D_HET_freqs<-c()
D_RH_freqs<-c()
for (l in 1:length(genotypes)){
  if (k != l){
    genotypes_D<-c(paste((substr(genotypes[l],2,2)),substr(genotypes[l],6,6),sep=""))
    if (genotypes_D=="DD"){
      D_DH_freqs<-append(D_DH_freqs,freq_genotypes_demes[j,l])
    }else if (genotypes_D=="Dd"|genotypes_D=="dD"){
      D_HET_freqs<-append(D_HET_freqs,freq_genotypes_demes[j,l])
    }else if (genotypes_D=="dd"){
      D_RH_freqs<-append(D_RH_freqs,freq_genotypes_demes[j,l])}

D_DH_freqs<-sum(D_DH_freqs)
D_HET_freqs<-sum(D_HET_freqs)
D_RH_freqs<-sum(D_RH_freqs)

if (genotype_D=="DD"){
  W_genD<-1-(selectionD*(D_RH_freqs))
}else if (genotype_D=="Dd"|genotype_D=="dD"){
  W_genD<-1-(selectionD*(D_RH_freqs))
}else if (genotype_D=="dd"){
  W_genD<-1-(selectionD*(D_DH_freqs+D_HET_freqs))}
}}

#selection on the N locus

```

```

genotype_N<-paste(substr(genotypes[k],3,3),substr(genotypes[k],7,7),sep="")
N_DH_freqs<-c()
N_HET_freqs<-c()
N_RH_freqs<-c()
for (l in 1:length(genotypes)){
  if (k != l){
    genotypes_N<-c(paste((substr(genotypes[l],3,3)),substr(genotypes[l],7,7),sep=""))
    if (genotypes_N=="NN"){
      N_DH_freqs<-append(N_DH_freqs,freq_genotypes_demes[j,l])
    }else if (genotypes_N=="Nn" | genotypes_N=="nN"){
      N_HET_freqs<-append(N_HET_freqs,freq_genotypes_demes[j,l])
    }else if (genotypes_N=="nn"){
      N_RH_freqs<-append(N_RH_freqs,freq_genotypes_demes[j,l])
    }

    N_DH_freqs<-sum(N_DH_freqs)
    N_HET_freqs<-sum(N_HET_freqs)
    N_RH_freqs<-sum(N_RH_freqs)

    if (genotype_N=="NN"){
      W_genN<-1-(selectionN*(N_RH_freqs))
    }else if (genotype_N=="Nn" | genotype_N=="nN"){
      W_genN<-1-(selectionN*(N_RH_freqs))
    }else if (genotype_N=="nn"){
      W_genN<-1-(selectionN*(N_DH_freqs+N_HET_freqs))
    }
  }
}

#selection on the Y locus
genotype_Y<-paste(substr(genotypes[k],4,4),substr(genotypes[k],8,8),sep="")
Y_DH_freqs<-c()
Y_HET_freqs<-c()
Y_RH_freqs<-c()
for (l in 1:length(genotypes)){
  if (k != l){
    genotypes_Y<-c(paste((substr(genotypes[l],4,4)),substr(genotypes[l],8,8),sep=""))
    if (genotypes_Y=="YY"){
      Y_DH_freqs<-append(Y_DH_freqs,freq_genotypes_demes[j,l])
    }else if (genotypes_Y=="Yy" | genotypes_Y=="yY"){
      Y_HET_freqs<-append(Y_HET_freqs,freq_genotypes_demes[j,l])
    }else if (genotypes_Y=="yy"){
      Y_RH_freqs<-append(Y_RH_freqs,freq_genotypes_demes[j,l])
    }

    Y_DH_freqs<-sum(Y_DH_freqs)
    Y_HET_freqs<-sum(Y_HET_freqs)
    Y_RH_freqs<-sum(Y_RH_freqs)

    if (genotype_Y=="YY"){
      W_genY<-1-(selectionY*(Y_RH_freqs))
    }else if (genotype_Y=="Yy" | genotype_Y=="yY"){

```

```

W_genY<-1-(selectionY*(Y_RH_freqs))
}else if (genotype_Y=="yy"){
W_genY<-1-(selectionY*(Y_DH_freqs+Y_HET_freqs))
}}

#now calculate the proportion surviving given the proportion surviving at each locus
new_freq_genotypes_demes[j,k]<-(W_genB*W_genD*W_genN*W_genY)*freq_genotypes_demes[j,k]

}}

#Now calculate the new genotype frequencies for each deme
for (j in 1:length(demes)){
total<-sum(new_freq_genotypes_demes[j,])
for (k in 1:length(genotypes)){
new_freq_genotypes_demes[j,k]<-new_freq_genotypes_demes[j,k]/total
}}

freq_genotypes_demes<-new_freq_genotypes_demes

#calculate gametes frequencies for each deme:
new_freq_genotypes_demes<-matrix(0,nrow=length(demes), ncol=length(genotypes))
for (j in 1:length(demes)){
list_of_gametes<-c()
gamete_frequencies<-c()
for (k in 1:length(genotypes)){
genotype<-unlist(strsplit(genotypes[k],split=""))
#make a vector with all the possible gametes for that genotype..
gametes<-c()
gametes<-append(gametes,paste(genotype[1],genotype[2],genotype[3],genotype[4],sep=""))
gametes<-append(gametes,paste(genotype[1],genotype[2],genotype[3],genotype[8],sep=""))
gametes<-append(gametes,paste(genotype[1],genotype[2],genotype[7],genotype[4],sep=""))
gametes<-append(gametes,paste(genotype[1],genotype[2],genotype[7],genotype[8],sep=""))
gametes<-append(gametes,paste(genotype[1],genotype[6],genotype[3],genotype[4],sep=""))
gametes<-append(gametes,paste(genotype[1],genotype[6],genotype[3],genotype[8],sep=""))
gametes<-append(gametes,paste(genotype[1],genotype[6],genotype[7],genotype[4],sep=""))
gametes<-append(gametes,paste(genotype[1],genotype[6],genotype[7],genotype[8],sep=""))
gametes<-append(gametes,paste(genotype[5],genotype[2],genotype[3],genotype[4],sep=""))
gametes<-append(gametes,paste(genotype[5],genotype[2],genotype[3],genotype[8],sep=""))
gametes<-append(gametes,paste(genotype[5],genotype[2],genotype[7],genotype[4],sep=""))
gametes<-append(gametes,paste(genotype[5],genotype[2],genotype[7],genotype[8],sep=""))
gametes<-append(gametes,paste(genotype[5],genotype[6],genotype[3],genotype[4],sep=""))
gametes<-append(gametes,paste(genotype[5],genotype[6],genotype[3],genotype[8],sep=""))
gametes<-append(gametes,paste(genotype[5],genotype[6],genotype[7],genotype[4],sep=""))
gametes<-append(gametes,paste(genotype[5],genotype[6],genotype[7],genotype[8],sep=""))

#now make a list of the probabilities of each gametic type.
genotype_R<-
c(paste(genotype[1],1,sep=""),paste(genotype[2],2,sep=""),paste(genotype[3],3,sep=""),paste(genotype

```

```

[4],4,sep=""),paste(genotype[5],5,sep=""),paste(genotype[6],6,sep=""),paste(genotype[7],7,sep=""),paste(
e(genotype[8],8,sep=""))
#make a vector with all the possible gametes for that genotype..
gametes_R<-c()
gametes_R<-
append(gametes_R,paste(genotype_R[1],genotype_R[2],genotype_R[3],genotype_R[4],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[1],genotype_R[2],genotype_R[3],genotype_R[8],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[1],genotype_R[2],genotype_R[7],genotype_R[4],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[1],genotype_R[2],genotype_R[7],genotype_R[8],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[1],genotype_R[6],genotype_R[3],genotype_R[4],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[1],genotype_R[6],genotype_R[3],genotype_R[8],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[1],genotype_R[6],genotype_R[7],genotype_R[4],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[1],genotype_R[6],genotype_R[7],genotype_R[8],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[5],genotype_R[2],genotype_R[3],genotype_R[4],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[5],genotype_R[2],genotype_R[3],genotype_R[8],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[5],genotype_R[2],genotype_R[7],genotype_R[4],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[5],genotype_R[2],genotype_R[7],genotype_R[8],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[5],genotype_R[6],genotype_R[3],genotype_R[4],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[5],genotype_R[6],genotype_R[3],genotype_R[8],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[5],genotype_R[6],genotype_R[7],genotype_R[4],sep=""))
gametes_R<-
append(gametes_R,paste(genotype_R[5],genotype_R[6],genotype_R[7],genotype_R[8],sep=""))

gametic_probs<-c()

genotype_DB_pos1<-paste(genotype_R[1],genotype_R[2],sep="")
genotype_DB_pos2<-paste(genotype_R[5],genotype_R[6],sep="")
genotype_NY_pos1<-paste(genotype_R[3],genotype_R[4],sep="")
genotype_NY_pos2<-paste(genotype_R[7],genotype_R[8],sep="")

for (i in 1:length(gametes_R)){
gamete_split<-unlist(strsplit(gametes_R[i],split=""))
gamete_DB<-paste(gamete_split[1],gamete_split[2],gamete_split[3],gamete_split[4],sep="")
gamete_NY<-paste(gamete_split[5],gamete_split[6],gamete_split[7],gamete_split[8],sep="")

```

```

if ((gamete_DB==genotype_DB_pos1|gamete_DB==genotype_DB_pos2) &
(gamete_NY==genotype_NY_pos1|gamete_NY==genotype_NY_pos2)){
gametic_probs<-append(gametic_probs,recomb_at_niether)
}else if ((gamete_DB!=genotype_DB_pos1|gamete_DB!=genotype_DB_pos2) &
(gamete_NY==genotype_NY_pos1|gamete_NY==genotype_NY_pos2)){
gametic_probs<-append(gametic_probs,recomb_at_BD)
}else if ((gamete_DB==genotype_DB_pos1|gamete_DB==genotype_DB_pos2) &
(gamete_NY!=genotype_NY_pos1|gamete_NY!=genotype_NY_pos2)){
gametic_probs<-append(gametic_probs,recomb_at_NY)
}else if ((gamete_DB!=genotype_DB_pos1|gamete_DB!=genotype_DB_pos2) &
(gamete_NY!=genotype_NY_pos1|gamete_NY!=genotype_NY_pos2)){
gametic_probs<-append(gametic_probs,recomb_at_both)
}}

gametic_freqs<-freq_genotypes_demes[j,k]*gametic_probs
list_of_gametes<-append(list_of_gametes,gametes)
gamete_frequencies<-append(gamete_frequencies,gametic_freqs)}
#Use the data.table package to get the list of unique gametes and sum their frequencies...
gamete_table<-data.frame(list_of_gametes,gamete_frequencies)
gamete_table<-data.table(gamete_table)
gamete_table<-
gamete_table[,list(gamete_frequencies=sum(gamete_frequencies)),by='list_of_gametes']

#Now calculate the genotype frequencies generated by these gametes
punnet_square<-matrix(nrow=length(gamete_table$list_of_gametes),
ncol=length(gamete_table$list_of_gametes))
rownames(punnet_square) <- gamete_table$list_of_gametes
colnames(punnet_square) <- gamete_table$list_of_gametes

for (l in 1:length(gamete_table$list_of_gametes)){
for (m in 1:length(gamete_table$list_of_gametes)){
punnet_square[l,m]<-gamete_table$gamete_frequencies[l]*gamete_table$gamete_frequencies[m]}}

colnames(new_freq_genotypes_demes) <- genotypes

for (z in 1:length(genotypes)){

genotype_freq_from_Punnet<-
punnet_square[substr(genotypes[z],1,nchar(genotypes[z])/2),substr(genotypes[z],nchar(genotypes[z])/2
+1,nchar(genotypes[z]))]
new_freq_genotypes_demes[j,z]<-genotype_freq_from_Punnet
}}
freq_genotypes_demes<-new_freq_genotypes_demes

#Now save the allele frequencies
newB<-c()
newD<-c()

```

```

newN<-c()
newY<-c()

for (j in 1:length(demes)){
  demicB<-c()
  demicD<-c()
  demicN<-c()
  demicY<-c()
  for (k in 1:length(genotypes)){
    genotypeB<-c(paste((substr(genotypes[k],1,1)),substr(genotypes[k],5,5),sep=""))
    genotypeD<-c(paste((substr(genotypes[k],2,2)),substr(genotypes[k],6,6),sep=""))
    genotypeN<-c(paste((substr(genotypes[k],3,3)),substr(genotypes[k],7,7),sep=""))
    genotypeY<-c(paste((substr(genotypes[k],4,4)),substr(genotypes[k],8,8),sep=""))

    if(genotypeB=="BB"){
      demicB<-append(demicB,new_freq_genotypes_demes[j,k])
    }
    if(genotypeB=="Bb" | genotypeB=="bB"){
      demicB<-append(demicB,new_freq_genotypes_demes[j,k]/2)}

    if(genotypeD=="DD"){
      demicD<-append(demicD,new_freq_genotypes_demes[j,k])
    }
    if(genotypeD=="Dd" | genotypeD=="dD"){
      demicD<-append(demicD,new_freq_genotypes_demes[j,k]/2)}

    if(genotypeN=="NN"){
      demicN<-append(demicN,new_freq_genotypes_demes[j,k])
    }
    if(genotypeN=="Nn" | genotypeN=="nN"){
      demicN<-append(demicN,new_freq_genotypes_demes[j,k]/2)}

    if(genotypeY=="YY"){
      demicY<-append(demicY,new_freq_genotypes_demes[j,k])
    }
    if(genotypeY=="Yy" | genotypeY=="yY"){
      demicY<-append(demicY,new_freq_genotypes_demes[j,k]/2)}

  }
  newB<-append(newB,sum(demicB))
  newD<-append(newD,sum(demicD))
  newN<-append(newN,sum(demicN))
  newY<-append(newY,sum(demicY))

  B_all<-newB
  b_all<-1-B_all
  D_all<-newD
  d_all<-1-D_all
  N_all<-newN
  n_all<-1-N_all
  Y_all<-newY
  y_all<-1-Y_all

```

```

demic_freqsB<-rbind(demic_freqsB,B_all)
demic_freqsD<-rbind(demic_freqsD,D_all)
demic_freqsN<-rbind(demic_freqsN,N_all)
demic_freqsY<-rbind(demic_freqsY,Y_all)
}

results<-
rbind(results,c("B",selectionB,selectionD,selectionN,selectionY,n,recomb_rate_BD,recomb_rate_NY,max
(time),demic_freqsB[max(time)+1,]))
results<-
rbind(results,c("D",selectionB,selectionD,selectionN,selectionY,n,recomb_rate_BD,recomb_rate_NY,max
x(time),demic_freqsD[max(time)+1,]))
results<-
rbind(results,c("N",selectionB,selectionD,selectionN,selectionY,n,recomb_rate_BD,recomb_rate_NY,max
x(time),demic_freqsN[max(time)+1,]))
results<-
rbind(results,c("Y",selectionB,selectionD,selectionN,selectionY,n,recomb_rate_BD,recomb_rate_NY,max
(time),demic_freqsY[max(time)+1,]))

}}}}}}

write.csv(results,"results_melpomene.csv")

```

```

#####
###USE LIKELIHOOD TO FIT THE SIMULATED ERATO CLINES TO THE OBSERVED DATA###
#####

```

```
library(akima)
```

```
###likelihood functions to calculate the likelihoods for each cline separately;
```

```

func_likelihood_ERATO_LIKE_D <- function(x) {
  freqs_D <- aspline((x[2]*pred_D$km)+x[1], pred_D$freq, xout=obs$km, degree=10)
  p_D <- freqs_D$y
  p_D[p_D==0] <- 1e-5
  p_D[p_D==1] <- 1-1e-5
  likelihood<-sum((log(p_D^y_D))+ log((1-p_D)^(k_D-y_D)))
  return(likelihood)
}

```

```

func_likelihood_ERATO_LIKE_C <- function(x) {
  freqs_C <- aspline((x[2]*pred_C$km)+x[1], pred_C$freq, xout=obs$km, degree=10)
  p_C <- ((freqs_C$y*freqs_C$y)+(2*(freqs_C$y*(1-freqs_C$y))))
  p_C[p_C==0] <- 1e-55
  p_C[p_C==1] <- 1-1e-5
  likelihood<-sum((log(p_C^y_C))+ log((1-p_C)^(k_C-y_C)))
  return(likelihood)
}

```

```

    }

func_likelihood_ERATO_LIKE_S <- function(x) {
    freqs_S <- aspline((x[2]*pred_S$km)+x[1], pred_S$freq, xout=obs$km, degree=10)
    p_S <- ((freqs_S$y*freqs_S$y)+(2*(freqs_S$y*(1-freqs_S$y))))
    p_S[p_S==0] <- 1e-55
    p_S[p_S==1] <- 1-1e-5
    likelihood<-sum((log(p_S^y_S))+ log((1-p_S)^(k_S-y_S)))
    return(likelihood)
}

# function to calculate the likelihood of the three clines fitted simultaneously:

#THIS VERSION ALLOWS THE POSITIONS OF THE CLINES TO MOVE INDEPENDENTLY and the stretch
factor to change..

# requires vector x with SIX values(THREE constants that is added to the km pos of the predicted values
to make them match up with the observed data, AND THREE stretch factors),

# and then for each locus need to have the following data loaded up:
# pred_km (vector; km positions (demes) of simulated data), pred_freqs (vector; predicted frequency of
dominant PHENOTYPE for demes), obs_km (vector; observed km positions),
# k (vector; n trials in observed data), y (vector; n successes in observed data)

#do it for the D locus, ussing number of D alleles and expected proportion of D alleles

#Find max slope of simulated cline function
max_slope <- function(x) {
slopes<-c()
for (i in 1:(length(x)-1)){
slopes<-append(slopes,x[i+1]-x[i])}
return(max(slopes))
}

#observed data
obsJM <- read.csv("erato1990.csv", header=T, sep=",")
obsNR <- read.csv("erato2011.csv", header=T, sep=",")
combined <- read.csv("erato_combined.csv", header=T, sep=",")
predicted_all_data <- read.csv("erato_sim_results.csv", header=T, sep=",")
predicted_all_data <- subset(predicted_all_data, selectionD==0.6 & selectionC == 0.6 & selectionS ==
0.6)

#Approximate point of the middle of the hybrid zone on the transect (this is used as a starting point in
the optimization algorithm) (123.57 is Mallet 1990 estimated centre of D cline).
approx_mid_point<-123.57

#change this to the date set you want to analyse
obs<-combined

```

```

obs <- obs[order(obs$km),]

#Subset the D locus
predicted_D<-subset(predicted_all_data, predicted_all_data$locus == "D")
rownames(predicted_D) <- seq(length=nrow(predicted_D))
#this predicted for loops is just the allele frequencies without the row details, so you may have to
change the "5" depending on the input file
predicted_for_loops_D<-predicted_D[7:ncol(predicted_D)]
demes<-c(1:ncol(predicted_for_loops_D))

k_D <- obs$total_alleles ## total number of gene copies sampled
y_D <- obs$n_D_alleles ## number of successes

#Subset the C locus
predicted_C<-subset(predicted_all_data, predicted_all_data$locus == "C")
rownames(predicted_C) <- seq(length=nrow(predicted_C))

#this predicted for loops is just the allele frequencies without the row details, so you may have to
change the "5" depending on the input file
predicted_for_loops_C<-predicted_C[7:ncol(predicted_C)]
demes<-c(1:ncol(predicted_for_loops_C))

k_C <- obs$total_alleles/2 ## total number individuals
y_C <- k_C-(obs$n_cc_individuals) ## total number of C- individuals

#Subset the S locus
predicted_S<-subset(predicted_all_data, predicted_all_data$locus == "S")
rownames(predicted_S) <- seq(length=nrow(predicted_S))

#this predicted for loops is just the allele frequencies without the row details, so you may have to
change the "5" depending on the input file
predicted_for_loops_S<-predicted_S[7:ncol(predicted_S)]
demes<-c(1:ncol(predicted_for_loops_S))

k_S <- obs$total_alleles/2 ## total number individuals
y_S <- k_S-(obs$n_ss_individuals) ## total number of C- individuals

#Now loop through all the parameter combinations, calculate the likelihood and return
likelihoods<-c()
likelihoods_D<-c()
likelihoods_C<-c()
likelihoods_S<-c()
positions_D<-c()
positions_C<-c()
positions_S<-c()
stretch_factors_D<-c()
stretch_factors_C<-c()
stretch_factors_S<-c()

```

```

D_centres<-c()
C_centres<-c()
S_centres<-c()
D_cline_widths<-c()
C_cline_widths<-c()
S_cline_widths<-c()
D_max_slope<-c()
C_max_slope<-c()
S_max_slope<-c()

for (i in 1:nrow(predicted_D)){

pred_D <- data.frame(demes,t(predicted_for_loops_D[i,]))
pred_C <- data.frame(demes,t(predicted_for_loops_C[i,]))
pred_S <- data.frame(demes,t(predicted_for_loops_S[i,]))

names(pred_D) <- c("km_D","freq_D")
names(pred_C) <- c("km_C","freq_C")
names(pred_S) <- c("km_S","freq_S")

#optimisation algorithm,
# starting parameters = 1) align the approx middle of the hybrid zone with the middle of the simulated
# cline, and 2) using 1 as the stretch factor (i.e. no stretch: less than 1 squeezes the cline together, greater
# than 1 pulls it apart).
approx_centre_sim_D<-which.min(abs(pred_D$freq_D-0.5))
starting_param_1<-approx_mid_point-approx_centre_sim_D #this find the difference between the
observed data and simulated data using the D locus, and uses this as a starting point for each of the loci
likelihood_D <- optim(par = c(starting_param_1,1), func_likelihood_ERATO_LIKE_D, NULL,
control=list(fnscale=-1))
likelihood_C <- optim(par = c(starting_param_1,1), func_likelihood_ERATO_LIKE_C, NULL,
control=list(fnscale=-1))
likelihood_S <- optim(par = c(starting_param_1,1), func_likelihood_ERATO_LIKE_S, NULL,
control=list(fnscale=-1))
likelihood<-sum(likelihood_D$value,likelihood_C$value,likelihood_S$value)
#find middle of clines and widths
#this is a clunky bit to find the centre of the cline; aspline doesn't seem to be able to find the x value for
a given y value, only the other way round, so had to do it like this
interp.dat<-aspline((likelihood_D$par[2]*pred_D$km)+likelihood_D$par[1], pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
D_cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
D_cline_max_slope<-max_slope(interp.dat$y)
D_cline_width<-(1/D_cline_max_slope)/100 #divide by 100 to bring into km scale (10m was specified in
the interp.dat fine scale)
interp.dat<-aspline((likelihood_C$par[2]*pred_C$km)+likelihood_C$par[1], pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
C_cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
C_cline_max_slope<-max_slope(interp.dat$y)

```

```

C_cline_width<-(1/C_cline_max_slope)/100 #divide by 100 to bring into km scale (10m was specified in
the interp.dat fine scale)
interp.dat<-aspline((likelihood_S$par[2]*pred_S$km)+likelihood_S$par[1], pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
S_cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
S_cline_max_slope<-max_slope(interp.dat$y)
S_cline_width<-(1/S_cline_max_slope)/100 #divide by 100 to bring into km scale (10m was specified in
the interp.dat fine scale)

```

```

#append results to results vectors..

```

```

likelihoods<-append(likelihoods,likelihood)
likelihoods_D<-append(likelihoods_D,likelihood_D$value)
likelihoods_C<-append(likelihoods_C,likelihood_C$value)
likelihoods_S<-append(likelihoods_S,likelihood_S$value)
positions_D<-append(positions_D,likelihood_D$par[1])
positions_C<-append(positions_C,likelihood_C$par[1])
positions_S<-append(positions_S,likelihood_S$par[1])
stretch_factors_D<-append(stretch_factors_D,likelihood_D$par[2])
stretch_factors_C<-append(stretch_factors_C,likelihood_C$par[2])
stretch_factors_S<-append(stretch_factors_S,likelihood_S$par[2])
D_centres<-append(D_centres,D_cline_centre)
C_centres<-append(C_centres,C_cline_centre)
S_centres<-append(S_centres,S_cline_centre)
D_cline_widths<-append(D_cline_widths,D_cline_width)
C_cline_widths<-append(C_cline_widths,C_cline_width)
S_cline_widths<-append(S_cline_widths,S_cline_width)
D_max_slope<-append(D_max_slope,D_cline_max_slope*100)#times 100 to bring on to km scale
C_max_slope<-append(C_max_slope,C_cline_max_slope*100)#times 100 to bring on to km scale
S_max_slope<-append(S_max_slope,S_cline_max_slope*100)#times 100 to bring on to km scale

```

```

}

```

```

####

```

```

#NaNs may be produced - this is OK - its just when the Optimization tests weird starting parameters I
think.

```

```

#you may need to change 1:4 to get the right columns to paste into the results....

```

```

results<-
cbind(predicted_D[1:5],likelihoods,positions_D,positions_C,positions_S,stretch_factors_D,stretch_factor
s_C,stretch_factors_S,D_centres,C_centres,S_centres,D_cline_widths,C_cline_widths,S_cline_widths,D_
max_slope,C_max_slope,S_max_slope,likelihoods_D,likelihoods_C,likelihoods_S)

```

```

results<-results[order(-results$likelihoods),]

```

```

results[1:20,]

```

```

#Now extract the best fitting simulated cline for plotting etc (need to centre on zero again).

```

```

pred_D <- data.frame(demes,t(predicted_for_loops_D[row.names(results)[1,]]))
pred_C <- data.frame(demes,t(predicted_for_loops_C[row.names(results)[1,]]))
pred_S <- data.frame(demes,t(predicted_for_loops_S[row.names(results)[1,]]))

```

```

names(pred_D) <- c("km_D","freq_D")
names(pred_C) <- c("km_C","freq_C")
names(pred_S) <- c("km_S","freq_S")

```

```

par(mfrow=c(3,1))
plot(obs$km,obs$propD)
lines(pred_D$freq ~ l((results[1,10]*pred_D$km)+results[1,7]), col="red")
plot(obs$km,obs$propC)
lines(pred_C$freq ~ l((results[1,11]*pred_C$km)+results[1,8]), col="red")
plot(obs$km,obs$propS)
lines(pred_S$freq ~ l((results[1,12]*(pred_S$km)+results[1,9]), col="red")

```

THIS IS FOR FINDING SLOPE SUPPORT LIMITS

#this code plays with the stretch factor increasing (and then decreasing it) until the log likelihoods drop by -2.

#the code is quite long because each time you play with the stretch factor that moves the position of the centre of the cline, so you keep

#having to realign the cline and make the centre in the same place it was before.

#do it for the D cline

```

mid_point_for_pivot<-results[1,13]
positions<-results[1,7]
test.val.upper<-results[1,10]
while(func_likelihood_ERATO_LIKE_D(c(positions,test.val.upper)) > results[1,22] - 2 )
{test.val.upper<-test.val.upper+0.0001
interp.dat<-aspline((test.val.upper*pred_D$km)+positions, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_D(c(positions,test.val.upper)),sep=""))
interp.dat<-aspline((test.val.upper*pred_D$km)+positions, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
D_cline_slope_lower_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
D_cline_width_upper_interval<-(1/D_cline_slope_lower_interval)

```

```

mid_point_for_pivot<-results[1,13]
positions<-results[1,7]
test.val.lower<-results[1,10]
while(func_likelihood_ERATO_LIKE_D(c(positions,test.val.lower)) > results[1,22] - 2 )
{test.val.lower<-test.val.lower-0.0001
interp.dat<-aspline((test.val.lower*pred_D$km)+positions, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)

```

```

cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_D(c(positions,test.val.lower)),sep=""))
interp.dat<-aspline((test.val.lower*pred_D$km)+positions, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
D_cline_slope_upper_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
D_cline_width_lower_interval<-(1/D_cline_slope_upper_interval)

#do it for the C cline
mid_point_for_pivot<-results[1,14]
positions<-results[1,8]
test.val.upper<-results[1,11]
while(func_likelihood_ERATO_LIKE_C(c(positions,test.val.upper)) > results[1,23] - 2 )
{test.val.upper<-test.val.upper+0.0001
interp.dat<-aspline((test.val.upper*pred_C$km)+positions, pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_C(c(positions,test.val.upper)),sep=""))
interp.dat<-aspline((test.val.upper*pred_C$km)+positions, pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
C_cline_slope_lower_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
C_cline_width_upper_interval<-(1/C_cline_slope_lower_interval)

mid_point_for_pivot<-results[1,14]
positions<-results[1,8]
test.val.lower<-results[1,11]
while(func_likelihood_ERATO_LIKE_C(c(positions,test.val.lower)) > results[1,23] - 2 )
{test.val.lower<-test.val.lower-0.0001
interp.dat<-aspline((test.val.lower*pred_C$km)+positions, pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_C(c(positions,test.val.lower)),sep=""))
interp.dat<-aspline((test.val.lower*pred_C$km)+positions, pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
C_cline_slope_upper_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
C_cline_width_lower_interval<-(1/C_cline_slope_upper_interval)

#do it for the S cline
mid_point_for_pivot<-results[1,15]
positions<-results[1,9]

```

```

test.val.upper<-results[1,12]
while(func_likelihood_ERATO_LIKE_S(c(positions,test.val.upper)) > results[1,24] - 2 )
{test.val.upper<-test.val.upper+0.0001
interp.dat<-aspline((test.val.upper*pred_S$km)+positions, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_S(c(positions,test.val.upper)),sep=""))
interp.dat<-aspline((test.val.upper*pred_S$km)+positions, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
S_cline_slope_lower_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
S_cline_width_upper_interval<-(1/S_cline_slope_lower_interval)

mid_point_for_pivot<-results[1,15]
positions<-results[1,9]
test.val.lower<-results[1,12]
while(func_likelihood_ERATO_LIKE_S(c(positions,test.val.lower)) > results[1,24] - 2 )
{test.val.lower<-test.val.lower-0.0001
interp.dat<-aspline((test.val.lower*pred_S$km)+positions, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_S(c(positions,test.val.lower)),sep=""))
interp.dat<-aspline((test.val.lower*pred_S$km)+positions, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
S_cline_slope_upper_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
S_cline_width_lower_interval<-(1/S_cline_slope_upper_interval)

# THIS IS FOR FINDING CENTRE SUPPORT LIMITS
#do it for the D cline
test.val.upper<-results[1,7]
while(func_likelihood_ERATO_LIKE_D(c(test.val.upper,results[1,10])) > results[1,22] - 2 )
{test.val.upper<-test.val.upper+0.0001
}
print(paste("centre upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_D(c(test.val.upper,results[1,10])),sep=""))
test.val.lower<-results[1,7]
while(func_likelihood_ERATO_LIKE_D(c(test.val.lower,results[1,10])) > results[1,22] - 2 )
{test.val.lower<-test.val.lower-0.0001
}
print(paste("centre lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_D(c(test.val.lower,results[1,10])),sep=""))

```

```

interp.dat<-aspline((results[1,10]*pred_D$km)+test.val.upper, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
D_cline_centre_upper<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
interp.dat<-aspline((results[1,10]*pred_D$km)+test.val.lower, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
D_cline_centre_lower<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]

#do it for the C cline
test.val.upper<-results[1,8]
while(func_likelihoood_ERATO_LIKE_C(c(test.val.upper,results[1,11])) > results[1,23] - 2 )
{test.val.upper<-test.val.upper+0.0001
}
print(paste("centre upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihoood_ERATO_LIKE_C(c(test.val.upper,results[1,11])),sep=""))
test.val.lower<-results[1,8]
while(func_likelihoood_ERATO_LIKE_C(c(test.val.lower,results[1,11])) > results[1,23] - 2 )
{test.val.lower<-test.val.lower-0.0001
}
print(paste("centre lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihoood_ERATO_LIKE_C(c(test.val.lower,results[1,11])),sep=""))

interp.dat<-aspline((results[1,11]*pred_C$km)+test.val.upper, pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
C_cline_centre_upper<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
interp.dat<-aspline((results[1,11]*pred_C$km)+test.val.lower, pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
C_cline_centre_lower<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]

#do it for the S cline
test.val.upper<-results[1,9]
while(func_likelihoood_ERATO_LIKE_S(c(test.val.upper,results[1,12])) > results[1,24] - 2 )
{test.val.upper<-test.val.upper+0.0001
}
print(paste("centre upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihoood_ERATO_LIKE_S(c(test.val.upper,results[1,12])),sep=""))
test.val.lower<-results[1,9]
while(func_likelihoood_ERATO_LIKE_S(c(test.val.lower,results[1,12])) > results[1,24] - 2 )
{test.val.lower<-test.val.lower-0.0001
}
print(paste("centre lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihoood_ERATO_LIKE_S(c(test.val.lower,results[1,12])),sep=""))

interp.dat<-aspline((results[1,12]*pred_S$km)+test.val.upper, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
S_cline_centre_upper<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
interp.dat<-aspline((results[1,12]*pred_S$km)+test.val.lower, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
S_cline_centre_lower<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]

```

```

results_for_best_fitting_cline<-results[1,]
results_for_best_fitting_cline<-
cbind(results_for_best_fitting_cline,D_cline_slope_lower_interval,D_cline_slope_upper_interval,D_cline_width_lower_interval,D_cline_width_upper_interval)
results_for_best_fitting_cline<-
cbind(results_for_best_fitting_cline,C_cline_slope_lower_interval,C_cline_slope_upper_interval,C_cline_width_lower_interval,C_cline_width_upper_interval)
results_for_best_fitting_cline<-
cbind(results_for_best_fitting_cline,S_cline_slope_lower_interval,S_cline_slope_upper_interval,S_cline_width_lower_interval,S_cline_width_upper_interval)
results_for_best_fitting_cline<-
cbind(results_for_best_fitting_cline,D_cline_centre_upper,D_cline_centre_lower,C_cline_centre_upper,C_cline_centre_lower,S_cline_centre_upper,S_cline_centre_lower)

```

```

results_for_best_fitting_cline<-results_for_best_fitting_cline[-(c(1,7:12))]

```

#FIGURE COMBINED WITH SEPARATE POINTS FOR TIME PERIODS

```

par(mfrow=c(4,1),mar=c(2.5, 3, 1, 1.5),mgp=c(1.5,0.5,0))
plot(obsJM$km,obsJM$propD,col="white",pch=1,ylab=expression(paste(italic(D),"
frequency"))),xlab="Transect position (km)")
legend("topleft",inset=c(0.025,0.05),c("1986","2011"),pch=c(1,4),col=c("blue","red"))
lines(pred_D$freq_D ~ I((results[1,10]*pred_D$km_D)+results[1,7]), col="black")
points(obsJM$km,obsJM$propD,col="blue",pch=1,ylab=expression(paste(italic(D),"
frequency"))),xlab="Transect position (km)")
points(obsNR$km,obsNR$propD,col="red",pch=4)
plot(obsJM$km,obsJM$propC,col="white",pch=1,ylab=expression(paste(italic(Cr),"
frequency"))),xlab="Transect position (km)")
lines(pred_C$freq_C ~ I((results[1,11]*pred_C$km_C)+results[1,8]), col="black")
points(obsJM$km,obsJM$propC,col="blue",pch=1,ylab=expression(paste(italic(Cr))),xlab="Transect
position (km)")
points(obsNR$km,obsNR$propC,col="red",pch=4)
plot(obsJM$km,obsJM$propS,col="white",pch=1,ylab=expression(paste(italic(Sd),"
frequency"))),xlab="Transect position (km)")
lines(pred_S$freq_S ~ I((results[1,12]*pred_S$km_S)+results[1,9]), col="black")
points(obsJM$km,obsJM$propS,col="blue",pch=1,ylab=expression(paste(italic(Sd))),xlab="Transect
position (km)")
points(obsNR$km,obsNR$propS,col="red",pch=4)

```

#BLUBBLE PLOT FIGURE COMBINED WITH SEPARATE POINTS FOR TIME PERIODS

```

par(mfrow=c(4,1),mar=c(2.5, 3, 1.5, 1.5),mgp=c(1.5,0.5,0))
plot(obsJM$km,obsJM$propD,col="white",pch=1,ylab=expression(paste(italic(D),"
frequency"))),xlab="Transect position (km)",,main="FIGURE 2A")
legend("topleft",inset=c(0.025,0.05),c("1986","2011"),pch=c(1,1),col=c("blue","red"))

```

```

legend("bottomright",inset=c(0.025,0.05),c("1","10","50"),pch=1,pt.cex=c(0.53103,1.53103,2.23),col=c("
black"),title=expression(paste(italic(n))))
lines(pred_D$freq_D ~ I((results[1,10]*pred_D$km_D)+results[1,7]), col="black")
points(obsJM$km,obsJM$propD,col="blue",pch=1,cex=0.23+c(log10(obsJM$total_alleles)))
points(obsNR$km,obsNR$propD,col="red",pch=1,cex=0.23+c(log10(obsNR$total_alleles)))
plot(obsJM$km,obsJM$propC,col="white",pch=1,ylab=expression(paste(italic(Cr),"
frequency")),xlab="Transect position (km)")
lines(pred_C$freq_C ~ I((results[1,11]*pred_C$km_C)+results[1,8]), col="black")
points(obsJM$km,obsJM$propC,col="blue",pch=1,cex=c(0.23+log10(obsJM$total_alleles)))
points(obsNR$km,obsNR$propC,col="red",pch=1,cex=c(0.23+log10(obsNR$total_alleles)))
plot(obsJM$km,obsJM$propS,col="white",pch=1,ylab=expression(paste(italic(Sd),"
frequency")),xlab="Transect position (km)")
lines(pred_S$freq_S ~ I((results[1,12]*pred_S$km_S)+results[1,9]), col="black")
points(obsJM$km,obsJM$propS,col="blue",pch=1,cex=c(0.23+log10(obsJM$total_alleles)))
points(obsNR$km,obsNR$propS,col="red",pch=1,cex=c(0.23+log10(obsNR$total_alleles)))

```

```

#####
###USE LIKELIHOOD TO FIT THE SIMULATED MELPOMENE CLINES TO THE OBSERVED DATA###
#####

```

```

library(akima)

```

```

func_likelihood_MELP_B <- function(x) {
  freqs_B <- aspline((x[2]*pred_B$km)+x[1], pred_B$freq, xout=obs$km, degree=10)
  p_B <- ((freqs_B$y*freqs_B$y)+(2*(freqs_B$y*(1-freqs_B$y))))
  p_B[p_B==0] <- 1e-55
  p_B[p_B==1] <- 1-1e-5
  #had to divide likelihood calculation up as R stored very small probabilities as zero (before was coded
  as likelihood_B<-sum((log(p_B^y_B))+ log((1-p_B)^(k_B-y_B))) - see erato script)
  part_a<-p_B^y_B
  part_b<-(1-p_B)^(k_B-y_B)
  part_a[part_a==0] <- 1e-55
  part_b[part_b==0] <- 1e-55
  likelihood<-sum(log(part_a)+log(part_b))
  return(likelihood)
}

```

```

func_likelihood_MELP_D <- function(x) {
  freqs_D <- aspline((x[2]*pred_D$km)+x[1], pred_D$freq, xout=obs$km, degree=10)
  p_D <- ((freqs_D$y*freqs_D$y)+(2*(freqs_D$y*(1-freqs_D$y))))
  p_D[p_D==0] <- 1e-55
  p_D[p_D==1] <- 1-1e-5
  part_a<-p_D^y_D
  part_b<-(1-p_D)^(k_D-y_D)
  part_a[part_a==0] <- 1e-55
  part_b[part_b==0] <- 1e-55
  likelihood<-sum(log(part_a)+log(part_b))
}

```

```

return(likelihood)
}

func_likelihood_MELP_N <- function(x) {
  freqs_N <- aspline((x[2]*pred_N$km)+x[1], pred_N$freq, xout=obs$km, degree=10)
  p_N <- ((freqs_N$y*freqs_N$y)+(2*(freqs_N$y*(1-freqs_N$y))))
  p_N[p_N==0] <- 1e-55
  p_N[p_N==1] <- 1-1e-5
  part_a<-p_N^y_N
  part_b<-(1-p_N)^(k_N-y_N)
  part_a[part_a==0] <- 1e-55
  part_b[part_b==0] <- 1e-55
  likelihood<-sum(log(part_a)+log(part_b))
  return(likelihood)
}

func_likelihood_MELP_Y <- function(x) {
  freqs_Y <- aspline((x[2]*pred_Y$km)+x[1], pred_Y$freq, xout=obs$km, degree=10)
  p_Y <- ((freqs_Y$y*freqs_Y$y)+(2*(freqs_Y$y*(1-freqs_Y$y))))
  p_Y[p_Y==0] <- 1e-55
  p_Y[p_Y==1] <- 1-1e-5
  part_a<-p_Y^y_Y
  part_b<-(1-p_Y)^(k_Y-y_Y)
  part_a[part_a==0] <- 1e-55
  part_b[part_b==0] <- 1e-55
  likelihood<-sum(log(part_a)+log(part_b))
  return(likelihood)
}

#Find max slope of simulated cline function
max_slope <- function(x) {
  slopes<-c()
  for (i in 1:(length(x)-1)){
    slopes<-append(slopes,x[i+1]-x[i])
  }
  return(max(slopes))
}

#observed data
obsJM <- read.csv("melpomene1990.csv", header=T, sep=",")
obsNR <- read.csv("melpomene2011.csv", header=T, sep=",")
combined <- read.csv("melpomene_combined.csv", header=T, sep=",")
predicted_all_data <- read.csv("melp_sim_results.csv", header=T, sep=",")
predicted_all_data<-
subset(predicted_all_data,selectionB==0.6&selectionD==0.6&selectionN==0.6&selectionY==0.6&recom
b_r_BD==0.135)

#Approximate point of the middle of the hybrid zone on the transect (this is used as a starting point in
the optimization algorithm) (123.57 is Mallet 1990 estimated centre of D cline).

```

```
approx_mid_point<-123.57
```

```
#change this to the date set you want to analyse
```

```
obs<-combined
```

```
obs <- obs[order(obs$km),]
```

```
#Subset the B locus
```

```
predicted_B<-subset(predicted_all_data, predicted_all_data$locus == "B")
```

```
rownames(predicted_B) <- seq(length=nrow(predicted_B))
```

```
#this predicted for loops is just the allele frequencies without the row details, so you may have to  
change the "5" depending on the input file
```

```
predicted_for_loops_B<-predicted_B[10:ncol(predicted_B)]
```

```
demes<-c(1:ncol(predicted_for_loops_B))
```

```
k_B <- obs$total_alleles/2 ## total number individuals
```

```
y_B <- k_B-(obs$n_bb_individuals) ## total number of B- individuals
```

```
#Subset the D locus
```

```
predicted_D<-subset(predicted_all_data, predicted_all_data$locus == "D")
```

```
rownames(predicted_D) <- seq(length=nrow(predicted_D))
```

```
#this predicted for loops is just the allele frequencies without the row details, so you may have to  
change the "5" depending on the input file
```

```
predicted_for_loops_D<-predicted_D[10:ncol(predicted_D)]
```

```
demes<-c(1:ncol(predicted_for_loops_D))
```

```
k_D <- obs$total_alleles/2 ## total number individuals
```

```
y_D <- k_D-(obs$n_dd_individuals) ## total number of D- individuals
```

```
#Subset the N locus
```

```
predicted_N<-subset(predicted_all_data, predicted_all_data$locus == "N")
```

```
rownames(predicted_N) <- seq(length=nrow(predicted_N))
```

```
#this predicted for loops is just the allele frequencies without the row details, so you may have to  
change the "5" depending on the input file
```

```
predicted_for_loops_N<-predicted_N[10:ncol(predicted_N)]
```

```
demes<-c(1:ncol(predicted_for_loops_N))
```

```
k_N <- obs$total_alleles/2 ## total number individuals
```

```
y_N <- k_N-(obs$n_nn_individuals) ## total number of N- individuals
```

```
#Subset the Y locus (predicted cline is the same as the N locus assuming very tight linkage)...
```

```
predicted_Y<-subset(predicted_all_data, predicted_all_data$locus == "Y")
```

```
rownames(predicted_Y) <- seq(length=nrow(predicted_Y))
```

```
#this predicted for loops is just the allele frequencies without the row details, so you may have to  
change the "5" depending on the input file
```

```

predicted_for_loops_Y<-predicted_Y[10:ncol(predicted_Y)]
demes<-c(1:ncol(predicted_for_loops_Y))

k_Y <- obs$total_alleles/2 ## total number individuals
y_Y <- k_Y-(obs$n_yy_individuals) ## total number of Y- individuals

#Now loop through all the parameter combinations, calculate the likelihood and return
likelihoods<-c()
likelihoods_B<-c()
likelihoods_D<-c()
likelihoods_N<-c()
likelihoods_Y<-c()
positions_B<-c()
positions_D<-c()
positions_N<-c()
positions_Y<-c()
stretch_factors_B<-c()
stretch_factors_D<-c()
stretch_factors_N<-c()
stretch_factors_Y<-c()
B_centres<-c()
D_centres<-c()
N_centres<-c()
Y_centres<-c()
B_cline_widths<-c()
D_cline_widths<-c()
N_cline_widths<-c()
Y_cline_widths<-c()
B_max_slope<-c()
D_max_slope<-c()
N_max_slope<-c()
Y_max_slope<-c()

for (i in 1:nrow(predicted_B)){

pred_B <- data.frame(demes,t(predicted_for_loops_B[i,]))
pred_D <- data.frame(demes,t(predicted_for_loops_D[i,]))
pred_N <- data.frame(demes,t(predicted_for_loops_N[i,]))
pred_Y <- data.frame(demes,t(predicted_for_loops_Y[i,]))

names(pred_B) <- c("km_B","freq_B")
names(pred_D) <- c("km_D","freq_D")
names(pred_N) <- c("km_N","freq_N")
names(pred_Y) <- c("km_Y","freq_Y")

#optimisation algorithm,

```

starting parameters = 1) align the approx middle of the hybrid zone with the middle of the simulated cline, and 2) using 1 as the stretch factor (i.e. no stretch: less than 1 squeezes the cline together, greater than 1 pulls it apart).

```
approx_centre_sim_D<-which.min(abs(pred_D$freq_D-0.5))
```

```
starting_param_1<-approx_mid_point-approx_centre_sim_D #this find the difference between the  
observed data and simulated data using the D locus, and uses this as a starting point
```

```
likelihood_B <- optim(par = c(starting_param_1,1), func_likelihood_MELP_B, NULL, control=list(fnscale=-  
1))
```

```
likelihood_D <- optim(par = c(starting_param_1,1), func_likelihood_MELP_D, NULL,  
control=list(fnscale=-1))
```

```
likelihood_N <- optim(par = c(starting_param_1,1), func_likelihood_MELP_N, NULL,  
control=list(fnscale=-1))
```

```
likelihood_Y <- optim(par = c(starting_param_1,1), func_likelihood_MELP_Y, NULL, control=list(fnscale=-  
1))
```

```
likelihood<-sum(likelihood_B$value,likelihood_D$value,likelihood_N$value,likelihood_Y$value)
```

```
#find middle of clines and widths
```

```
#this is a clunky bit to find the centre of the cline; the spline doesn't seem to be able to find the x value  
for a given y value, only the other way round, so had to do it like this
```

```
interp.dat<-aspline((likelihood_B$par[2]*pred_B$km)+likelihood_B$par[1], pred_B$freq, xout=  
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
```

```
B_cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
```

```
B_cline_max_slope<-max_slope(1-interp.dat$y)
```

```
B_cline_width<-(1/B_cline_max_slope)/100 #divide by 100 to bring into km scale (10m was specified in  
the interp.dat fine scale)
```

```
interp.dat<-aspline((likelihood_D$par[2]*pred_D$km)+likelihood_D$par[1], pred_D$freq, xout=
```

```
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
```

```
D_cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
```

```
D_cline_max_slope<-max_slope(interp.dat$y)
```

```
D_cline_width<-(1/D_cline_max_slope)/100 #divide by 100 to bring into km scale (10m was specified in  
the interp.dat fine scale)
```

```
interp.dat<-aspline((likelihood_N$par[2]*pred_N$km)+likelihood_N$par[1], pred_N$freq, xout=
```

```
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
```

```
N_cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
```

```
N_cline_max_slope<-max_slope(interp.dat$y)
```

```
N_cline_width<-(1/N_cline_max_slope)/100 #divide by 100 to bring into km scale (10m was specified in  
the interp.dat fine scale)
```

```
interp.dat<-aspline((likelihood_Y$par[2]*pred_Y$km)+likelihood_Y$par[1], pred_Y$freq, xout=
```

```
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
```

```
Y_cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
```

```
Y_cline_max_slope<-max_slope(interp.dat$y)
```

```
Y_cline_width<-(1/Y_cline_max_slope)/100 #divide by 100 to bring into km scale (10m was specified in  
the interp.dat fine scale)
```

```
#append results to results vectors..
```

```
likelihoods<-append(likelihoods,likelihood)
```

```

likelihoods_B<-append(likelihoods_B,likelihood_B$value)
likelihoods_D<-append(likelihoods_D,likelihood_D$value)
likelihoods_N<-append(likelihoods_N,likelihood_N$value)
likelihoods_Y<-append(likelihoods_Y,likelihood_Y$value)
positions_B<-append(positions_B,likelihood_B$par[1])
positions_D<-append(positions_D,likelihood_D$par[1])
positions_N<-append(positions_N,likelihood_N$par[1])
positions_Y<-append(positions_Y,likelihood_Y$par[1])
stretch_factors_B<-append(stretch_factors_B,likelihood_B$par[2])
stretch_factors_D<-append(stretch_factors_D,likelihood_D$par[2])
stretch_factors_N<-append(stretch_factors_N,likelihood_N$par[2])
stretch_factors_Y<-append(stretch_factors_Y,likelihood_Y$par[2])
B_centres<-append(B_centres,B_cline_centre)
D_centres<-append(D_centres,D_cline_centre)
N_centres<-append(N_centres,N_cline_centre)
Y_centres<-append(Y_centres,Y_cline_centre)
B_cline_widths<-append(B_cline_widths,B_cline_width)
D_cline_widths<-append(D_cline_widths,D_cline_width)
N_cline_widths<-append(N_cline_widths,N_cline_width)
Y_cline_widths<-append(Y_cline_widths,Y_cline_width)
B_max_slope<-append(B_max_slope,B_cline_max_slope*100)#times 100 to bring on to km scale
D_max_slope<-append(D_max_slope,D_cline_max_slope*100)#times 100 to bring on to km scale
N_max_slope<-append(N_max_slope,N_cline_max_slope*100)#times 100 to bring on to km scale
Y_max_slope<-append(Y_max_slope,Y_cline_max_slope*100)#times 100 to bring on to km scale

}

results<-
cbind(predicted_D[1:9],likelihoods,likelihoods_B,likelihoods_D,likelihoods_N,likelihoods_Y,positions_B,
positions_D,positions_N,positions_Y,stretch_factors_B,stretch_factors_D,stretch_factors_N,stretch_factors_Y,B_centres,D_centres,N_centres,Y_centres,B_cline_widths,D_cline_widths,N_cline_widths,Y_cline_widths,B_max_slope,D_max_slope,N_max_slope,Y_max_slope)

results<-results[order(-results$likelihoods),]

#put in the row for the most likely selection values here...
pred_B <- data.frame(demes,t(predicted_for_loops_B[row.names(results)[1],]))
pred_D <- data.frame(demes,t(predicted_for_loops_D[row.names(results)[1],]))
pred_N <- data.frame(demes,t(predicted_for_loops_N[row.names(results)[1],]))
pred_Y <- data.frame(demes,t(predicted_for_loops_Y[row.names(results)[1],]))

names(pred_B) <- c("km_B","freq_B")
names(pred_D) <- c("km_D","freq_D")
names(pred_N) <- c("km_N","freq_N")
names(pred_Y) <- c("km_Y","freq_Y")

#FIGURE FOR COMBINED WITH SEPARATE POINTS FOR TIME PERIODS
par(mfrow=c(4,1),mar=c(2.5, 3, 1, 1.5),mgp=c(1.5,0.5,0))

```

```

plot(obsJM$km,obsJM$propB,col="white",pch=1,ylab=expression(paste(italic(b),"
frequency"))),xlab="Transect position (km)")
legend("topleft",inset=c(0.025,0.05),c("1986","2011"),pch=c(1,4),col=c("blue","red"))
lines(1-pred_B$freq_B ~ I((results[1,19]*pred_B$km_B)+results[1,15]), col="black")
points(obsNR$km,obsNR$propB,col="red",pch=4)
points(obsJM$km,obsJM$propB,col="blue",pch=1)

```

```

plot(obsJM$km,obsJM$propD,col="white",pch=1,ylab=expression(paste(italic(D),"
frequency"))),xlab="Transect position (km)")
lines(pred_D$freq_D ~ I((results[1,20]*pred_D$km_D)+results[1,16]), col="black")
points(obsJM$km,obsJM$propD,col="blue",pch=1)
points(obsNR$km,obsNR$propD,col="red",pch=4)

```

```

plot(obsJM$km,obsJM$propN,col="white",pch=1,ylab=expression(paste(italic(N),"
frequency"))),xlab="Transect position (km)")
lines(pred_N$freq_N ~ I((results[1,21]*pred_N$km_N)+results[1,17]), col="black")
points(obsJM$km,obsJM$propN,col="blue",pch=1)
points(obsNR$km,obsNR$propN,col="red",pch=4)

```

```

plot(obsJM$km,obsJM$propY,col="white",pch=1,ylab=expression(paste(italic(Y),"
frequency"))),xlab="Transect position (km)")
points(obsJM$km,obsJM$propY,col="blue",pch=1)
points(obsNR$km,obsNR$propY,col="red",pch=4)
lines(pred_Y$freq_Y ~ I((results[1,22]*pred_Y$km_Y)+results[1,18]), col="black")

```

#FIGURE FOR COMBINED WITH SEPARATE POINTS FOR TIME PERIODS for the B cline (slope and and centres inserted from other data)

```

par(mfrow=c(4,1),mar=c(2.5, 3, 1, 1.5),mgp=c(1.5,0.5,0))
plot(obsJM$km,obsJM$propB,col="white",pch=1,ylab=expression(paste(italic(b),"
frequency"))),xlab="Transect position (km)")
legend("topleft",inset=c(0.025,0.05),c("1986","2011"),pch=c(1,4),col=c("blue","red"))
lines(1-pred_B$freq_B ~ I((2.285966*pred_B$km_B)+-107.30842), col="blue")
lines(1-pred_B$freq_B ~ I((1.377966*pred_B$km_B)+ -16.42842), col="red")
points(obsJM$km,obsJM$propB,col="blue",pch=1)
points(obsNR$km,obsNR$propB,col="red",pch=4)

```

```

plot(obsJM$km,obsJM$propD,col="white",pch=1,ylab=expression(paste(italic(D),"
frequency"))),xlab="Transect position (km)")
lines(pred_D$freq_D ~ I((results[1,20]*pred_D$km_D)+results[1,16]), col="black")
points(obsJM$km,obsJM$propD,col="blue",pch=1)
points(obsNR$km,obsNR$propD,col="red",pch=4)

```

```

plot(obsJM$km,obsJM$propN,col="white",pch=1,ylab=expression(paste(italic(N),"
frequency"))),xlab="Transect position (km)")
lines(pred_N$freq_N ~ I((results[1,21]*pred_N$km_N)+results[1,17]), col="black")
points(obsJM$km,obsJM$propN,col="blue",pch=1)
points(obsNR$km,obsNR$propN,col="red",pch=4)

```

```

plot(obsJM$km,obsJM$propY,col="white",pch=1,ylab=expression(paste(italic(Yb),"
frequency"))),xlab="Transect position (km)")
lines(pred_Y$freq_Y ~ I((results[1,22]*pred_Y$km_Y)+results[1,18]), col="black")
points(obsJM$km,obsJM$propY,col="blue",pch=1,ylab="Yb frequency",xlab="Transect position (km)")
points(obsNR$km,obsNR$propY,col="red",pch=4)

```

```

#BUBBLE PLOT FIGURE FOR COMBINED WITH SEPARATE POINTS FOR TIME PERIODS for the B cline
(slope and and centres inserted from other data)
#0.23 added to cex to make smallest points little circles rather than dots.
par(mfrow=c(4,1),mar=c(2.5, 3, 1.5, 1.5),mgp=c(1.5,0.5,0))
plot(obsJM$km,obsJM$propB,col="white",pch=1,ylab=expression(paste(italic(b),"
frequency"))),xlab="Transect position (km)",main="FIGURE 2B")
legend("topleft",inset=c(0.025,0.05),c("1986", "2011"),pch=c(1,1),col=c("blue","red"))
legend("bottomright",inset=c(0.025,0.05),c("1", "10", "50"),pch=1,pt.cex=c(0.53103,1.53103,2.23),col=c("
black"),title=expression(paste(italic(n))))
lines(1-pred_B$freq_B ~ I((2.285966*pred_B$km_B)+-107.30842), col="blue")
lines(1-pred_B$freq_B ~ I((1.377966*pred_B$km_B)+ -16.42842), col="red")
points(obsJM$km,obsJM$propB,col="blue",pch=1,cex=c(0.23+log10(obsJM$total_alleles)))
points(obsNR$km,obsNR$propB,col="red",pch=1,cex=c(0.23+log10(obsNR$total_alleles)))

```

```

plot(obsJM$km,obsJM$propD,col="white",pch=1,ylab=expression(paste(italic(D),"
frequency"))),xlab="Transect position (km)")
lines(pred_D$freq_D ~ I((results[1,20]*pred_D$km_D)+results[1,16]), col="black")
points(obsJM$km,obsJM$propD,col="blue",pch=1,cex=c(0.23+log10(obsJM$total_alleles)))
points(obsNR$km,obsNR$propD,col="red",pch=1,cex=c(0.23+log10(obsNR$total_alleles)))

```

```

plot(obsJM$km,obsJM$propN,col="white",pch=1,ylab=expression(paste(italic(N),"
frequency"))),xlab="Transect position (km)")
lines(pred_N$freq_N ~ I((results[1,21]*pred_N$km_N)+results[1,17]), col="black")
points(obsJM$km,obsJM$propN,col="blue",pch=1,cex=c(0.23+log10(obsJM$total_alleles)))
points(obsNR$km,obsNR$propN,col="red",pch=1,cex=c(0.23+log10(obsNR$total_alleles)))

```

```

plot(obsJM$km,obsJM$propY,col="white",pch=1,ylab=expression(paste(italic(Yb),"
frequency"))),xlab="Transect position (km)")
lines(pred_Y$freq_Y ~ I((results[1,22]*pred_Y$km_Y)+results[1,18]), col="black")
points(obsJM$km,obsJM$propY,col="blue",pch=1,cex=c(0.23+log10(obsJM$total_alleles)))
points(obsNR$km,obsNR$propY,col="red",pch=1,cex=c(0.23+log10(obsNR$total_alleles)))

```

```

# THIS IS FOR FINDING SLOPE/ WIDTH / CENTRE SUPPORT LIMITS FOR THE BEST FITTING CLINE
#this code plays with the stretch factor increasing (and then decreasing it) until the log likelihoods drop
by -2.
#the code is quite long because each time you play with the stretch factor that moves the position of the
centre of the cline, so you keep
#having to realign the cline and make the centre in the same place it was before.

```

```

results<-results[1,]

```

```

#do it for the B cline
mid_point_for_pivot<-results$B_centres
positions<-results$positions_B
test.val.upper<-results$stretch_factors_B
best_fit_LL<-func_likelihood_MELP_B(c(positions,results$stretch_factors_B))
while(func_likelihood_MELP_B(c(positions,test.val.upper)) > best_fit_LL - 2 )
{test.val.upper<-test.val.upper+0.0001
interp.dat<-aspline((test.val.upper*pred_B$km)+positions, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_B(c(positions,test.val.upper)),sep=""))
interp.dat<-aspline((test.val.upper*pred_B$km)+positions, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
B_cline_slope_lower_interval<-max_slope(1-interp.dat$y)*100 #times by 100 to bring onto km scale
B_cline_width_upper_interval<-(1/B_cline_slope_lower_interval)

mid_point_for_pivot<-results$B_centres
positions<-results$positions_B
test.val.lower<-results$stretch_factors_B
best_fit_LL<-func_likelihood_MELP_B(c(positions,results$stretch_factors_B))
while(func_likelihood_MELP_B(c(positions,test.val.lower)) > best_fit_LL - 2 )
{test.val.lower<-test.val.lower-0.0001
interp.dat<-aspline((test.val.lower*pred_B$km)+positions, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_B(c(positions,test.val.lower)),sep=""))
interp.dat<-aspline((test.val.lower*pred_B$km)+positions, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
B_cline_slope_upper_interval<-max_slope(1-interp.dat$y)*100 #times by 100 to bring onto km scale
B_cline_width_lower_interval<-(1/B_cline_slope_upper_interval)

#do it for the D cline
mid_point_for_pivot<-results$D_centres
positions<-results$positions_D
test.val.upper<-results$stretch_factors_D
best_fit_LL<-func_likelihood_MELP_D(c(positions,results$stretch_factors_D))
while(func_likelihood_MELP_D(c(positions,test.val.upper)) > best_fit_LL - 2 )
{test.val.upper<-test.val.upper+0.0001
interp.dat<-aspline((test.val.upper*pred_D$km)+positions, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}

```

```

}
print(paste("stretch upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_D(c(positions,test.val.upper)),sep=""))
interp.dat<-aspline((test.val.upper*pred_D$km)+positions, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
D_cline_slope_lower_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
D_cline_width_upper_interval<-(1/D_cline_slope_lower_interval)

mid_point_for_pivot<-results$D_centres
positions<-results$positions_D
test.val.lower<-results$stretch_factors_D
best_fit_LL<-func_likelihood_MELP_D(c(positions,results$stretch_factors_D))
while(func_likelihood_MELP_D(c(positions,test.val.lower)) > best_fit_LL - 2 )
{test.val.lower<-test.val.lower-0.0001
interp.dat<-aspline((test.val.lower*pred_D$km)+positions, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_D(c(positions,test.val.lower)),sep=""))
interp.dat<-aspline((test.val.lower*pred_D$km)+positions, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
D_cline_slope_upper_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
D_cline_width_lower_interval<-(1/D_cline_slope_upper_interval)

#do it for the N cline
mid_point_for_pivot<-results$N_centres
positions<-results$positions_N
test.val.upper<-results$stretch_factors_N
best_fit_LL<-func_likelihood_MELP_N(c(positions,results$stretch_factors_N))
while(func_likelihood_MELP_N(c(positions,test.val.upper)) > best_fit_LL - 2 )
{test.val.upper<-test.val.upper+0.0001
interp.dat<-aspline((test.val.upper*pred_N$km)+positions, pred_N$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_N(c(positions,test.val.upper)),sep=""))
interp.dat<-aspline((test.val.upper*pred_N$km)+positions, pred_N$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
N_cline_slope_lower_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
N_cline_width_upper_interval<-(1/N_cline_slope_lower_interval)

mid_point_for_pivot<-results$N_centres
positions<-results$positions_N
test.val.lower<-results$stretch_factors_N

```

```

best_fit_LL<-func_likelihood_MELP_N(c(positions,results$stretch_factors_N))
while(func_likelihood_MELP_N(c(positions,test.val.lower)) > best_fit_LL - 2 )
{test.val.lower<-test.val.lower-0.0001
interp.dat<-aspline((test.val.lower*pred_N$km)+positions, pred_N$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_N(c(positions,test.val.lower)),sep=""))
interp.dat<-aspline((test.val.lower*pred_N$km)+positions, pred_N$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
N_cline_slope_upper_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
N_cline_width_lower_interval<-(1/N_cline_slope_upper_interval)

#do it for the Y cline
mid_point_for_pivot<-results$Y_centres
positions<-results$positions_Y
test.val.upper<-results$stretch_factors_Y
best_fit_LL<-func_likelihood_MELP_Y(c(positions,results$stretch_factors_Y))
while(func_likelihood_MELP_Y(c(positions,test.val.upper)) > best_fit_LL - 2 )
{test.val.upper<-test.val.upper+0.0001
interp.dat<-aspline((test.val.upper*pred_Y$km)+positions, pred_Y$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_Y(c(positions,test.val.upper)),sep=""))
interp.dat<-aspline((test.val.upper*pred_Y$km)+positions, pred_Y$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
Y_cline_slope_lower_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
Y_cline_width_upper_interval<-(1/Y_cline_slope_lower_interval)

mid_point_for_pivot<-results$Y_centres
positions<-results$positions_Y
test.val.lower<-results$stretch_factors_Y
best_fit_LL<-func_likelihood_MELP_Y(c(positions,results$stretch_factors_Y))
while(func_likelihood_MELP_Y(c(positions,test.val.lower)) > best_fit_LL - 2 )
{test.val.lower<-test.val.lower-0.0001
interp.dat<-aspline((test.val.lower*pred_Y$km)+positions, pred_Y$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_Y(c(positions,test.val.lower)),sep=""))

```

```

interp.dat<-aspline((test.val.lower*pred_Y$km)+positions, pred_Y$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
Y_cline_slope_upper_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
Y_cline_width_lower_interval<-(1/Y_cline_slope_upper_interval)

```

THIS IS FOR FINDING CENTRE SUPPORT LIMITS

```

#do it for the B cline
test.val.upper<-results$positions_B
best_fit_LL<-func_likelihood_MELP_B(c(results$positions_B,results$stretch_factors_B))
while(func_likelihood_MELP_B(c(test.val.upper,results$stretch_factors_B)) > best_fit_LL - 2 )
{test.val.upper<-test.val.upper+0.0001
}
print(paste("centre upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_B(c(test.val.upper,results$stretch_factors_B)),sep=""))
test.val.lower<-results$positions_B
while(func_likelihood_MELP_B(c(test.val.lower,results$stretch_factors_B)) > best_fit_LL - 2 )
{test.val.lower<-test.val.lower-0.0001
}
print(paste("centre lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_B(c(test.val.lower,results$stretch_factors_B)),sep=""))

```

```

interp.dat<-aspline((results$stretch_factors_B*pred_B$km)+test.val.upper, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
B_cline_centre_upper<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
interp.dat<-aspline((results$stretch_factors_B*pred_B$km)+test.val.lower, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
B_cline_centre_lower<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]

```

```

#do it for the D cline
test.val.upper<-results$positions_D
best_fit_LL<-func_likelihood_MELP_D(c(results$positions_D,results$stretch_factors_D))
while(func_likelihood_MELP_D(c(test.val.upper,results$stretch_factors_D)) > best_fit_LL - 2 )
{test.val.upper<-test.val.upper+0.0001
}
print(paste("centre upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_D(c(test.val.upper,results$stretch_factors_D)),sep=""))
test.val.lower<-results$positions_D
while(func_likelihood_MELP_D(c(test.val.lower,results$stretch_factors_D)) > best_fit_LL - 2 )
{test.val.lower<-test.val.lower-0.0001
}
print(paste("centre lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_D(c(test.val.lower,results$stretch_factors_D)),sep=""))

```

```

interp.dat<-aspline((results$stretch_factors_D*pred_D$km)+test.val.upper, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
D_cline_centre_upper<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]

```

```

interp.dat<-aspline((results$stretch_factors_D*pred_D$km)+test.val.lower, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
D_cline_centre_lower<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]

```

#do it for the N cline

```

test.val.upper<-results$positions_N
best_fit_LL<-func_likelihood_MELP_N(c(results$positions_N,results$stretch_factors_N))
while(func_likelihood_MELP_N(c(test.val.upper,results$stretch_factors_N)) > best_fit_LL - 2 )
{test.val.upper<-test.val.upper+0.0001
}
print(paste("centre upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_N(c(test.val.upper,results$stretch_factors_N)),sep=""))
test.val.lower<-results$positions_N
while(func_likelihood_MELP_N(c(test.val.lower,results$stretch_factors_N)) > best_fit_LL - 2 )
{test.val.lower<-test.val.lower-0.0001
}
print(paste("centre lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_N(c(test.val.lower,results$stretch_factors_N)),sep=""))

```

```

interp.dat<-aspline((results$stretch_factors_N*pred_N$km)+test.val.upper, pred_N$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
N_cline_centre_upper<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
interp.dat<-aspline((results$stretch_factors_N*pred_N$km)+test.val.lower, pred_N$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
N_cline_centre_lower<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]

```

#do it for the Y cline

```

test.val.upper<-results$positions_Y
best_fit_LL<-func_likelihood_MELP_Y(c(results$positions_Y,results$stretch_factors_Y))
while(func_likelihood_MELP_Y(c(test.val.upper,results$stretch_factors_Y)) > best_fit_LL - 2 )
{test.val.upper<-test.val.upper+0.0001
}
print(paste("centre upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_Y(c(test.val.upper,results$stretch_factors_Y)),sep=""))
test.val.lower<-results$positions_Y
while(func_likelihood_MELP_Y(c(test.val.lower,results$stretch_factors_Y)) > best_fit_LL - 2 )
{test.val.lower<-test.val.lower-0.0001
}
print(paste("centre lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_MELP_Y(c(test.val.lower,results$stretch_factors_Y)),sep=""))

```

```

interp.dat<-aspline((results$stretch_factors_Y*pred_Y$km)+test.val.upper, pred_Y$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
Y_cline_centre_upper<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
interp.dat<-aspline((results$stretch_factors_Y*pred_Y$km)+test.val.lower, pred_Y$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
Y_cline_centre_lower<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]

```

```

results_for_best_fitting_cline<-results[1,]
results_for_best_fitting_cline<-
cbind(results_for_best_fitting_cline,B_cline_slope_lower_interval,B_cline_slope_upper_interval,B_cline
_width_lower_interval,B_cline_width_upper_interval)
results_for_best_fitting_cline<-
cbind(results_for_best_fitting_cline,D_cline_slope_lower_interval,D_cline_slope_upper_interval,D_cline
_width_lower_interval,D_cline_width_upper_interval)
results_for_best_fitting_cline<-
cbind(results_for_best_fitting_cline,N_cline_slope_lower_interval,N_cline_slope_upper_interval,N_cline
_width_lower_interval,N_cline_width_upper_interval)
results_for_best_fitting_cline<-
cbind(results_for_best_fitting_cline,Y_cline_slope_lower_interval,Y_cline_slope_upper_interval,Y_cline
_width_lower_interval,Y_cline_width_upper_interval)
results_for_best_fitting_cline<-
cbind(results_for_best_fitting_cline,B_cline_centre_upper,B_cline_centre_lower,D_cline_centre_upper,
D_cline_centre_lower,N_cline_centre_upper,N_cline_centre_lower,Y_cline_centre_upper,Y_cline_centre
_lower)

```

```

#####
###FIND SUPPORT LIMITS FOR WIDTH RATIO BETWEEN ERATO CLINES###
#####

```

```

library(akima)

```

```

###likelihood functions to calculate the likelihoods for each cline separately;

```

```

func_likelihood_ERATO_LIKE_D <- function(x) {
  freqs_D <- aspline((x[2]*pred_D$km)+x[1], pred_D$freq, xout=obs$km, degree=10)
  p_D <- freqs_D$y
  p_D[p_D==0] <- 1e-5
  p_D[p_D==1] <- 1-1e-5
  likelihood<-sum((log(p_D^y_D))+ log((1-p_D)^(k_D-y_D)))
  return(likelihood)
}

```

```

func_likelihood_ERATO_LIKE_C <- function(x) {
  freqs_C <- aspline((x[2]*pred_C$km)+x[1], pred_C$freq, xout=obs$km, degree=10)
  p_C <- ((freqs_C$y*freqs_C$y)+(2*(freqs_C$y*(1-freqs_C$y))))
  p_C[p_C==0] <- 1e-55
  p_C[p_C==1] <- 1-1e-5
  likelihood<-sum((log(p_C^y_C))+ log((1-p_C)^(k_C-y_C)))
  return(likelihood)
}

```

```

func_likelihood_ERATO_LIKE_S <- function(x) {
  freqs_S <- aspline((x[2]*pred_S$km)+x[1], pred_S$freq, xout=obs$km, degree=10)
  p_S <- ((freqs_S$y*freqs_S$y)+(2*(freqs_S$y*(1-freqs_S$y))))

```

```

p_S[p_S==0] <- 1e-55
p_S[p_S==1] <- 1-1e-5
likelihood<-sum((log(p_S^y_S))+ log((1-p_S)^(k_S-y_S)))
return(likelihood)
}

```

function to calculate the likelihood of the three clines fitted simultaneously:

#THIS VERSION ALLOWS THE POSITIONS OF THE CLINES TO MOVE INDEPENDENTLY and the stretch factor to change..

requires vector x with SIX values(THREE constants that is added to the km pos of the predicted values to make them match up with the observed data, AND THREE stretch factors),

and then for each locus need to have the following data loaded up:

pred_km (vector; km positions (demes) of simulated data), pred_freqs (vector; predicted frequency of dominant PHENOTYPE for demes), obs_km (vector; observed km positions),
k (vector; n trials in observed data), y (vector; n successes in observed data)

#do it for the D locus, ussing number of D alleles and expected proportion of D alleles

#do it for the D locus, ussing number of D alleles and expected proportion of D alleles

#Find max slope of simulated cline function

```

max_slope <- function(x) {
slopes<-c()
for (i in 1:(length(x)-1)){
slopes<-append(slopes,x[i+1]-x[i])}
return(max(slopes))
}

```

#observed data

```

obsJM <- read.csv("erato1990.csv", header=T, sep=",")
obsNR <- read.csv("erato2011.csv", header=T, sep=",")
combined <- read.csv("erato_combined.csv", header=T, sep=",")
predicted_all_data <- read.csv("erato_sim_results.csv", header=T, sep=",")
predicted_all_data <- subset(predicted_all_data, selectionD==0.6 & selectionC == 0.6 & selectionS == 0.6)

```

#Approximate point of the middle of the hybrid zone on the transect (this is used as a starting point in the optimization algorithm) (123.57 is Mallet 1990 estimated centre of D cline).

```
approx_mid_point<-123.57
```

#change this to the date set you want to analyse

```

obs<-combined
obs <- obs[order(obs$km),]

```

#Subset the D locus

```
predicted_D<-subset(predicted_all_data, predicted_all_data$locus == "D")
```

```

rownames(predicted_D) <- seq(length=nrow(predicted_D))
#this predicted for loops is just the allele frequencies without the row details, so you may have to
change the "5" depending on the input file
predicted_for_loops_D<-predicted_D[7:ncol(predicted_D)]
demes<-c(1:ncol(predicted_for_loops_D))

k_D <- obs$total_alleles ## total number of gene copies sampled
y_D <- obs$n_D_alleles ## number of successes

#Subset the C locus
predicted_C<-subset(predicted_all_data, predicted_all_data$locus == "C")
rownames(predicted_C) <- seq(length=nrow(predicted_C))

#this predicted for loops is just the allele frequencies without the row details, so you may have to
change the "5" depending on the input file
predicted_for_loops_C<-predicted_C[7:ncol(predicted_C)]
demes<-c(1:ncol(predicted_for_loops_C))

k_C <- obs$total_alleles/2 ## total number individuals
y_C <- k_C-(obs$n_cc_individuals) ## total number of C- individuals

#Subset the S locus
predicted_S<-subset(predicted_all_data, predicted_all_data$locus == "S")
rownames(predicted_S) <- seq(length=nrow(predicted_S))

#this predicted for loops is just the allele frequencies without the row details, so you may have to
change the "5" depending on the input file
predicted_for_loops_S<-predicted_S[7:ncol(predicted_S)]
demes<-c(1:ncol(predicted_for_loops_S))

k_S <- obs$total_alleles/2 ## total number individuals
y_S <- k_S-(obs$n_ss_individuals) ## total number of C- individuals

#Now loop through all the parameter combinations, calculate the likelihood and return
likelihoods<-c()
likelihoods_D<-c()
likelihoods_C<-c()
likelihoods_S<-c()
positions_D<-c()
positions_C<-c()
positions_S<-c()
stretch_factors_D<-c()
stretch_factors_C<-c()
stretch_factors_S<-c()
D_centres<-c()
C_centres<-c()
S_centres<-c()
D_cline_widths<-c()

```

```

C_cline_widths<-c()
S_cline_widths<-c()
D_max_slope<-c()
C_max_slope<-c()
S_max_slope<-c()

for (i in 1:nrow(predicted_D)){

pred_D <- data.frame(demes,t(predicted_for_loops_D[i,]))
pred_C <- data.frame(demes,t(predicted_for_loops_C[i,]))
pred_S <- data.frame(demes,t(predicted_for_loops_S[i,]))

names(pred_D) <- c("km_D","freq_D")
names(pred_C) <- c("km_C","freq_C")
names(pred_S) <- c("km_S","freq_S")

#optimisation algorithm,
# starting parameters = 1) align the approx middle of the hybrid zone with the middle of the simulated
# cline, and 2) using 1 as the stretch factor (i.e. no stretch: less than 1 squeezes the cline together, greater
# than 1 pulls it apart).
approx_centre_sim_D<-which.min(abs(pred_D$freq_D-0.5))
starting_param_1<-approx_mid_point-approx_centre_sim_D #this find the difference between the
observed data and simulated data using the D locus, and uses this as a starting point for each of the loci
likelihood_D <- optim(par = c(starting_param_1,1), func_likelihood_ERATO_LIKE_D, NULL,
control=list(fnscale=-1))
likelihood_C <- optim(par = c(starting_param_1,1), func_likelihood_ERATO_LIKE_C, NULL,
control=list(fnscale=-1))
likelihood_S <- optim(par = c(starting_param_1,1), func_likelihood_ERATO_LIKE_S, NULL,
control=list(fnscale=-1))
likelihood<-sum(likelihood_D$value,likelihood_C$value,likelihood_S$value)
#find middle of clines and widths
#this is a clunky bit to find the centre of the cline; aspline doesn't seem to be able to find the x value for
a given y value, only the other way round, so had to do it like this
interp.dat<-aspline((likelihood_D$par[2]*pred_D$km)+likelihood_D$par[1], pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
D_cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
D_cline_max_slope<-max_slope(interp.dat$y)
D_cline_width<-(1/D_cline_max_slope)/100 #divide by 100 to bring into km scale (10m was specified in
the interp.dat fine scale)
interp.dat<-aspline((likelihood_C$par[2]*pred_C$km)+likelihood_C$par[1], pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
C_cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
C_cline_max_slope<-max_slope(interp.dat$y)
C_cline_width<-(1/C_cline_max_slope)/100 #divide by 100 to bring into km scale (10m was specified in
the interp.dat fine scale)
interp.dat<-aspline((likelihood_S$par[2]*pred_S$km)+likelihood_S$par[1], pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
S_cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]

```

```
S_cline_max_slope<-max_slope(interp.dat$y)
S_cline_width<-(1/S_cline_max_slope)/100 #divide by 100 to bring into km scale (10m was specified in
the interp.dat fine scale)
```

```
#append results to results vectors..
```

```
likelihoods<-append(likelihoods,likelihood)
likelihoods_D<-append(likelihoods_D,likelihood_D$value)
likelihoods_C<-append(likelihoods_C,likelihood_C$value)
likelihoods_S<-append(likelihoods_S,likelihood_S$value)
positions_D<-append(positions_D,likelihood_D$par[1])
positions_C<-append(positions_C,likelihood_C$par[1])
positions_S<-append(positions_S,likelihood_S$par[1])
stretch_factors_D<-append(stretch_factors_D,likelihood_D$par[2])
stretch_factors_C<-append(stretch_factors_C,likelihood_C$par[2])
stretch_factors_S<-append(stretch_factors_S,likelihood_S$par[2])
D_centres<-append(D_centres,D_cline_centre)
C_centres<-append(C_centres,C_cline_centre)
S_centres<-append(S_centres,S_cline_centre)
D_cline_widths<-append(D_cline_widths,D_cline_width)
C_cline_widths<-append(C_cline_widths,C_cline_width)
S_cline_widths<-append(S_cline_widths,S_cline_width)
D_max_slope<-append(D_max_slope,D_cline_max_slope*100)#times 100 to bring on to km scale
C_max_slope<-append(C_max_slope,C_cline_max_slope*100)#times 100 to bring on to km scale
S_max_slope<-append(S_max_slope,S_cline_max_slope*100)#times 100 to bring on to km scale

}
```

```
####
```

```
#NaNs may be produced - this is OK - its just when the Optimization tests weird starting parameters I
think.
```

```
#you may need to change 1:4 to get the right columns to paste into the results....
```

```
results<-
cbind(predicted_D[1:5],likelihoods,positions_D,positions_C,positions_S,stretch_factors_D,stretch_factor
s_C,stretch_factors_S,D_centres,C_centres,S_centres,D_cline_widths,C_cline_widths,S_cline_widths,D_
max_slope,C_max_slope,S_max_slope,likelihoods_D,likelihoods_C,likelihoods_S)
```

```
results<-results[order(-results$likelihoods),]
```

```
results[1:20,]
```

```
#Now extract the best fitting simulated cline for plotting etc (need to centre on zero again).
```

```
pred_D <- data.frame(demes,t(predicted_for_loops_D[row.names(results)[1,]]))
pred_C <- data.frame(demes,t(predicted_for_loops_C[row.names(results)[1,]]))
pred_S <- data.frame(demes,t(predicted_for_loops_S[row.names(results)[1,]]))
```

```
names(pred_D) <- c("km_D","freq_D")
```

```
names(pred_C) <- c("km_C","freq_C")
names(pred_S) <- c("km_S","freq_S")
```

```
# THIS IS FOR FINDING THE WIDTH RATIO SUPPORT LIMITS
```

```
# basically I widen one cline while simulataneously narrowing the other, until the summed likelihoods of
the two clines drops by -2/
#the code is quite long because each time you play with the stretch factor that moves the position of the
centre of the cline, so you keep
#having to realign the cline and make the centre in the same place it was before.
```

```
### DO IT FOR D VERSUS C
```

```
#find upper bound
```

```
##starting ratio of widths
```

```
D.vs.C_ratio<-results$D_cline_widths/results$C_cline_widths
```

```
#for the D cline
```

```
mid_point_for_pivotD<-results$D_centres
```

```
positionsD<-results$positions_D
```

```
test.val.D<-results$stretch_factors_D
```

```
#for the C cline
```

```
mid_point_for_pivotC<-results$C_centres
```

```
positionsC<-results$positions_C
```

```
test.val.C<-results$stretch_factors_C
```

```
starting_likelihood<-func_likelihood_ERATO_LIKE_D(c(positionsD,test.val.D)) +
func_likelihood_ERATO_LIKE_C(c(positionsC,test.val.C))
```

```
while(func_likelihood_ERATO_LIKE_D(c(positionsD,test.val.D)) +
func_likelihood_ERATO_LIKE_C(c(positionsC,test.val.C)) > starting_likelihood - 2 )
```

```
{test.val.D<-test.val.D+0.0001
```

```
test.val.C<-test.val.C-0.0001
```

```
interp.dat.D<-aspline((test.val.D*pred_D$km)+positionsD, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
```

```
interp.dat.C<-aspline((test.val.C*pred_C$km)+positionsC, pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
```

```
cline_centre.D<-interp.dat.D$x[which.min(abs(interp.dat.D$y-0.5))]
```

```
cline_centre.C<-interp.dat.C$x[which.min(abs(interp.dat.C$y-0.5))]
```

```
positionsD<-positionsD-(cline_centre.D-mid_point_for_pivotD)
```

```
positionsC<-positionsC-(cline_centre.C-mid_point_for_pivotC)
```

```
}
```

```
interp.dat.D<-aspline((test.val.D*pred_D$km)+positionsD, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
```

```
D_cline_slope<-max_slope(interp.dat.D$y)*100 #times by 100 to bring onto km scale
```

```
D_cline_width<-(1/D_cline_slope)
```

```
interp.dat.C<-aspline((test.val.C*pred_C$km)+positionsC, pred_C$freq, xout=
```

```
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
```

```
C_cline_slope<-max_slope(interp.dat.C$y)*100 #times by 100 to bring onto km scale
```

```
C_cline_width<-(1/C_cline_slope)
```

```
D.vs.C_ratio.upper<-D_cline_width/C_cline_width
```

```
#Find lower bound
```

```
#for the D cline
```

```
mid_point_for_pivotD<-results$D_centres
```

```
positionsD<-results$positions_D
```

```
test.val.D<-results$stretch_factors_D
```

```
#for the C cline
```

```
mid_point_for_pivotC<-results$C_centres
```

```
positionsC<-results$positions_C
```

```
test.val.C<-results$stretch_factors_C
```

```
starting_likelihood<-func_likelihood_ERATO_LIKE_D(c(positionsD,test.val.D)) +
```

```
func_likelihood_ERATO_LIKE_C(c(positionsC,test.val.C))
```

```
while(func_likelihood_ERATO_LIKE_D(c(positionsD,test.val.D)) +
```

```
func_likelihood_ERATO_LIKE_C(c(positionsC,test.val.C)) > starting_likelihood - 2 )
```

```
{test.val.D<-test.val.D-0.0001
```

```
test.val.C<-test.val.C+0.0001
```

```
interp.dat.D<-aspline((test.val.D*pred_D$km)+positionsD, pred_D$freq, xout=
```

```
seq(min(obs$km),max(obs$km),0.01), degree=10)
```

```
interp.dat.C<-aspline((test.val.C*pred_C$km)+positionsC, pred_C$freq, xout=
```

```
seq(min(obs$km),max(obs$km),0.01), degree=10)
```

```
cline_centre.D<-interp.dat.D$x[which.min(abs(interp.dat.D$y-0.5))]
```

```
cline_centre.C<-interp.dat.C$x[which.min(abs(interp.dat.C$y-0.5))]
```

```
positionsD<-positionsD-(cline_centre.D-mid_point_for_pivotD)
```

```
positionsC<-positionsC-(cline_centre.C-mid_point_for_pivotC)
```

```
}
```

```
interp.dat.D<-aspline((test.val.D*pred_D$km)+positionsD, pred_D$freq, xout=
```

```
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
```

```
D_cline_slope<-max_slope(interp.dat.D$y)*100 #times by 100 to bring onto km scale
```

```
D_cline_width<-(1/D_cline_slope)
```

```
interp.dat.C<-aspline((test.val.C*pred_C$km)+positionsC, pred_C$freq, xout=
```

```
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
```

```
C_cline_slope<-max_slope(interp.dat.C$y)*100 #times by 100 to bring onto km scale
```

```
C_cline_width<-(1/C_cline_slope)
```

```
D.vs.C_ratio.lower<-D_cline_width/C_cline_width
```

```
### DO IT FOR D VERSUS S
```

```
#FIND UPPER BOUND
```

```
##starting ratio of widths
```

```
D.vs.S_ratio<-results$D_cline_widths/results$S_cline_widths
```

```
mid_point_for_pivotD<-results$D_centres
```

```
positionsD<-results$positions_D
```

```
test.val.D<-results$stretch_factors_D
```

```
mid_point_for_pivotS<-results$S_centres
```

```

positionsS<-results$positions_S
test.val.S<-results$stretch_factors_S

starting_likelihood<-func_likelihood_ERATO_LIKE_D(c(positionsD,test.val.D)) +
func_likelihood_ERATO_LIKE_S(c(positionsS,test.val.S))

while(func_likelihood_ERATO_LIKE_D(c(positionsD,test.val.D)) +
func_likelihood_ERATO_LIKE_S(c(positionsS,test.val.S)) > starting_likelihood - 2 )
{test.val.D<-test.val.D+0.0001
test.val.S<-test.val.S-0.0001
interp.dat.D<-aspline((test.val.D*pred_D$km)+positionsD, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
interp.dat.S<-aspline((test.val.S*pred_S$km)+positionsS, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre.D<-interp.dat.D$x[which.min(abs(interp.dat.D$y-0.5))]
cline_centre.S<-interp.dat.S$x[which.min(abs(interp.dat.S$y-0.5))]
positionsD<-positionsD-(cline_centre.D-mid_point_for_pivotD)
positionsS<-positionsS-(cline_centre.S-mid_point_for_pivotS)
}
interp.dat.D<-aspline((test.val.D*pred_D$km)+positionsD, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
D_cline_slope<-max_slope(interp.dat.D$y)*100 #times by 100 to bring onto km scale
D_cline_width<-(1/D_cline_slope)
interp.dat.S<-aspline((test.val.S*pred_S$km)+positionsS, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
S_cline_slope<-max_slope(interp.dat.S$y)*100 #times by 100 to bring onto km scale
S_cline_width<-(1/S_cline_slope)

D.vs.S_ratio.upper<-D_cline_width/S_cline_width

#FIND lower BOUND
##starting ratio of widths
D.vs.S_ratio<-results$D_cline_widths/results$S_cline_widths

mid_point_for_pivotD<-results$D_centres
positionsD<-results$positions_D
test.val.D<-results$stretch_factors_D
mid_point_for_pivotS<-results$S_centres
positionsS<-results$positions_S
test.val.S<-results$stretch_factors_S

starting_likelihood<-func_likelihood_ERATO_LIKE_D(c(positionsD,test.val.D)) +
func_likelihood_ERATO_LIKE_S(c(positionsS,test.val.S))

while(func_likelihood_ERATO_LIKE_D(c(positionsD,test.val.D)) +
func_likelihood_ERATO_LIKE_S(c(positionsS,test.val.S)) > starting_likelihood - 2 )
{test.val.D<-test.val.D-0.0001
test.val.S<-test.val.S+0.0001

```

```

interp.dat.D<-aspline((test.val.D*pred_D$km)+positionsD, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
interp.dat.S<-aspline((test.val.S*pred_S$km)+positionsS, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre.D<-interp.dat.D$x[which.min(abs(interp.dat.D$y-0.5))]
cline_centre.S<-interp.dat.S$x[which.min(abs(interp.dat.S$y-0.5))]
positionsD<-positionsD-(cline_centre.D-mid_point_for_pivotD)
positionsS<-positionsS-(cline_centre.S-mid_point_for_pivotS)
}
interp.dat.D<-aspline((test.val.D*pred_D$km)+positionsD, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
D_cline_slope<-max_slope(interp.dat.D$y)*100 #times by 100 to bring onto km scale
D_cline_width<-(1/D_cline_slope)
interp.dat.S<-aspline((test.val.S*pred_S$km)+positionsS, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
S_cline_slope<-max_slope(interp.dat.S$y)*100 #times by 100 to bring onto km scale
S_cline_width<-(1/S_cline_slope)

D.vs.S_ratio.lower<-D_cline_width/S_cline_width

while(func_likelihood_ERATO_LIKE_D(c(positions,test.val.upper)) > results[1,22] - 2 )
{test.val.upper<-test.val.upper+0.0001
interp.dat<-aspline((test.val.upper*pred_D$km)+positions, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_D(c(positions,test.val.upper)),sep=""))
interp.dat<-aspline((test.val.upper*pred_D$km)+positions, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
D_cline_slope_lower_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
D_cline_width_upper_interval<-(1/D_cline_slope_lower_interval)

mid_point_for_pivot<-results[1,13]
positions<-results[1,7]
test.val.lower<-results[1,10]
while(func_likelihood_ERATO_LIKE_D(c(positions,test.val.lower)) > results[1,22] - 2 )
{test.val.lower<-test.val.lower-0.0001
interp.dat<-aspline((test.val.lower*pred_D$km)+positions, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_D(c(positions,test.val.lower)),sep=""))
interp.dat<-aspline((test.val.lower*pred_D$km)+positions, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)

```

```
D_cline_slope_upper_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
D_cline_width_lower_interval<-(1/D_cline_slope_upper_interval)
```

```
#do it for the C cline
```

```
mid_point_for_pivot<-results[1,14]
positions<-results[1,8]
test.val.upper<-results[1,11]
while(func_likelihood_ERATO_LIKE_C(c(positions,test.val.upper)) > results[1,23] - 2 )
{test.val.upper<-test.val.upper+0.0001
interp.dat<-aspline((test.val.upper*pred_C$km)+positions, pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_C(c(positions,test.val.upper)),sep=""))
interp.dat<-aspline((test.val.upper*pred_C$km)+positions, pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
C_cline_slope_lower_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
C_cline_width_upper_interval<-(1/C_cline_slope_lower_interval)
```

```
mid_point_for_pivot<-results[1,14]
positions<-results[1,8]
test.val.lower<-results[1,11]
while(func_likelihood_ERATO_LIKE_C(c(positions,test.val.lower)) > results[1,23] - 2 )
{test.val.lower<-test.val.lower-0.0001
interp.dat<-aspline((test.val.lower*pred_C$km)+positions, pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_C(c(positions,test.val.lower)),sep=""))
interp.dat<-aspline((test.val.lower*pred_C$km)+positions, pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
C_cline_slope_upper_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
C_cline_width_lower_interval<-(1/C_cline_slope_upper_interval)
```

```
#do it for the S cline
```

```
mid_point_for_pivot<-results[1,15]
positions<-results[1,9]
test.val.upper<-results[1,12]
while(func_likelihood_ERATO_LIKE_S(c(positions,test.val.upper)) > results[1,24] - 2 )
{test.val.upper<-test.val.upper+0.0001
interp.dat<-aspline((test.val.upper*pred_S$km)+positions, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
```

```

}
print(paste("stretch upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_S(c(positions,test.val.upper)),sep=""))
interp.dat<-aspline((test.val.upper*pred_S$km)+positions, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
S_cline_slope_lower_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
S_cline_width_upper_interval<-(1/S_cline_slope_lower_interval)

mid_point_for_pivot<-results[1,15]
positions<-results[1,9]
test.val.lower<-results[1,12]
while(func_likelihood_ERATO_LIKE_S(c(positions,test.val.lower)) > results[1,24] - 2 )
{test.val.lower<-test.val.lower-0.0001
interp.dat<-aspline((test.val.lower*pred_S$km)+positions, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
positions<-positions-(cline_centre-mid_point_for_pivot)
}
print(paste("stretch lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_S(c(positions,test.val.lower)),sep=""))
interp.dat<-aspline((test.val.lower*pred_S$km)+positions, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
S_cline_slope_upper_interval<-max_slope(interp.dat$y)*100 #times by 100 to bring onto km scale
S_cline_width_lower_interval<-(1/S_cline_slope_upper_interval)

# THIS IS FOR FINDING CENTRE SUPPORT LIMITS
#do it for the D cline
test.val.upper<-results[1,7]
while(func_likelihood_ERATO_LIKE_D(c(test.val.upper,results[1,10])) > results[1,22] - 2 )
{test.val.upper<-test.val.upper+0.0001
}
print(paste("centre upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_D(c(test.val.upper,results[1,10])),sep=""))
test.val.lower<-results[1,7]
while(func_likelihood_ERATO_LIKE_D(c(test.val.lower,results[1,10])) > results[1,22] - 2 )
{test.val.lower<-test.val.lower-0.0001
}
print(paste("centre lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_D(c(test.val.lower,results[1,10])),sep=""))

interp.dat<-aspline((results[1,10]*pred_D$km)+test.val.upper, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
D_cline_centre_upper<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
interp.dat<-aspline((results[1,10]*pred_D$km)+test.val.lower, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
D_cline_centre_lower<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]

```

```

#do it for the C cline
test.val.upper<-results[1,8]
while(func_likelihood_ERATO_LIKE_C(c(test.val.upper,results[1,11])) > results[1,23] - 2 )
{test.val.upper<-test.val.upper+0.0001
}
print(paste("centre upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_C(c(test.val.upper,results[1,11])),sep=""))
test.val.lower<-results[1,8]
while(func_likelihood_ERATO_LIKE_C(c(test.val.lower,results[1,11])) > results[1,23] - 2 )
{test.val.lower<-test.val.lower-0.0001
}
print(paste("centre lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_C(c(test.val.lower,results[1,11])),sep=""))

interp.dat<-aspline((results[1,11]*pred_C$km)+test.val.upper, pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
C_cline_centre_upper<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
interp.dat<-aspline((results[1,11]*pred_C$km)+test.val.lower, pred_C$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
C_cline_centre_lower<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]

#do it for the S cline
test.val.upper<-results[1,9]
while(func_likelihood_ERATO_LIKE_S(c(test.val.upper,results[1,12])) > results[1,24] - 2 )
{test.val.upper<-test.val.upper+0.0001
}
print(paste("centre upper bound =",test.val.upper,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_S(c(test.val.upper,results[1,12])),sep=""))
test.val.lower<-results[1,9]
while(func_likelihood_ERATO_LIKE_S(c(test.val.lower,results[1,12])) > results[1,24] - 2 )
{test.val.lower<-test.val.lower-0.0001
}
print(paste("centre lower bound =",test.val.lower,sep=""))
print(paste("likelihood = ",func_likelihood_ERATO_LIKE_S(c(test.val.lower,results[1,12])),sep=""))

interp.dat<-aspline((results[1,12]*pred_S$km)+test.val.upper, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
S_cline_centre_upper<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
interp.dat<-aspline((results[1,12]*pred_S$km)+test.val.lower, pred_S$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
S_cline_centre_lower<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]

results_for_best_fitting_cline<-results[1,]
results_for_best_fitting_cline<-
cbind(results_for_best_fitting_cline,D_cline_slope_lower_interval,D_cline_slope_upper_interval,D_clin
e_width_lower_interval,D_cline_width_upper_interval)

```

```

results_for_best_fitting_cline<-
cbind(results_for_best_fitting_cline,C_cline_slope_lower_interval,C_cline_slope_upper_interval,C_cline
_width_lower_interval,C_cline_width_upper_interval)
results_for_best_fitting_cline<-
cbind(results_for_best_fitting_cline,S_cline_slope_lower_interval,S_cline_slope_upper_interval,S_cline
_width_lower_interval,S_cline_width_upper_interval)
results_for_best_fitting_cline<-
cbind(results_for_best_fitting_cline,D_cline_centre_upper,D_cline_centre_lower,C_cline_centre_upper,
C_cline_centre_lower,S_cline_centre_upper,S_cline_centre_lower)

```

```

results_for_best_fitting_cline<-results_for_best_fitting_cline[-(c(1,7:12))]

```

```

#####
###FIND SUPPORT LIMITS FOR WIDTH RATIO BETWEEN MELPOMENE CLINES###
#####

```

```

library(akima)

```

```

func_likelihood_MELP_B <- function(x) {
  freqs_B <- aspline((x[2]*pred_B$km)+x[1], pred_B$freq, xout=obs$km, degree=10)
  p_B <- ((freqs_B$y*freqs_B$y)+(2*(freqs_B$y*(1-freqs_B$y))))
  p_B[p_B==0] <- 1e-55
  p_B[p_B==1] <- 1-1e-5
  #had to divide likelihood calculation up as R stored very small probabilities as zero (before was coded
as likelihood_B<-sum((log(p_B^y_B))+ log((1-p_B)^(k_B-y_B))) - see erato script)
  part_a<-p_B^y_B
  part_b<-(1-p_B)^(k_B-y_B)
  part_a[part_a==0] <- 1e-55
  part_b[part_b==0] <- 1e-55
  likelihood<-sum(log(part_a)+log(part_b))
  return(likelihood)
}

```

```

func_likelihood_MELP_D <- function(x) {
  freqs_D <- aspline((x[2]*pred_D$km)+x[1], pred_D$freq, xout=obs$km, degree=10)
  p_D <- ((freqs_D$y*freqs_D$y)+(2*(freqs_D$y*(1-freqs_D$y))))
  p_D[p_D==0] <- 1e-55
  p_D[p_D==1] <- 1-1e-5
  part_a<-p_D^y_D
  part_b<-(1-p_D)^(k_D-y_D)
  part_a[part_a==0] <- 1e-55
  part_b[part_b==0] <- 1e-55
  likelihood<-sum(log(part_a)+log(part_b))
  return(likelihood)
}

```

```

func_likelihood_MELP_N <- function(x) {

```

```

      freqs_N <- aspline((x[2]*pred_N$km)+x[1], pred_N$freq, xout=obs$km, degree=10)
p_N <- ((freqs_N$y*freqs_N$y)+(2*(freqs_N$y*(1-freqs_N$y))))
p_N[p_N==0] <- 1e-55
p_N[p_N==1] <- 1-1e-5
part_a<-p_N^y_N
part_b<-(1-p_N)^(k_N-y_N)
part_a[part_a==0] <- 1e-55
part_b[part_b==0] <- 1e-55
likelihood<-sum(log(part_a)+log(part_b))
return(likelihood)
}

func_likelihood_MELP_Y <- function(x) {
  freqs_Y <- aspline((x[2]*pred_Y$km)+x[1], pred_Y$freq, xout=obs$km, degree=10)
p_Y <- ((freqs_Y$y*freqs_Y$y)+(2*(freqs_Y$y*(1-freqs_Y$y))))
p_Y[p_Y==0] <- 1e-55
p_Y[p_Y==1] <- 1-1e-5
part_a<-p_Y^y_Y
part_b<-(1-p_Y)^(k_Y-y_Y)
part_a[part_a==0] <- 1e-55
part_b[part_b==0] <- 1e-55
likelihood<-sum(log(part_a)+log(part_b))
return(likelihood)
}

#Find max slope of simulated cline function
max_slope <- function(x) {
slopes<-c()
for (i in 1:(length(x)-1)){
slopes<-append(slopes,x[i+1]-x[i])}
return(max(slopes))
}

#observed data
obsJM <- read.csv("melpomene1990.csv", header=T, sep=",")
obsNR <- read.csv("melpomene2011.csv", header=T, sep=",")
combined <- read.csv("melpomene_combined.csv", header=T, sep=",")
predicted_all_data <- read.csv("melp_sim_results.csv", header=T, sep=",")
predicted_all_data<-
subset(predicted_all_data,selectionB==0.6&selectionD==0.6&selectionN==0.6&selectionY==0.6&recom
b_r_BD==0.135)

#Approximate point of the middle of the hybrid zone on the transect (this is used as a starting point in
the optimization algorithm) (123.57 is Mallet 1990 estimated centre of D cline).
approx_mid_point<-123.57

#change this to the date set you want to analyse
obs<-combined

```

```

obs <- obs[order(obs$km),]

#Subset the B locus
predicted_B<-subset(predicted_all_data, predicted_all_data$locus == "B")
rownames(predicted_B) <- seq(length=nrow(predicted_B))

#this predicted for loops is just the allele frequencies without the row details, so you may have to
change the "5" depending on the input file
predicted_for_loops_B<-predicted_B[10:ncol(predicted_B)]
demes<-c(1:ncol(predicted_for_loops_B))

k_B <- obs$total_alleles/2 ## total number individuals
y_B <- k_B-(obs$n_bb_individuals) ## total number of B- individuals

#Subset the D locus
predicted_D<-subset(predicted_all_data, predicted_all_data$locus == "D")
rownames(predicted_D) <- seq(length=nrow(predicted_D))

#this predicted for loops is just the allele frequencies without the row details, so you may have to
change the "5" depending on the input file
predicted_for_loops_D<-predicted_D[10:ncol(predicted_D)]
demes<-c(1:ncol(predicted_for_loops_D))

k_D <- obs$total_alleles/2 ## total number individuals
y_D <- k_D-(obs$n_dd_individuals) ## total number of D- individuals

#Subset the N locus
predicted_N<-subset(predicted_all_data, predicted_all_data$locus == "N")
rownames(predicted_N) <- seq(length=nrow(predicted_N))

#this predicted for loops is just the allele frequencies without the row details, so you may have to
change the "5" depending on the input file
predicted_for_loops_N<-predicted_N[10:ncol(predicted_N)]
demes<-c(1:ncol(predicted_for_loops_N))

k_N <- obs$total_alleles/2 ## total number individuals
y_N <- k_N-(obs$n_nn_individuals) ## total number of N- individuals

#Subset the Y locus (predicted cline is the same as the N locus assuming very tight linkage)...
predicted_Y<-subset(predicted_all_data, predicted_all_data$locus == "Y")
rownames(predicted_Y) <- seq(length=nrow(predicted_Y))

#this predicted for loops is just the allele frequencies without the row details, so you may have to
change the "5" depending on the input file
predicted_for_loops_Y<-predicted_Y[10:ncol(predicted_Y)]
demes<-c(1:ncol(predicted_for_loops_Y))

k_Y <- obs$total_alleles/2 ## total number individuals

```

```
y_Y <- k_Y-(obs$n_yy_individuals) ## total number of Y- individuals
```

```
#Now loop through all the parameter combinations, calculate the likelihood and return
```

```
likelihoods<-c()
likelihoods_B<-c()
likelihoods_D<-c()
likelihoods_N<-c()
likelihoods_Y<-c()
positions_B<-c()
positions_D<-c()
positions_N<-c()
positions_Y<-c()
stretch_factors_B<-c()
stretch_factors_D<-c()
stretch_factors_N<-c()
stretch_factors_Y<-c()
B_centres<-c()
D_centres<-c()
N_centres<-c()
Y_centres<-c()
B_cline_widths<-c()
D_cline_widths<-c()
N_cline_widths<-c()
Y_cline_widths<-c()
B_max_slope<-c()
D_max_slope<-c()
N_max_slope<-c()
Y_max_slope<-c()
```

```
for (i in 1:nrow(predicted_B)){
```

```
  pred_B <- data.frame(demes,t(predicted_for_loops_B[i,]))
  pred_D <- data.frame(demes,t(predicted_for_loops_D[i,]))
  pred_N <- data.frame(demes,t(predicted_for_loops_N[i,]))
  pred_Y <- data.frame(demes,t(predicted_for_loops_Y[i,]))
```

```
  names(pred_B) <- c("km_B","freq_B")
  names(pred_D) <- c("km_D","freq_D")
  names(pred_N) <- c("km_N","freq_N")
  names(pred_Y) <- c("km_Y","freq_Y")
```

```
#optimisation algorithm,
```

```
# starting parameters = 1) align the approx middle of the hybrid zone with the middle of the simulated cline, and 2) using 1 as the stretch factor (i.e. no stretch: less than 1 squeezes the cline together, greater than 1 pulls it apart).
```

```
approx_centre_sim_D<-which.min(abs(pred_D$freq_D-0.5))
```

```
starting_param_1<-approx_mid_point-approx_centre_sim_D #this find the difference between the observed data and simulated data using the D locus, and uses this as a starting point
```

```

likelihood_B <- optim(par = c(starting_param_1,1), func_likelihood_MELP_B, NULL, control=list(fnscale=-1))
likelihood_D <- optim(par = c(starting_param_1,1), func_likelihood_MELP_D, NULL,
control=list(fnscale=-1))
likelihood_N <- optim(par = c(starting_param_1,1), func_likelihood_MELP_N, NULL,
control=list(fnscale=-1))
likelihood_Y <- optim(par = c(starting_param_1,1), func_likelihood_MELP_Y, NULL, control=list(fnscale=-1))
likelihood<-sum(likelihood_B$value,likelihood_D$value,likelihood_N$value,likelihood_Y$value)
#find middle of clines and widths
#this is a clunky bit to find the centre of the cline; the spline doesn't seem to be able to find the x value
for a given y value, only the other way round, so had to do it like this
interp.dat<-aspline((likelihood_B$par[2]*pred_B$km)+likelihood_B$par[1], pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
B_cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
B_cline_max_slope<-max_slope(1-interp.dat$y)
B_cline_width<-(1/B_cline_max_slope)/100 #divide by 100 to bring into km scale (10m was specified in
the interp.dat fine scale)

interp.dat<-aspline((likelihood_D$par[2]*pred_D$km)+likelihood_D$par[1], pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
D_cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
D_cline_max_slope<-max_slope(interp.dat$y)
D_cline_width<-(1/D_cline_max_slope)/100 #divide by 100 to bring into km scale (10m was specified in
the interp.dat fine scale)

interp.dat<-aspline((likelihood_N$par[2]*pred_N$km)+likelihood_N$par[1], pred_N$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
N_cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
N_cline_max_slope<-max_slope(interp.dat$y)
N_cline_width<-(1/N_cline_max_slope)/100 #divide by 100 to bring into km scale (10m was specified in
the interp.dat fine scale)

interp.dat<-aspline((likelihood_Y$par[2]*pred_Y$km)+likelihood_Y$par[1], pred_Y$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
Y_cline_centre<-interp.dat$x[which.min(abs(interp.dat$y-0.5))]
Y_cline_max_slope<-max_slope(interp.dat$y)
Y_cline_width<-(1/Y_cline_max_slope)/100 #divide by 100 to bring into km scale (10m was specified in
the interp.dat fine scale)

#append results to results vectors..
likelihoods<-append(likelihoods,likelihood)
likelihoods_B<-append(likelihoods_B,likelihood_B$value)
likelihoods_D<-append(likelihoods_D,likelihood_D$value)
likelihoods_N<-append(likelihoods_N,likelihood_N$value)
likelihoods_Y<-append(likelihoods_Y,likelihood_Y$value)
positions_B<-append(positions_B,likelihood_B$par[1])
positions_D<-append(positions_D,likelihood_D$par[1])

```

```

positions_N<-append(positions_N,likelihood_N$par[1])
positions_Y<-append(positions_Y,likelihood_Y$par[1])
stretch_factors_B<-append(stretch_factors_B,likelihood_B$par[2])
stretch_factors_D<-append(stretch_factors_D,likelihood_D$par[2])
stretch_factors_N<-append(stretch_factors_N,likelihood_N$par[2])
stretch_factors_Y<-append(stretch_factors_Y,likelihood_Y$par[2])
B_centres<-append(B_centres,B_cline_centre)
D_centres<-append(D_centres,D_cline_centre)
N_centres<-append(N_centres,N_cline_centre)
Y_centres<-append(Y_centres,Y_cline_centre)
B_cline_widths<-append(B_cline_widths,B_cline_width)
D_cline_widths<-append(D_cline_widths,D_cline_width)
N_cline_widths<-append(N_cline_widths,N_cline_width)
Y_cline_widths<-append(Y_cline_widths,Y_cline_width)
B_max_slope<-append(B_max_slope,B_cline_max_slope*100)#times 100 to bring on to km scale
D_max_slope<-append(D_max_slope,D_cline_max_slope*100)#times 100 to bring on to km scale
N_max_slope<-append(N_max_slope,N_cline_max_slope*100)#times 100 to bring on to km scale
Y_max_slope<-append(Y_max_slope,Y_cline_max_slope*100)#times 100 to bring on to km scale

}

results<-
cbind(predicted_D[1:9],likelihoods,likelihoods_B,likelihoods_D,likelihoods_N,likelihoods_Y,positions_B,
positions_D,positions_N,positions_Y,stretch_factors_B,stretch_factors_D,stretch_factors_N,stretch_factors_Y,
B_centres,D_centres,N_centres,Y_centres,B_cline_widths,D_cline_widths,N_cline_widths,Y_cline_widths,
B_max_slope,D_max_slope,N_max_slope,Y_max_slope)

results<-results[order(-results$likelihoods),]

#put in the row for the most likely selection values here...
pred_B <- data.frame(demes,t(predicted_for_loops_B[row.names(results)[1],]))
pred_D <- data.frame(demes,t(predicted_for_loops_D[row.names(results)[1],]))
pred_N <- data.frame(demes,t(predicted_for_loops_N[row.names(results)[1],]))
pred_Y <- data.frame(demes,t(predicted_for_loops_Y[row.names(results)[1],]))

names(pred_B) <- c("km_B","freq_B")
names(pred_D) <- c("km_D","freq_D")
names(pred_N) <- c("km_N","freq_N")
names(pred_Y) <- c("km_Y","freq_Y")

# THIS IS FOR FINDING THE WIDTH RATIO SUPPORT LIMITS
# basically I widen one cline while simulataneously narrowing the other, until the summed likelihoods of
the two clines drops by -2/
#the code is quite long because each time you play with the stretch factor that moves the position of the
centre of the cline, so you keep
#having to realign the cline and make the centre in the same place it was before.

### DO IT FOR D VERSUS B

```

```

#Find the upper bound
##starting ratio of widths
D.vs.B_ratio<-results$D_cline_widths/results$B_cline_widths

#for the D cline
mid_point_for_pivotD<-results$D_centres
positionsD<-results$positions_D
test.val.D<-results$stretch_factors_D
#for the B cline
mid_point_for_pivotB<-results$B_centres
positionsB<-results$positions_B
test.val.B<-results$stretch_factors_B

starting_likelihood<-func_likelihood_MELP_D(c(positionsD,test.val.D)) +
func_likelihood_MELP_B(c(positionsB,test.val.B))

while(func_likelihood_MELP_D(c(positionsD,test.val.D)) +
func_likelihood_MELP_B(c(positionsB,test.val.B)) > starting_likelihood - 2 )
{test.val.D<-test.val.D+0.0001
test.val.B<-test.val.B-0.0001
interp.dat.D<-aspline((test.val.D*pred_D$km)+positionsD, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
interp.dat.B<-aspline((test.val.B*pred_B$km)+positionsB, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre.D<-interp.dat.D$x[which.min(abs(interp.dat.D$y-0.5))]
cline_centre.B<-interp.dat.B$x[which.min(abs(interp.dat.B$y-0.5))]
positionsD<-positionsD-(cline_centre.D-mid_point_for_pivotD)
positionsB<-positionsB-(cline_centre.B-mid_point_for_pivotB)
}
interp.dat.D<-aspline((test.val.D*pred_D$km)+positionsD, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
D_cline_slope<-max_slope(interp.dat.D$y)*100 #times by 100 to bring onto km scale
D_cline_width<-(1/D_cline_slope)
interp.dat.B<-aspline((test.val.B*pred_B$km)+positionsB, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
B_cline_slope<-max_slope(1-interp.dat.B$y)*100 #times by 100 to bring onto km scale
B_cline_width<-(1/B_cline_slope)

D.vs.B_ratio.upper<-D_cline_width/B_cline_width

#Find the lower bound
##starting ratio of widths
D.vs.B_ratio<-results$D_cline_widths/results$B_cline_widths

#for the D cline
mid_point_for_pivotD<-results$D_centres
positionsD<-results$positions_D
test.val.D<-results$stretch_factors_D

```

```

#for the B cline
mid_point_for_pivotB<-results$B_centres
positionsB<-results$positions_B
test.val.B<-results$stretch_factors_B

starting_likelihood<-func_likelihood_MELP_D(c(positionsD,test.val.D)) +
func_likelihood_MELP_B(c(positionsB,test.val.B))

while(func_likelihood_MELP_D(c(positionsD,test.val.D)) +
func_likelihood_MELP_B(c(positionsB,test.val.B)) > starting_likelihood - 2 )
{test.val.D<-test.val.D-0.0001
test.val.B<-test.val.B+0.0001
interp.dat.D<-aspline((test.val.D*pred_D$km)+positionsD, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
interp.dat.B<-aspline((test.val.B*pred_B$km)+positionsB, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre.D<-interp.dat.D$x[which.min(abs(interp.dat.D$y-0.5))]
cline_centre.B<-interp.dat.B$x[which.min(abs(interp.dat.B$y-0.5))]
positionsD<-positionsD-(cline_centre.D-mid_point_for_pivotD)
positionsB<-positionsB-(cline_centre.B-mid_point_for_pivotB)
}
interp.dat.D<-aspline((test.val.D*pred_D$km)+positionsD, pred_D$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
D_cline_slope<-max_slope(interp.dat.D$y)*100 #times by 100 to bring onto km scale
D_cline_width<-(1/D_cline_slope)
interp.dat.B<-aspline((test.val.B*pred_B$km)+positionsB, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
B_cline_slope<-max_slope(1-interp.dat.B$y)*100 #times by 100 to bring onto km scale
B_cline_width<-(1/B_cline_slope)

D.vs.B_ratio.lower<-D_cline_width/B_cline_width

### DO IT FOR N VERSUS B
#Find the upper bound
##starting ratio of widths
N.vs.B_ratio<-results$N_cline_widths/results$B_cline_widths

#for the N cline
mid_point_for_pivotN<-results$N_centres
positionsN<-results$positions_N
test.val.N<-results$stretch_factors_N
#for the B cline
mid_point_for_pivotB<-results$B_centres
positionsB<-results$positions_B
test.val.B<-results$stretch_factors_B

starting_likelihood<-func_likelihood_MELP_N(c(positionsN,test.val.N)) +
func_likelihood_MELP_B(c(positionsB,test.val.B))

```

```

while(func_likelihood_MELP_N(c(positionsN,test.val.N)) +
func_likelihood_MELP_B(c(positionsB,test.val.B)) > starting_likelihood - 2 )
{test.val.N<-test.val.N+0.0001
test.val.B<-test.val.B-0.0001
interp.dat.N<-aspline((test.val.N*pred_N$km)+positionsN, pred_N$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
interp.dat.B<-aspline((test.val.B*pred_B$km)+positionsB, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre.N<-interp.dat.N$x[which.min(abs(interp.dat.N$y-0.5))]
cline_centre.B<-interp.dat.B$x[which.min(abs(interp.dat.B$y-0.5))]
positionsN<-positionsN-(cline_centre.N-mid_point_for_pivotN)
positionsB<-positionsB-(cline_centre.B-mid_point_for_pivotB)
}
interp.dat.N<-aspline((test.val.N*pred_N$km)+positionsN, pred_N$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
N_cline_slope<-max_slope(interp.dat.N$y)*100 #times by 100 to bring onto km scale
N_cline_width<-(1/N_cline_slope)
interp.dat.B<-aspline((test.val.B*pred_B$km)+positionsB, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
B_cline_slope<-max_slope(1-interp.dat.B$y)*100 #times by 100 to bring onto km scale
B_cline_width<-(1/B_cline_slope)

N.vs.B_ratio.upper<-N_cline_width/B_cline_width

#Find the lower bound
##starting ratio of widths
N.vs.B_ratio<-results$N_cline_widths/results$B_cline_widths

#for the N cline
mid_point_for_pivotN<-results$N_centres
positionsN<-results$positions_N
test.val.N<-results$stretch_factors_N
#for the B cline
mid_point_for_pivotB<-results$B_centres
positionsB<-results$positions_B
test.val.B<-results$stretch_factors_B

starting_likelihood<-func_likelihood_MELP_N(c(positionsN,test.val.N)) +
func_likelihood_MELP_B(c(positionsB,test.val.B))

while(func_likelihood_MELP_N(c(positionsN,test.val.N)) +
func_likelihood_MELP_B(c(positionsB,test.val.B)) > starting_likelihood - 2 )
{test.val.N<-test.val.N-0.0001
test.val.B<-test.val.B+0.0001
interp.dat.N<-aspline((test.val.N*pred_N$km)+positionsN, pred_N$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)

```

```

interp.dat.B<-aspline((test.val.B*pred_B$km)+positionsB, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre.N<-interp.dat.N$x[which.min(abs(interp.dat.N$y-0.5))]
cline_centre.B<-interp.dat.B$x[which.min(abs(interp.dat.B$y-0.5))]
positionsN<-positionsN-(cline_centre.N-mid_point_for_pivotN)
positionsB<-positionsB-(cline_centre.B-mid_point_for_pivotB)
}
interp.dat.N<-aspline((test.val.N*pred_N$km)+positionsN, pred_N$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
N_cline_slope<-max_slope(interp.dat.N$y)*100 #times by 100 to bring onto km scale
N_cline_width<-(1/N_cline_slope)
interp.dat.B<-aspline((test.val.B*pred_B$km)+positionsB, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
B_cline_slope<-max_slope(1-interp.dat.B$y)*100 #times by 100 to bring onto km scale
B_cline_width<-(1/B_cline_slope)

N.vs.B_ratio.lower<-N_cline_width/B_cline_width

#### DO IT FOR Y VERSUS B
#Find the upper bound
##starting ratio of widths
Y.vs.B_ratio<-results$Y_cline_widths/results$B_cline_widths

#for the Y cline
mid_point_for_pivotY<-results$Y_centres
positionsY<-results$positions_Y
test.val.Y<-results$stretch_factors_Y
#for the B cline
mid_point_for_pivotB<-results$B_centres
positionsB<-results$positions_B
test.val.B<-results$stretch_factors_B

starting_likelihood<-func_likelihood_MELP_Y(c(positionsY,test.val.Y)) +
func_likelihood_MELP_B(c(positionsB,test.val.B))

while(func_likelihood_MELP_Y(c(positionsY,test.val.Y)) +
func_likelihood_MELP_B(c(positionsB,test.val.B)) > starting_likelihood - 2 )
{test.val.Y<-test.val.Y+0.0001
test.val.B<-test.val.B-0.0001
interp.dat.Y<-aspline((test.val.Y*pred_Y$km)+positionsY, pred_Y$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
interp.dat.B<-aspline((test.val.B*pred_B$km)+positionsB, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre.Y<-interp.dat.Y$x[which.min(abs(interp.dat.Y$y-0.5))]
cline_centre.B<-interp.dat.B$x[which.min(abs(interp.dat.B$y-0.5))]
positionsY<-positionsY-(cline_centre.Y-mid_point_for_pivotY)
positionsB<-positionsB-(cline_centre.B-mid_point_for_pivotB)
}

```

```

interp.dat.Y<-aspline((test.val.Y*pred_Y$km)+positionsY, pred_Y$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
Y_cline_slope<-max_slope(interp.dat.Y$y)*100 #times by 100 to bring onto km scale
Y_cline_width<-(1/Y_cline_slope)
interp.dat.B<-aspline((test.val.B*pred_B$km)+positionsB, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
B_cline_slope<-max_slope(1-interp.dat.B$y)*100 #times by 100 to bring onto km scale
B_cline_width<-(1/B_cline_slope)

Y.vs.B_ratio.upper<-Y_cline_width/B_cline_width

#Find the lower bound
##starting ratio of widths
Y.vs.B_ratio<-results$Y_cline_widths/results$B_cline_widths

#for the Y cline
mid_point_for_pivotY<-results$Y_centres
positionsY<-results$positions_Y
test.val.Y<-results$stretch_factors_Y
#for the B cline
mid_point_for_pivotB<-results$B_centres
positionsB<-results$positions_B
test.val.B<-results$stretch_factors_B

starting_likelihood<-func_likelihood_MELP_Y(c(positionsY,test.val.Y)) +
func_likelihood_MELP_B(c(positionsB,test.val.B))

while(func_likelihood_MELP_Y(c(positionsY,test.val.Y)) +
func_likelihood_MELP_B(c(positionsB,test.val.B)) > starting_likelihood - 2 )
{test.val.Y<-test.val.Y-0.0001
test.val.B<-test.val.B+0.0001
interp.dat.Y<-aspline((test.val.Y*pred_Y$km)+positionsY, pred_Y$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
interp.dat.B<-aspline((test.val.B*pred_B$km)+positionsB, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10)
cline_centre.Y<-interp.dat.Y$x[which.min(abs(interp.dat.Y$y-0.5))]
cline_centre.B<-interp.dat.B$x[which.min(abs(interp.dat.B$y-0.5))]
positionsY<-positionsY-(cline_centre.Y-mid_point_for_pivotY)
positionsB<-positionsB-(cline_centre.B-mid_point_for_pivotB)
}
interp.dat.Y<-aspline((test.val.Y*pred_Y$km)+positionsY, pred_Y$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
Y_cline_slope<-max_slope(interp.dat.Y$y)*100 #times by 100 to bring onto km scale
Y_cline_width<-(1/Y_cline_slope)
interp.dat.B<-aspline((test.val.B*pred_B$km)+positionsB, pred_B$freq, xout=
seq(min(obs$km),max(obs$km),0.01), degree=10) #generate interpolated values at a fine scale (to 10m)
B_cline_slope<-max_slope(1-interp.dat.B$y)*100 #times by 100 to bring onto km scale
B_cline_width<-(1/B_cline_slope)

```

```
Y.vs.B_ratio.lower<=-Y_cline_width/B_cline_width
```

Chazuta	S6.56654 W76.12738	dd	cc	ss
Chazuta	S6.56156 W76.12345	dd	cc	ss
Chazuta	S6.55604 W76.12113	dd	cc	ss
Chazuta	S6.56654 W76.12749	dd	cc	ss
Chazuta	S6.56654 W76.12749	dd	cc	ss
Chazuta	S6.56654 W76.12749	dd	cc	ss
Chazuta	S6.56658 W76.12732	dd	cc	ss
Chazuta	S6.56380 W76.12621	dd	cc	ss
Chazuta	S6.56380 W76.12621	dd	cc	ss
Chazuta	S6.56345 W76.12596	dd	cc	ss
Chazuta	S6.56311 W76.12593	dd	cc	ss
Chazuta	S6.56379 W76.12296	dd	cc	ss
Chazuta	S6.56654 W76.12749	dd	cc	ss
Chazuta	S6.55865 W76.12026	dd	cc	ss
Chazuta	S6.55865 W76.12026	dd	cc	ss
Chazuta	S6.55865 W76.12026	dd	cc	ss
Chazuta	S6.55865 W76.12026	dd	cc	ss
Chazuta	S6.55865 W76.12026	dd	cc	ss
Chazuta	S6.56379 W76.12296	dd	cc	ss
Chazuta	S6.56379 W76.12296	dd	cc	ss
Chazuta	S6.55865 W76.12026	dd	cc	ss
Chazuta	S6.55865 W76.12026	dd	cc	ss
Chazuta	S6.55865 W76.12026	dd	cc	ss
Chazuta	S6.55865 W76.12026	dd	cc	ss
Chazuta	S6.55865 W76.12026	dd	cc	ss
Chazuta	S6.56379 W76.12296	dd	cc	ss
Chazuta	S6.56379 W76.12296	dd	cc	ss
Chazuta	S6.56379 W76.12296	dd	cc	ss
Chazuta	S6.56379 W76.12296	dd	cc	ss
Chazuta	S6.56532 W76.12724	dd	cc	ss
Chazuta	S6.56532 W76.12724	dd	cc	ss
Chazuta	S6.56379 W76.12296	dd	cc	ss
Chazuta	S6.56379 W76.12296	dd	cc	ss
Chazuta	S6.56379 W76.12296	dd	cc	ss
Davicillo	S6.25082 W76.26954	DD	cc	S-
Davicillo	S6.24880 W76.26436	DD	C-	Ss
Davicillo	S6.25061 W76.26853	DD	C-	S-
Davicillo	S6.25080 W76.26948	DD	C-	S-
Davicillo	S6.25080 W76.26948	DD	C-	S-
Davicillo	S6.25080 W76.26948	DD	C-	S-
Davicillo	S6.25080 W76.26948	DD	C-	S-
Davicillo	S6.25080 W76.26948	Dd	C-	SS
Davicillo	S6.25080 W76.26948	DD	C-	S-
Davicillo	S6.25087 W76.26925	DD	C-	S-
Davicillo	S6.25135 W76.26749	DD	C-	S-
Davicillo	S6.25135 W76.26749	Dd	C-	SS

Davicillo	S6.24880 W76.26441			DD	C-	S-
Davicillo	S6.25135 W76.26749			DD	C-	S-
Davicillo	S6.24880 W76.26441			DD	C-	S-
Davicillo	S6.24880 W76.26441			DD	C-	S-
Davicillo	S6.24880 W76.26441			DD	C-	S-
Davicillo	S6.24880 W76.26441			DD	C-	S-
Davicillo	S6.24880 W76.26441			DD	C-	S-
Davicillo	S6.25080 W76.26986			DD	C-	S-
Davicillo	S6.25135 W76.26749			DD	C-	S-
Davicillo	S6.25135 W76.26749			DD	C-	S-
Davicillo		S6.248668 W76.264448	±150m	DD	C-	S-
Davicillo		S6.248668 W76.264448	±150m	DD	C-	S-
Davicillo		S6.248668 W76.264448	±150m	DD	C-	S-
Davicillo	S6.24886 W76.26434			DD	C-	S-
Davicillo	S6.24886 W76.26434			DD	C-	S-
Davicillo	S6.24886 W76.26434			DD	C-	S-
Davicillo	S6.24886 W76.26434			DD	C-	S-
Davicillo	S6.24886 W76.26434			DD	C-	S-
Davicillo		S6.248668 W76.264448	±150m	DD	C-	S-
Davicillo	S6.25127 W76.26755			DD	C-	S-
Davicillo	S6.25127 W76.26755			DD	C-	S-
Davicillo	S6.25127 W76.26755			DD	C-	S-
Davicillo	S6.25127 W76.26755			DD	C-	S-
Davicillo	S6.25061 W76.26843			Dd	C-	S-
Davicillo	S6.25061 W76.26843			DD	C-	S-
Davicillo	S6.25061 W76.26843			DD	C-	S-
Davicillo	S6.25061 W76.26843			DD	C-	S-
Davicillo	S6.25061 W76.26843			DD	C-	S-
Davicillo	S6.25061 W76.26843			DD	C-	S-
Davicillo	S6.25061 W76.26843			DD	C-	S-
Davicillo	S6.25061 W76.26843			DD	C-	S-
Davicillo	S6.249882 W76.266064			DD	C-	S-
Davicillo	S6.249882 W76.266064			DD	C-	S-
Davicillo	S6.24881 W76.26441			Dd	C-	S-
Davicillo	S6.24881 W76.26441			DD	C-	S-
Davicillo	S6.24871 W76.26431			DD	C-	S-
El pongo del Cainarachi		S6.338617 W76.288283	±50m	dd	C-	S-
El Tunel	S6.44850 W76.28728			dd	cc	ss
El Tunel	S6.45423 W76.28515			dd	cc	Ss
El Tunel	S6.45551 W76.28430			dd	cc	ss
El Tunel	S6.45551 W76.28430			dd	cc	ss
El Tunel		S6.453253 W76.287319	±100m	dd	cc	ss
El Tunel	S6.45289 W76.28637			dd	cc	ss
El Tunel	S6.45359 W76.28789			dd	cc	ss
El Tunel	S6.45396 W76.28547			dd	cc	ss
El Tunel	S6.45396 W76.28547			dd	cc	Ss
El Tunel	S6.45396 W76.28547			dd	cc	ss

EI Tunel	S6.45396 W76.28547			dd	cc	ss
EI Tunel	S6.45673 W76.28439			dd	cc	ss
EI Tunel	S6.45673 W76.28439			dd	cc	ss
EI Tunel	S6.45673 W76.28439			dd	cc	Ss
EI Tunel	S6.45673 W76.28439			dd	cc	ss
EI Tunel	S6.45673 W76.28439			dd	cc	ss
EI Tunel	S6.45785 W76.28507			dd	cc	ss
EI Tunel	S6.45785 W76.28507			dd	cc	ss
EI Tunel	S6.45785 W76.28507			dd	cc	ss
EI Tunel	S6.45785 W76.28507			Dd	cc	Ss
EI Tunel	S6.45669 W76.28441			dd	cc	ss
EI Tunel	S6.45588 W76.28398			dd	cc	ss
EI Tunel	S6.45588 W76.28398			dd	cc	ss
EI Tunel		S6.454000 W76.285439	±150m	dd	cc	ss
EI Tunel	S6.45277 W76.28632			dd	cc	ss
EI Tunel - km32.5 trail	S6.44894 W76.28132			dd	cc	ss
EI Tunel - km32.5 trail	S6.45628 W76.27714			dd	cc	ss
Florida	S6.26930 W76.75069			dd	cc	ss
Florida	S6.26930 W76.75069			dd	cc	ss
Florida	S6.26930 W76.75069			dd	cc	ss
Florida	S6.26930 W76.75069			dd	cc	ss
Florida	S6.26930 W76.75069			dd	cc	ss
Florida	S6.26930 W76.75069			dd	cc	ss
Florida	S6.26930 W76.75069			dd	cc	ss
Florida	S6.26930 W76.75069			dd	cc	ss
Florida	S6.26930 W76.75069			dd	cc	ss
Florida	S6.26930 W76.75069			dd	cc	ss
Florida	S6.26930 W76.75069			dd	cc	ss
Florida	S6.26930 W76.75069			dd	cc	ss
Florida	S6.26930 W76.75069			dd	cc	ss
Florida	S6.26930 W76.75069			dd	cc	ss
Florida		S6.267960 W76.749276	±300m	dd	cc	ss
Florida		S6.267960 W76.749276	±300m	dd	cc	ss
Florida		S6.267960 W76.749276	±300m	dd	cc	ss
Florida	S6.266642 W76.74789			dd	cc	ss
Fraser's Pardalinius Trail	S6.47669 W76.35183			dd	cc	ss
Fraser's Pardalinius Trail	S6.47668 W76.35199			dd	cc	ss
Fraser's Pardalinius Trail		S6.47622 W76.35176	±150m	dd	cc	ss
Fraser's Pardalinius Trail	S6.47617 W76.35177			dd	cc	ss
Fraser's Pardalinius Trail	S6.47730 W76.35170			dd	cc	ss
Fraser's Pardalinius Trail	S6.47676 W76.35184			dd	cc	ss
Fraser's Pardalinius Trail	S6.47234 W76.35004			dd	cc	ss
Fraser's Pardalinius Trail	S6.46992 W76.34619			dd	cc	ss
Fraser's Pardalinius Trail	S6.46992 W76.34619			dd	cc	ss
Fraser's Pardalinius Trail	S6.47045 W76.34783			dd	cc	ss
Jepelacio	S6.10206 W76.94178			dd	cc	ss
La Antena	S6.45468 W76.29933			dd	cc	ss
La Antena	S6.45468 W76.29933			dd	cc	ss
La Antena	S6.454679 W76.299326			dd	cc	ss

La Antena	S6.45431 W76.29889	dd	cc	ss
La Antena	S06.45431 W076.29889	dd	cc	ss
La Carachamera	S6.43152 W76.25316	dd	cc	ss
La Carachamera	S6.43166 W76.25295	dd	cc	ss
Micaela Bastida km6.1	S5.94594 W76.24637	DD	C-	S-
Micaela Bastida km6.1	S5.94582 W76.24720	DD	C-	S-
Micaela Bastida km6.1	S5.94576 W76.24687	DD	C-	S-
Micaela Bastida km6.1	S5.94583 W76.24658	DD	C-	S-
Micaela Bastida km6.1	S5.94583 W76.24658	DD	C-	S-
Micaela Ba-tida km 4.8	S5.960674 W76.244929	DD	C-	S-
Micaela Ba-tida km 4.8	S5.954868 W76.239834	DD	C-	S-
Micaela Ba-tida km 6.1	S5.9458 W76.24552	DD	C-	S-
Micaela Ba-tida km 6.1	S5.94587 W76.24526	DD	C-	S-
Micaela Ba-tida km 6.1	S5.94587 W76.24526	DD	C-	S-
Micaela Ba-tida km 6.1	S5.945549 W76.245444	DD	C-	S-
Micaela Ba-tida km 6.1	S5.9456 W76.24495	DD	C-	S-
Moyobamba Antenna	S6.09736 W76.98174	dd	cc	ss
Moyobamba Antenna	S6.09736 W76.98174	dd	cc	ss
Moyobamba Antenna	S6.09701 W76.98145	dd	cc	ss
Moyobamba Antenna	S6.09736 W76.98174	dd	cc	ss
Moyobamba Antenna	S6.09736 W76.98174	dd	cc	ss
Moyobamba Antenna	S6.09588 W76.97728	dd	cc	ss
Moyobamba Antenna	S6.09602 W76.97742	dd	cc	ss
Moyobamba Antenna	S6.09428 W76.97146	dd	cc	ss
Moyobamba Antenna	S6.09512 W76.97936	dd	cc	ss
Moyobamba Antenna	S6.09588 W76.97728	dd	cc	ss
Moyobamba Antenna	S6.09512 W76.97936	dd	cc	ss
Moyobamba Antenna	S6.09445 W76.97080	dd	cc	ss
Pongo	S06.29917 W076.23529	DD	C-	S-
Pongo - Barranquita km3.5	S6.31371 W76.25829	DD	C-	S-
Pongo - Barranquita km3.5	S6.30868 W76.25721	Dd	C-	S-
Pongo - Barranquita km3.5	S6.30868 W76.25721	DD	C-	S-
Pongo - Barranquitas km6	S6.29990 W76.23557	DD	C-	S-
Pongo - Barranquitas km6.3	S6.297089 W76.235164	DD	C-	S-
Pongo - Barranquitas: either km 6.3 or km7.2	S6.290752 W76.235435	DD	C-	Ss
Pongo Cataratas	S6.35276 W76.30186	dd	C-	Ss
Pongo Cataratas	S6.34888 W76.30000	dd	Cc	ss
Pongo Cataratas	S6.35044 W76.30023	dd	cc	ss
Pongo Cataratas	S6.34950 W76.30008	dd	Cc	ss
Pongo Cataratas	S6.34950 W76.30008	Dd	C-	Ss
Pongo Cataratas	S6.34947 W76.29986	Dd	C-	SS
Pongo Cataratas	S6.34947 W76.29986	DD	cc	ss
Pongo Cataratas	S6.34947 W76.29986	dd	Cc	ss
Pongo Cataratas	S6.35254 W76.30157	dd	Cc	ss
Pongo Cataratas	S6.35018 W76.30008	Dd	C-	Ss
Pongo Cataratas	S6.348430 W76.302170 ±50m	Dd	C-	S-

Pongo Cataratas		S6.348430 W76.302170	±50m	dd	Cc	ss
Pongo Cataratas		S6.348430 W76.302170	±50m	Dd	cc	Ss
Pongo Cataratas		S6.348430 W76.302170	±50m	dd	cc	S-
Pongo Cataratas		S6.348430 W76.302170	±50m	dd	C-	S-
Pongo Cataratas	S6.35114 W76.30119			Dd	Cc	ss
Pongo Cataratas	S6.35081 W76.30034			Dd	cc	ss
Pongo de Cainarache	S6.33628 W76.27858			DD	C-	Ss
Pongo de Cainarache, south of	S6.33903 W76.28200			DD	C-	Ss
Pongo-Barranquitas km5.9	S6.30507 W76.23275			DD	C-	Ss
Pongo-Barranquitas km5.9	S6.30575 W76.23215			Dd	C-	Ss
Pongo-Barranquitas km5.9	S6.30733 W76.22989			DD	C-	S-
Pongo-Barranquitas km5.9	S6.30733 W76.22989			DD	C-	S-
Pongo-Barranquitas km5.9	S6.30733 W76.22989			Dd	C-	SS
Pongo-Barranquitas km5.9	S6.30733 W76.22989			DD	C-	S-
Pongo-Barranquitas km5.9	S6.30615 W76.22977			DD	C-	Ss
Pongo-Barranquitas km5.9	S6.30531 W76.23254			DD	C-	S-
Pongo-Barranquitas km5.9	S6.30355 W76.23367			dd	C-	Ss
Pongo-Barranquitas km5.9	S6.30079 W76.23603			DD	C-	S-
Pongo-Barranquitas km5.9	S6.30272 W76.23368			DD	cc	S-
Pongo-Barranquitas km5.9	S6.30079 W76.23603			DD	C-	S-
Pongo-Barranquitas km5.9	S6.30282 W76.23377			Dd	C-	Ss
Pongo-Barranquitas km5.9	S6.30282 W76.23377			DD	C-	S-
Pongo-Barranquitas km5.9	S6.302929 W76.234025			DD	C-	S-
Pongo-Barranquitas km5.9	S6.30472 W76.23311			DD	C-	S-
Pongo-Barranquitas km5.9	S6.30347 W76.23379			DD	C-	S-
Pongo-Barranquitas km5.9	S6.306407 W76.230117			DD	C-	S-
Pongo-Barranquitas km5.9	S6.30711 W76.22987			DD	C-	S-
Pongo-Barranquitas km5.9	S6.30527 W76.23254			DD	C-	S-
Pongo-Barranquitas km5.9	S6.30527 W76.23254			DD	C-	S-
Pongo-Barranquitas km5.9	S6.306407 W76.230117			DD	C-	S-
Pongo-Barranquitas km5.9	S6.30489 W76.23293			DD	C-	S-
Pongo-Barranquitas km5.9	S6.30716 W76.22979			DD	C-	S-
Pongo-Barranquitas km5.9	S6.30716 W76.22979			DD	C-	S-
Pongo-Barranquitas km6	S6.29990 W76.23557			DD	C-	S-
Pongo-Barranquitas km6		S6.298590 W76.237088	±200m	DD	C-	Ss
Pongo-Barranquitas km6		S6.298590 W76.237088	±200m	DD	C-	S-
Pongo-Barranquitas km6	S6.29960 W76.23534			DD	C-	S-
Pongo-Barranquitas km6.3				DD	C-	S-
Pongo-Barranquitas km6.3				DD	C-	S-
Rancho Vista	S6.46147 W76.35926			dd	cc	ss
Rancho Vista	S6.46002 W76.35941			dd	cc	ss
Rancho Vista	S6.45500 W76.35906			dd	cc	ss
Rancho Vista	S6.45635 W76.35852			dd	cc	ss
Rancho Vista	S6.45827 W76.35914			dd	cc	ss
Rancho Vista	S6.45657 W76.35836			dd	cc	ss
Rancho Vista	S6.45812 W76.35931			dd	cc	ss

Rancho Vista	S6.45812 W76.35931			dd	cc	ss
Rancho Vista	S6.45515 W76.35909			dd	cc	ss
Rancho Vista	S6.45159 W76.35851			dd	cc	ss
Rancho Vista	S6.45160 W76.35849			dd	cc	ss
Rancho Vista		S6.45515 W76.35909	±700m	dd	cc	ss
Rio Shilcayo	S6.45599 W76.34688			dd	cc	ss
Rio Shilcayo	S6.45508 W76.34611			dd	cc	ss
Rio Shilcayo	S6.45508 W76.34611			dd	cc	ss
Rio Shilcayo	S6.45508 W76.34611			dd	cc	ss
Rio Shilcayo	S6.45508 W76.34611			dd	cc	ss
Rio Shilcayo	S6.45508 W76.34611			dd	cc	ss
Rio Shilcayo	S6.45539 W76.34479			dd	cc	ss
Rio Shilcayo	S6.46708 W76.35302			dd	cc	ss
Rio Shilcayo	S6.45165 W76.34596			dd	cc	ss
Rio Shilcayo	S6.45508 W76.34611			dd	cc	ss
Rio Shilcayo	S6.45508 W76.34611			dd	cc	ss
Rio Shilcayo	S6.46402 W76.35125			dd	cc	ss
Rio Shilcayo	S6.45849 W76.34896			dd	cc	ss
Rio Shilcayo	S6.46913 W76.35440			dd	cc	ss
Rio Shilcayo	S6.46575 W76.35265			dd	cc	ss
Rio Shilcayo	S6.47382 W76.35425			dd	cc	ss
Rio Shilcayo		S6.46708 W76.35302	±150m	dd	cc	ss
Rio Shilcayo	S6.45735 W76.35266			dd	cc	ss
Rio Shilcayo	S6.45929 W76.35248			dd	cc	ss
Rio Shilcayo	S6.46001 W76.35054			dd	cc	ss
Rio Shilcayo	S6.45716 W76.35261			dd	cc	ss
Rio Shilcayo	S6.45658 W76.35249			dd	cc	ss
Rio Shilcayo	S6.45574 W76.35156			dd	cc	ss
Rio Shilcayo	S6.45574 W76.35156			dd	cc	ss
Rio Shilcayo	S6.45577 W76.35130			dd	cc	ss
Rio Shilcayo	S6.45453 W76.35068			dd	cc	ss
Rio Shilcayo	S6.45339 W76.34019			dd	cc	ss
Rio Shilcayo	S6.45417 W76.34190			dd	cc	ss
Rio Shilcayo	S6.45534 W76.34423			dd	cc	ss
Rio Shilcayo	S6.45271 W76.34011			dd	cc	ss
Rio Shilcayo	S6.45271 W76.34011			dd	cc	ss
Rio Shilcayo	S6.45271 W76.34011			dd	cc	ss
Rio Shilcayo	S6.44830 W76.34538			Dd	cc	ss
Rio Shilcayo	S6.44830 W76.34538			dd	cc	ss
Rio Shilcayo, east bank	S6.47451 W76.35384			dd	cc	ss
Rio Shilcayo, east bank	S6.46839 W76.35114			dd	cc	ss
Rio Shilcayo, east bank	S6.47451 W76.35384			dd	cc	ss
Rio Shilcayo, east bank	S6.46560 W76.34767			Dd	cc	ss
Rumipata	S6.07788 W76.96931			dd	cc	ss
Sachawawa		S6.268356 W76.299331	±50m	DD	Cc	ss
San Antonio de Cumbaza	S6.40201 W76.39571			dd	cc	ss

San Antonio de Cumbaza	S6.40779 W76.40247			dd	cc	ss
San Antonio de Cumbaza	S6.40465 W76.39723			dd	cc	ss
Tangarana	S6.32515 W76.69459			dd	cc	ss
Tangarana	S6.32159 W76.69527			dd	cc	ss
Tangarana	S6.32753 W76.69411			dd	cc	ss
Tangarana	S6.32159 W76.69527			dd	cc	ss
Tangarana				dd	cc	ss
Tangarana				dd	cc	ss
Tarapoto - Uruahasha	S6.46713 W76.33475			dd	cc	ss
Tarapoto - Uruahasha	S6.46713 W76.33475			dd	cc	ss
Tarapoto - Uruahasha	S6.46713 W76.33475			dd	cc	ss
Tarapoto - Uruahasha	S6.46202 W76.33473			dd	cc	ss
Tarapoto - Uruahasha	S6.46716 W76.33639			dd	cc	ss
Tarapoto - Uruahasha		S6.463401 W76.334952	±400m	dd	cc	ss
Tarapoto - Uruahasha	S6.47588 W76.34748			dd	cc	ss
Tarapoto - Uruahasha	S6.47292 W76.34485			dd	cc	ss
Tarapoto - Uruahasha	S6.47182 W76.34204			dd	cc	ss
Tarapoto - Uruahasha	S6.47175 W76.34043			dd	cc	ss
Tarapoto - Uruahasha	S6.46867 W76.33675			dd	cc	ss
Tarapoto - Uruahasha	S6.46773 W76.33631			dd	cc	ss
Tarapoto - Uruahasha	S6.46773 W76.33631			dd	cc	ss
Tarapoto - Uruahasha	S6.46718 W76.33637			dd	cc	ss
Tarapoto - Uruahasha	S6.46718 W76.33637			dd	cc	ss
Tarapoto - Uruahasha	S6.46595 W76.33492			dd	cc	ss
Tarapoto - Uruahasha	S6.46705 W76.33477			dd	cc	ss
Tarapoto - Uruahasha	S6.46705 W76.33477			dd	cc	ss
Tarapoto - Uruahasha	S6.46699 W76.33473			dd	cc	ss
Tarapoto - Uruahasha	S6.46699 W76.33473			dd	cc	ss
Tarapoto - Uruahasha	S6.46649 W76.33431			dd	cc	ss
Tarapoto - Uruahasha	S6.46632 W76.33398			dd	cc	ss
Tarapoto - Uruahasha	S6.46587 W76.33240			dd	cc	ss
Tarapoto - Uruahasha	S6.46278 W76.33546			dd	cc	ss
Tarapoto - Uruahasha	S6.46161 W76.33429			dd	cc	ss
Tarapoto - Uruahasha	S6.47213 W76.34123			dd	cc	ss
Tarapoto - Uruahasha		S6.466808 W76.339537	±350m	dd	cc	ss
Tarapoto - Uruahasha		S6.466808 W76.339537	±350m	dd	cc	ss
Tarapoto - Uruahasha	S6.46922 W76.34145			dd	cc	ss
Tarapoto - Uruahasha	S6.46922 W76.34145			dd	cc	ss
Tarapoto - Uruahasha	S6.46080 W76.33346			dd	cc	ss
Tarapoto - Uruahasha	S6.45661 W76.32987			dd	cc	ss
Tarapoto - Uruahasha	S6.46904 W76.33703			dd	cc	ss
Tarapoto - Uruahasha	S6.46857 W76.33663			dd	cc	ss
Tarapoto - Uruahasha	S6.46857 W76.33663			dd	cc	ss
Tarapoto - Uruahasha	S6.46780 W76.33633			dd	cc	ss
Tarapoto - Uruahasha	S6.46768 W76.33639			dd	cc	ss
Tarapoto - Uruahasha	S6.46768 W76.33639			dd	cc	ss

Tarapoto - Uruahasha	S6.46690 W76.33641			dd	cc	ss
Tarapoto - Uruahasha	S6.46690 W76.33641			dd	cc	ss
Tarapoto - Uruahasha	S6.46690 W76.33641			dd	cc	ss
Tarapoto - Uruahasha	S6.46148 W76.33415			dd	cc	ss
Tarapoto - Uruahasha	S6.45661 W76.32987			Dd	cc	ss
Tarapoto - Uruahasha	S6.45661 W76.32987			dd	cc	ss
Tarapoto - Uruahasha	S6.45753 W76.33072			dd	cc	ss
Tarapoto - Uruahasha	S6.45753 W76.33072			dd	cc	ss
Tarapoto - Uruahasha	S6.45753 W76.33072			dd	cc	ss
Tarapoto - Uruahasha	S6.45753 W76.33072			dd	cc	ss
Tarapoto - Uruahasha	S6.45753 W76.33072			dd	cc	ss
Tarapoto - Uruahasha	S6.46220 W76.33501			dd	cc	ss
Tarapoto - Uruahasha	S6.46276 W76.33602			dd	cc	ss
Tarapoto - Uruahasha	S6.46276 W76.33602			dd	cc	ss
Tarapoto - Uruahasha	S6.46429 W76.33629			dd	cc	ss
Tarapoto - Uruahasha	S6.45753 W76.33072			dd	cc	ss
Tarapoto - Yurimaguas (km 68)		S6.233789 W76.277508	±400m	Dd	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.28891 W76.26358			Dd	CC	SS
Tarapoto - Yurimaguas km 59.3	S6.29829 W76.28022			DD	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29808 W76.28087			DD	cc	Ss
Tarapoto - Yurimaguas km 59.3	S6.29824 W76.27996			Dd	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29815 W76.27898			DD	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29860 W76.27592			DD	cc	S-
Tarapoto - Yurimaguas km 59.3	S6.29860 W76.27592			Dd	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29860 W76.27592			DD	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29860 W76.27592			DD	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29860 W76.27592			Dd	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29831 W76.28007			DD	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29287 W76.28942			DD	Cc	Ss
Tarapoto - Yurimaguas km 59.3	S6.29417 W76.29009			DD	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29417 W76.29009			DD	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29826 W76.28058			DD	C-	Ss
Tarapoto - Yurimaguas km 59.3	S6.29829 W76.28022			DD	cc	S-
Tarapoto - Yurimaguas km 59.3	S6.29829 W76.28022			DD	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29829 W76.28022			Dd	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29817 W76.27910			DD	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29814 W76.28053			Dd	Cc	Ss
Tarapoto - Yurimaguas km 59.3	S6.29835 W76.28033			dd	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29812 W76.28049			DD	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29812 W76.28049			DD	C-	Ss
Tarapoto - Yurimaguas km 59.3	S6.29835 W76.28033			Dd	C-	S-
Tarapoto - Yurimaguas km 59.3		S6.298124 W76.276972	±100m	DD	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29647 W76.27267			DD	C-	S-
Tarapoto - Yurimaguas km 59.3	S6.29856 W76.27596			Dd	C-	Ss
Tarapoto - Yurimaguas km 61.1	S6.28633 W76.27614			Dd	C-	S-
Tarapoto - Yurimaguas km 68.2	S6.23225 W76.27182			DD	cc	S-

Tarapoto - Yurimaguas km 68.2	S6.23006 W76.29049		DD	C-	S-
Tarapoto - Yurimaguas km 73.3	S6.19115 W76.26257		DD	C-	S-
Tarapoto - Yurimaguas km 73.3	S6.19115 W76.26257		DD	C-	S-
Tarapoto - Yurimaguas km 73.3	S6.19213 W76.26287		DD	C-	S-
Tarapoto - Yurimaguas km 73.3	S6.19213 W76.26287		DD	C-	S-
Tarapoto - Yurimaguas km 73.3	S6.19256 W76.26290		DD	CC	Ss
Tarapoto - Yurimaguas km 73.3	S6.19115 W76.26257		DD	Cc	Ss
Tarapoto - Yurimaguas km 73.3	S6.19115 W76.26257		DD	C-	S-
Tarapoto - Yurimaguas km 73.3	S6.19115 W76.26257		DD	C-	S-
Tarapoto - Yurimaguas km 73.3	S6.19115 W76.26257		DD	C-	S-
Tarapoto - Yurimaguas km 73.3	S6.19252 W76.26296		DD	C-	Ss
Tarapoto - Yurimaguas km 73.3		S6.19120 W76.26254 ±100m	DD	C-	S-
Tarapoto - Yurimaguas km25	S6.41067 W76.32610		dd	cc	ss
Tarapoto - Yurimaguas km25	S6.41066 W76.32611		dd	cc	ss
Tarapoto - Yurimaguas km47.5	S6.37721 W76.28881		dd	cc	Ss
Tarapoto - Yurimaguas km47.5	S6.37670 W76.28859		dd	cc	ss
Tarapoto - Yurimaguas km49	S6.37785 W76.28940		Dd	cc	ss
Tarapoto - Yurimaguas km49	S6.37308 W76.28523		Dd	cc	ss
Tarapoto - Yurimaguas km49	S6.37785 W76.28940		dd	cc	ss
Tarapoto - Yurimaguas km49	S6.37785 W76.28940		dd	cc	ss
Tarapoto - Yurimaguas km49	S6.37785 W76.28940		dd	cc	ss
Tarapoto - Yurimaguas km49		S6.376850 W76.288282 ±50m	dd	cc	ss
Tarapoto - Yurimaguas km49		S6.372729 W76.284959 ±100m	dd	cc	ss
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37027 W76.29170		dd	cc	ss
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37027 W76.29170		dd	cc	ss
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.36837 W76.29188		dd	cc	SS
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37005 W76.29147		dd	cc	Ss
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37005 W76.29147		dd	cc	ss
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.36826 W76.29173		dd	cc	ss
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.36825 W76.29183		dd	cc	ss
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37178 W76.29251		dd	cc	ss
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37093 W76.29245		Dd	Cc	ss
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37036 W76.29160		dd	cc	Ss
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37039 W76.29160		Dd	Cc	ss
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.36826 W76.29173		Dd	cc	ss
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.36826 W76.29173		dd	cc	ss
Tarapoto - Yurimaguas km49.5	S6.36634 W76.29204		dd	cc	Ss
Tarapoto - Yurimaguas km49.5	S6.36631 W76.29191		dd	cc	ss
Tarapoto - Yurimaguas km50	S6.36250 W76.29433		Dd	C-	Ss
Tarapoto - Yurimaguas km50	S6.36250 W76.29433		dd	cc	SS
Tarapoto - Yurimaguas km50	S6.36249 W76.29431		dd	cc	Ss
Tarapoto - Yurimaguas km50	S6.36249 W76.29431		dd	cc	ss
Tarapoto - Yurimaguas km50	S6.36249 W76.29431		dd	cc	ss
Tarapoto - Yurimaguas km50	S6.36249 W76.29431		dd	C-	Ss
Tarapoto - Yurimaguas km50	S6.36250 W76.29422		Dd	Cc	ss
Tarapoto - Yurimaguas km50	S6.36250 W76.29422		dd	cc	ss

Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32174 W76.28149			Dd	Cc	ss
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32064 W76.28068			dd	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32118 W76.28120			Dd	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32174 W76.28152			Dd	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.31510 W76.27191			DD	Cc	Ss
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.31539 W76.28124			Dd	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.31510 W76.27191			DD	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.31590 W76.28014			DD	C-	Ss
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.31691 W76.27428			DD	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32049 W76.27571			DD	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32049 W76.27571			DD	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32049 W76.27571			dd	Cc	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32049 W76.27571			Dd	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32049 W76.27571			DD	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32049 W76.27571			DD	cc	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32049 W76.27571			Dd	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32049 W76.27571			DD	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32049 W76.27571			Dd	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32049 W76.27571			DD	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32140 W76.28282			Dd	C-	ss
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32145 W76.28166			Dd	C-	Ss
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32170 W76.28155			dd	Cc	ss
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32152 W76.28169			Dd	C-	S-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32152 W76.28169			DD	C-	Ss
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32152 W76.28169			DD	Cc	Ss
Tarapoto - Yurimaguas km73.3	S6.19198 W76.26285			DD	C-	Ss
Tarapoto - Yurimaguas km73.3		S6.191905 W76.261480	±150m	DD	C-	Ss
Tarapoto - Yurimaguas km73.3		S6.191905 W76.261480	±150m	DD	C-	S-
Tarapoto-Yurimagua- km 47.5 Trail to pylon-	S6.37773 W76.28932			dd	cc	ss
Tarapoto-Yurimagua- km 47.5 Trail to pylon-	S6.37773 W76.28932			dd	cc	ss
Tarapoto-Yurimagua- km 47.5 Trail to pylon-	S6.37773 W76.28932			Dd	cc	ss
Tioyacu	S5.99243 W77.28818			dd	cc	ss
Tunel	S06.45673 W076.28445			dd	cc	ss
Tunel	S06.45551 W076.28430			dd	cc	ss
Tunel	S06.45551 W076.28430			dd	cc	ss
Tunel	S06.45551 W076.28430			Dd	C-	ss
Urahuasha	S6.46697 W76.33475			dd	cc	ss
Urahuasha	S6.46697 W76.33475			dd	cc	ss
Urahuasha	S6.46697 W76.33475			dd	cc	ss
Urahuasha	S6.46697 W76.33475			dd	cc	ss
Urahuasha	S06.47588 W076.34748			dd	cc	ss
Urahuasha	S06.47588 W076.34748			dd	cc	ss
Waqanki	S06.08715 W076.98433			dd	cc	ss
Waqanki	S06.08800 W076.98580			dd	cc	ss

Waqanki	S06.08715 W076.98433	dd	cc	ss
Waqanki	S06.08715 W076.98433	dd	cc	ss
Waqanki	S06.08800 W076.98580	dd	cc	ss

[illegible]

Casa de Orlando	S6.44280 W76.33571		B-	dd	nn	yy
Casa de Orlando	S6.44280 W76.33571		B-	dd	nn	yy
Casa de Orlando	S6.44280 W76.33571		B-	dd	nn	yy
Cesar_trail	S06.45690 W076.30921		B-	dd	nn	yy
Chazuta	S6.55324 W76.12083		B-	dd	nn	yy
Chazuta	S6.56654 W76.12749		B-	dd	nn	yy
Chazuta	S6.55908 W76.12574		B-	dd	nn	yy
Chazuta	S6.56417 W76.12629		B-	dd	nn	yy
Chazuta	S6.56654 W76.12749		B-	dd	nn	yy
Chazuta	S6.56654 W76.12749		B-	dd	nn	yy
Davicillo	S6.24880 W76.26441		bb	D-	N-	Y-
El Tunel	S6.45289 W76.28637		B-	dd	nn	yy
El Tunel	S6.45289 W76.28637		B-	D-	nn	yy
El Tunel	S6.45314 W76.28763		B-	D-	nn	yy
El Tunel	S6.45544 W76.28453		B-	dd	nn	yy
El Tunel	S6.45277 W76.28632		B-	dd	nn	yy
El Tunel		S6.454222 W76.285140 ±200m	B-	dd	nn	yy
El Tunel	S6.45289 W76.28637		B-	dd	nn	yy
El Tunel	S6.45289 W76.28637		B-	dd	nn	yy
El Tunel	S6.45289 W76.28637		B-	dd	nn	yy
El Tunel	S6.45289 W76.28637		B-	dd	nn	yy
El Tunel	S6.45289 W76.28637		B-	dd	nn	yy
El Tunel	S6.45277 W76.28632		B-	dd	nn	yy
El Tunel	S6.45277 W76.28632		B-	dd	nn	yy
El Tunel	S6.45277 W76.28632		B-	dd	nn	yy
El Tunel	S6.45359 W76.28789		B-	dd	nn	yy
El Tunel		S6.453253 W76.287319 ±100m	B-	dd	nn	yy
El Tunel		S6.453253 W76.287319 ±100m	B-	dd	nn	yy
El Tunel	S6.45338 W76.28758		B-	dd	nn	yy
El Tunel	S6.45359 W76.28789		B-	dd	nn	yy
El Tunel	S6.45355 W76.28778		B-	dd	nn	yy
El Tunel	S6.45396 W76.28547		B-	dd	nn	yy
El Tunel	S6.45396 W76.28547		B-	dd	nn	yy
El Tunel	S6.45396 W76.28547		B-	dd	nn	yy
El Tunel	S6.45277 W76.28632		B-	dd	nn	yy
El Tunel	S6.45314 W76.28768		B-	dd	nn	yy
El Tunel	S6.45351 W76.28763		B-	dd	nn	yy
El Tunel	S6.45351 W76.28763		B-	dd	nn	yy
El Tunel	S6.45673 W76.28439		bb	D-	N-	Y-
El Tunel	S6.45673 W76.28439		B-	dd	nn	yy
El Tunel	S6.45673 W76.28439		B-	dd	nn	yy
El Tunel	S6.45689 W76.28453		B-	dd	nn	yy
El Tunel	S6.45673 W76.28439		B-	D-	nn	yy
El Tunel		S6.453110 W76.287176 ±100m	B-	dd	nn	yy
El Tunel	S6.45669 W76.28441		B-	D-	nn	yy
El Tunel	S6.45588 W76.28398		B-	dd	nn	yy

EI Tunel	S6.45669 W76.28441			B-	dd	nn	yy
EI Tunel	S6.45629 W76.28430			B-	dd	nn	yy
EI Tunel	S6.45629 W76.28430			B-	dd	nn	yy
EI Tunel	S6.45629 W76.28430			B-	dd	nn	yy
EI Tunel	S6.45502 W76.28477			B-	dd	nn	yy
EI Tunel	S6.45399 W76.28540			B-	dd	nn	yy
EI Tunel	S6.45327 W76.28603			B-	D-	nn	yy
EI Tunel - km32.5 trail	S6.44850 W76.28151			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.45412 W76.27756			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.44894 W76.28132			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.44894 W76.28132			B-	dd	N-	Y-
EI Tunel - km32.5 trail	S6.45437 W76.27739			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.45437 W76.27739			B-	D-	N-	Yy
EI Tunel - km32.5 trail	S6.45437 W76.27739			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.44804 W76.27225			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.43741 W76.27770			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.44880 W76.28139			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.44880 W76.28139			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.44880 W76.28139			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.44880 W76.28139			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.44880 W76.28139			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.44880 W76.28139			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.45399 W76.27764			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.45399 W76.27764			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.45434 W76.27728			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.45434 W76.27728			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.45434 W76.27728			B-	dd	nn	yy
EI Tunel - km32.5 trail	S6.43905 W76.27452			B-	dd	nn	yy
Escalera_Cesar'sTrail	S6.45353 W76.31189			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.47668 W76.35199			B-	dd	nn	yy
Fraser's Pardalinus Trail		S6.47622 W76.35176	±150m	B-	dd	nn	yy
Fraser's Pardalinus Trail		S6.47622 W76.35176	±150m	B-	dd	nn	yy
Fraser's Pardalinus Trail		S6.47622 W76.35176	±150m	B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.47693 W76.35100			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.47620 W76.35179			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.47490 W76.35118			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.47234 W76.35004			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.47034 W76.34725			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.47019 W76.34672			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.46862 W76.34629			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.46968 W76.34640			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.46992 W76.34619			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.46992 W76.34619			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.46992 W76.34619			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.46992 W76.34619			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.46992 W76.34619			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.47310 W76.35034			B-	dd	nn	yy

Fraser's Pardalinus Trail	S6.47221 W76.34998			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.47310 W76.35034			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.47339 W76.35055			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.47310 W76.35034			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.47339 W76.35055			B-	dd	nn	yy
Fraser's Pardalinus Trail	S6.47339 W76.35055			B-	dd	nn	yy
La Antena	S06.45474 W076.29936			B-	dd	nn	yy
La Antena	S6.45714 W76.29847			B-	dd	nn	yy
La Antena		S6.45468 W76.29933	accuracy unclear	B-	dd	nn	yy
La Antena	S06.45431 W076.29889			B-	dd	nn	yy
La Antena	S06.45431 W076.29889			B-	dd	nn	yy
La Antena	S06.45431 W076.29889			B-	dd	nn	yy
La Antena	S6.455593 W76.299112			B-	dd	nn	yy
La Carachamera	S6.43188 W76.25192			B-	dd	nn	yy
La Carachamera	S6.43188 W76.25192			B-	dd	nn	yy
La Carachamera	S6.43188 W76.25192			B-	dd	nn	yy
La Carachamera	S6.43782 W76.24503			B-	dd	nn	yy
La Carachamera	S6.43782 W76.24503			B-	dd	nn	yy
La Carachamera	S6.43802 W76.24488			B-	dd	nn	yy
La Carachamera	S6.43782 W76.24503			B-	dd	nn	yy
La Carachamera		S6.434777 W76.248140	±700m	B-	dd	nn	yy
La Carachamera	S6.43732 W76.24551			B-	dd	nn	yy
Micaela Bastida km6.1		S05.943371 W76.245272	±200m	bb	D-	N-	Y-
Micaela Bastida km6.1		S05.943371 W76.245272	±200m	bb	D-	N-	Y-
Micaela Bastida km6.1	S5.94577 W76.24530			bb	D-	N-	Y-
Micaela Bastida km6.1	S5.94583 W76.24658			bb	D-	N-	Y-
Micaela Ba-tida km 6.1	S5.9458 W76.24552			bb	D-	N-	Y-
Micaela Ba-tida km 6.1	S5.9458 W76.24552			bb	D-	N-	Y-
Micaela Ba-tida km 6.2	S5.945549 W76.245444			bb	D-	N-	Y-
Moyobamba Antenna	S6.09602 W76.97742			B-	dd	nn	yy
Pongo - Barranquita km3.5	S6.31371 W76.25829			bb	D-	N-	yy
Pongo - Barranquita km3.5	S6.31371 W76.25829			B-	D-	N-	Y-
Pongo - Barranquita km3.5	S6.31371 W76.25829			bb	D-	N-	Y-
Pongo - Barranquita km3.5	S6.30868 W76.25721			bb	D-	N-	Y-
Pongo - Barranquita km3.5	S6.30868 W76.25721			bb	D-	N-	Y-
Pongo - Barranquita km3.5	S6.30868 W76.25721			bb	D-	N-	Y-
Pongo - Barranquita km3.5	S6.30773 W76.25783			bb	D-	N-	Y-
Pongo - Barranquita km3.5	S6.30330 W76.25782			bb	D-	N-	Y-
Pongo - Barranquitas km7.2	S6.287826 W76.230743			bb	D-	N-	Y-
Pongo Cataratas	S6.34870 W76.30000			B-	dd	N-	Yy
Pongo Cataratas	S6.34870 W76.30000			bb	D-	N-	Y-
Pongo Cataratas	S6.34947 W76.29986			B-	D-	N-	Y-
Pongo Cataratas	S6.34870 W76.30000			B-	D-	nn	yy
Pongo Cataratas	S6.34857 W76.29997			bb	D-	Nn	Yy
Pongo Cataratas	S6.34857 W76.29997			B-	D-	N-	Yy
Pongo Cataratas	S6.34857 W76.29997			B-	dd	nn	yy

Pongo Cataratas	S6.35361 W76.30237			bb	D-	N-	Y-
Pongo Cataratas		S6.354520 W76.302640	±50m	B-	dd	nn	yy
Pongo Cataratas	S6.35028 W76.30017			B-	D-	N-	Yy
Pongo Cataratas	S6.35307 W76.30199			B-	D-	N-	yy
Pongo de Cainarache	S6.32625 W76.27248			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30733 W76.22989			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30733 W76.22989			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30733 W76.22989			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30733 W76.22989			B-	dd	nn	yy
Pongo-Barranquitas km5.9	S6.30531 W76.23254			B-	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30531 W76.23254			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30531 W76.23254			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30158 W76.23505			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30382 W76.23355			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30288 W76.23390			bb	D-	N-	yy
Pongo-Barranquitas km5.9	S6.30282 W76.23377			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30282 W76.23377			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30527 W76.23254			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30622 W76.23166			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30711 W76.22987			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30711 W76.22987			B-	D-	N-	Yy
Pongo-Barranquitas km5.9	S6.30711 W76.22987			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.306407 W76.230117			B-	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30499 W76.23290			bb	D-	N-	Y-
Pongo-Barranquitas km5.9		S06.304314 W76.233089	±500m	bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30387 W76.23347			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30079 W76.23603			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30716 W76.22979			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30716 W76.22979			B-	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30716 W76.22979			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30716 W76.22979			bb	dd	N-	Yy
Pongo-Barranquitas km5.9	S6.30716 W76.22979			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30716 W76.22979			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30716 W76.22979			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30716 W76.22979			bb	D-	N-	Y-
Pongo-Barranquitas km5.9	S6.30513 W76.23262			bb	D-	N-	Y-
Rancho Vista	S6.45131 W76.35851			B-	dd	nn	yy
Rancho Vista	S6.45238 W76.35845			B-	dd	nn	yy
Rancho Vista	S6.45238 W76.35845			B-	dd	nn	yy
Rancho Vista	S6.45680 W76.35904			B-	dd	nn	yy
Rancho Vista	S6.45144 W76.35844			B-	dd	nn	yy
Rancho Vista	S6.45144 W76.35844			B-	dd	nn	yy
Rancho Vista	S6.46147 W76.35926			B-	dd	nn	yy
Rancho Vista	S6.45500 W76.35906			B-	dd	nn	yy
Rancho Vista	S6.45657 W76.35836			B-	dd	nn	yy
Rancho Vista	S6.45812 W76.35931			B-	dd	nn	yy

Rancho Vista	S6.45812 W76.35931			B-	dd	nn	yy
Rancho Vista	S6.45812 W76.35931			B-	dd	nn	yy
Rancho Vista	S6.45635 W76.35852			B-	dd	nn	yy
Rancho Vista	S6.45812 W76.35931			B-	dd	nn	yy
Rancho Vista	S6.45997 W76.35946			B-	dd	nn	yy
Rancho Vista	S6.45609 W76.35854			B-	dd	nn	yy
Rancho Vista	S6.45160 W76.35849			B-	dd	nn	yy
Rancho Vista	S6.45160 W76.35849			B-	dd	nn	yy
Rancho Vista	S6.45160 W76.35849			B-	dd	nn	yy
Rancho Vista	S6.45160 W76.35849			B-	dd	nn	yy
Rancho Vista		S6.45515 W76.35909	±700m	B-	dd	nn	yy
Rancho Vista		S6.45515 W76.35909	±700m	B-	dd	nn	yy
Rancho Vista		S6.45515 W76.35909	±700m	B-	dd	nn	yy
Rancho Vista		S6.45515 W76.35909	±700m	B-	dd	nn	yy
Rancho Vista		S6.45515 W76.35909	±700m	B-	dd	nn	yy
Rancho Vista		S6.45515 W76.35909	±700m	B-	dd	nn	yy
Rancho_vista		S6.45515 W76.35909	±700m	B-	dd	nn	yy
Rancho_vista		S6.45515 W76.35910	±700m	B-	dd	nn	yy
Rancho_vista		S6.45515 W76.35911	±700m	B-	dd	nn	yy
Rancho_vista		S6.45515 W76.35912	±700m	B-	dd	nn	yy
Rancho_vista		S6.45515 W76.35913	±700m	B-	dd	nn	yy
Rancho_vista		S6.45515 W76.35913	±700m	B-	dd	nn	yy
Rio Shilcayo	S6.46336 W76.35129			B-	dd	nn	yy
Rio Shilcayo	S6.45508 W76.34611			B-	dd	nn	yy
Rio Shilcayo	S6.45638 W76.34716			B-	dd	nn	yy
Rio Shilcayo	S6.45508 W76.34611			B-	dd	nn	yy
Rio Shilcayo	S6.45514 W76.34578			B-	dd	nn	yy
Rio Shilcayo	S6.45415 W76.34200			B-	dd	nn	yy
Rio Shilcayo	S6.45539 W76.34479			B-	dd	nn	yy
Rio Shilcayo	S6.45539 W76.34479			B-	dd	nn	yy
Rio Shilcayo	S6.45508 W76.34611			B-	dd	nn	yy
Rio Shilcayo	S6.45508 W76.34611			B-	dd	nn	yy
Rio Shilcayo	S6.46336 W76.35129			B-	dd	nn	yy
Rio Shilcayo		S6.462348 W76.351388	±250m	B-	dd	nn	yy
Rio Shilcayo	S6.45658 W76.34748			B-	dd	nn	yy
Rio Shilcayo	S6.45508 W76.34611			B-	dd	nn	yy
Rio Shilcayo	S6.460240 W76.350249			B-	dd	nn	yy
Rio Shilcayo	S6.46848 W76.35334			B-	dd	nn	yy
Rio Shilcayo	S6.45658 W76.34748			B-	dd	nn	yy
Rio Shilcayo	S6.45627 W76.34692			B-	dd	nn	yy
Rio Shilcayo	S6.45652 W76.34740			B-	dd	nn	yy
Rio Shilcayo	S6.45652 W76.34740			B-	dd	nn	yy
Rio Shilcayo	S6.46251 W76.35139			B-	D-	nn	yy
Rio Shilcayo	S6.46505 W76.35161			B-	dd	nn	yy
Rio Shilcayo	S6.46376 W76.35125			B-	dd	nn	yy
Rio Shilcayo	S6.45575 W76.35159			B-	dd	nn	yy

Rio Shilcayo	S6.45607 W76.34612		B-	dd	nn	yy
Rio Shilcayo	S6.44863 W76.33520		B-	dd	nn	yy
Rio Shilcayo	S6.45420 W76.33983		B-	dd	nn	yy
Rio Shilcayo	S6.45420 W76.33983		B-	dd	nn	yy
Rio Shilcayo	S6.45534 W76.34423		B-	dd	nn	yy
Rio Shilcayo, east bank	S6.46684 W76.34876		B-	dd	nn	yy
Rio Shilcayo, east bank	S6.46743 W76.35009		B-	dd	nn	yy
Rio Shilcayo, east bank	S6.46743 W76.35009		B-	dd	nn	yy
Rio Shilcayo, east bank	S6.46780 W76.35045		B-	dd	nn	yy
Rio Shilcayo, east bank	S6.47086 W76.35202		B-	dd	nn	yy
Rio Shilcayo, east bank	S6.47086 W76.35202		B-	dd	nn	yy
Rio Shilcayo, east bank	S6.47086 W76.35202		B-	dd	nn	yy
Rio_Shilcayo	S6.45746 W76.34879		B-	dd	nn	yy
Rio_Shilcayo	S6.46251 W76.35139		B-	dd	nn	yy
Rio_Shilcayo	S6.46402 W76.35125		B-	dd	nn	yy
Rio_Shilcayo	S6.45322 W76.34677		B-	dd	nn	yy
Rio_Shilcayo	S6.45652 W76.34740		B-	dd	nn	yy
Rio_Shilcayo	S6.45658 W76.34748		B-	dd	nn	yy
Rio_Shilcayo	S6.45658 W76.34748		B-	dd	nn	yy
Rio_Shilcayo	S6.45658 W76.34748		B-	dd	nn	yy
Rio_Shilcayo	S6.45658 W76.34748		B-	dd	nn	yy
Rio_Shilcayo	S6.45658 W76.34748		B-	dd	nn	yy
San Antonio de Cumbaza	S6.407227 W76.401707		B-	dd	Nn	Yy
San Roque de Cumbaza	S6.38569 W76.44278		B-	dd	nn	yy
San Roque de Cumbaza	S6.38569 W76.44278		B-	dd	nn	yy
San Roque de Cumbaza	S6.38332 W76.44091		B-	dd	nn	yy
San Roque de Cumbaza	S6.38332 W76.44091		B-	dd	nn	yy
San_Antonio	S6.40779 W76.40247		B-	dd	nn	yy
San_Antonio	S06.40318 W076.39610		B-	dd	nn	yy
San_Antonio	S6.40416 W76.39639		B-	dd	nn	yy
San_Antonio	S6.40308 W76.39574		B-	dd	nn	yy
San_Antonio	S6.40308 W76.39597		B-	dd	nn	yy
San_Antonio	S6.40440 W76.39606		B-	dd	nn	yy
San_Antonio	S6.40723 W76.40171		B-	dd	nn	yy
San_Antonio	S6.40723 W76.40171		B-	dd	nn	yy
San_Antonio	S6.40362 W76.39261		B-	dd	nn	yy
San_Antonio	S6.40362 W76.39261		B-	dd	nn	yy
San_Antonio	S6.40735 W76.40174		B-	dd	nn	yy
San_Antonio	S6.40475 W76.39789		B-	dd	nn	yy
San_Antonio	S6.40277 W76.39563		B-	dd	nn	yy
San_Antonio	S6.40420 W76.39485		B-	dd	nn	yy
San_Antonio	S6.40420 W76.39485		B-	dd	nn	yy
San_Antonio	S6.40420 W76.39485		B-	dd	nn	yy
San_Antonio	S6.40420 W76.39485		B-	dd	nn	yy
Tarapoto - Uruahasha	S6.463401 W76.334952	±400m	B-	dd	nn	yy
Tarapoto - Uruahasha	S6.463401 W76.334952	±400m	B-	dd	nn	yy

[illegible]

Tarapoto - Yurimaguas km25	S6.41247 W76.31773			B-	dd	nn	yy
Tarapoto - Yurimaguas km25	S6.41247 W76.31773			B-	dd	nn	yy
Tarapoto - Yurimaguas km25	S6.41247 W76.31773			B-	D-	nn	yy
Tarapoto - Yurimaguas km25	S6.41247 W76.31773			B-	dd	nn	yy
Tarapoto - Yurimaguas km25	S6.41244 W76.31773			B-	dd	nn	yy
Tarapoto - Yurimaguas km25	S6.41244 W76.31773			B-	dd	nn	yy
Tarapoto - Yurimaguas km25	S6.41244 W76.31773			B-	dd	nn	yy
Tarapoto - Yurimaguas km36.7	S6.42957 W76.25994			B-	dd	nn	yy
Tarapoto - Yurimaguas km36.7	S6.42899 W76.25901			B-	dd	nn	yy
Tarapoto - Yurimaguas km38	S6.42550 W76.25338			B-	dd	nn	yy
Tarapoto - Yurimaguas km42	S6.40477 W76.26779			B-	dd	nn	yy
Tarapoto - Yurimaguas km42	S6.40456 W76.26789			B-	dd	nn	yy
Tarapoto - Yurimaguas km42	S6.40477 W76.26779			B-	dd	N-	Yy
Tarapoto - Yurimaguas km42	S6.40395 W76.26796			B-	dd	N-	Yy
Tarapoto - Yurimaguas km47.5	S6.37284 W76.28488			B-	D-	N-	Y-
Tarapoto - Yurimaguas km49		S6.374566 W76.287069	±400m	B-	dd	nn	yy
Tarapoto - Yurimaguas km49		S6.374566 W76.287069	±400m	bb	D-	N-	Y-
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37027 W76.29170			B-	D-	N-	Y-
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.36826 W76.29173			B-	dd	nn	yy
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.36826 W76.29173			B-	dd	N-	Yy
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.36825 W76.29183			B-	D-	nn	yy
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37046 W76.29162			B-	dd	nn	yy
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37093 W76.29245			B-	dd	nn	yy
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37093 W76.29245			B-	D-	nn	yy
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37093 W76.29245			B-	dd	N-	Yy
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37093 W76.29245			B-	D-	N-	Yy
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37093 W76.29245			bb	D-	nn	yy
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37093 W76.29245			B-	D-	nn	yy
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37093 W76.29245			B-	D-	N-	Y-
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37093 W76.29245			B-	dd	nn	yy
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.37093 W76.29245			B-	D-	nn	yy
Tarapoto - Yurimaguas km49 (Puente Sisayacu)	S6.36826 W76.29173			bb	D-	N-	Y-
Tarapoto - Yurimaguas km49.5	S6.36634 W76.29204			B-	D-	N-	Yy
Tarapoto - Yurimaguas km49.5	S6.36634 W76.29204			B-	dd	nn	yy
Tarapoto - Yurimaguas km49.5	S6.36631 W76.29191			bb	D-	N-	Y-
Tarapoto - Yurimaguas km49.5	S6.36631 W76.29191			bb	D-	N-	Y-
Tarapoto - Yurimaguas km49.5	S6.36630 W76.29190			B-	D-	N-	Yy
Tarapoto - Yurimaguas km50	S6.36250 W76.29433			B-	dd	nn	yy
Tarapoto - Yurimaguas km50	S6.36471 W76.29292			B-	dd	Nn	yy
Tarapoto - Yurimaguas km50	S6.36249 W76.29431			B-	dd	nn	yy
Tarapoto - Yurimaguas km50	S6.36249 W76.29431			B-	dd	nn	yy
Tarapoto - Yurimaguas km50	S6.36250 W76.29422			B-	dd	N-	Yy
Tarapoto - Yurimaguas km50	S6.36250 W76.29422			B-	dd	nn	yy
Tarapoto - Yurimaguas km50	S6.36250 W76.29422			B-	dd	nn	yy
Tarapoto - Yurimaguas km50	S6.36250 W76.29422			bb	D-	Nn	Y-
Tarapoto - Yurimaguas km50	S6.36250 W76.29422			B-	dd	N-	Y-

Tarapoto - Yurimaguas km50	S6.36250 W76.29422	B-	dd	nn	yy
Tarapoto - Yurimaguas km50	S6.36250 W76.29422	bb	D-	N-	Y-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.31543 W76.28149	bb	D-	N-	Y-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32220 W76.27905	bb	D-	N-	Y-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32220 W76.27905	bb	D-	N-	Y-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32220 W76.27905	bb	D-	N-	Yy
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32693 W76.28107	B-	dd	nn	yy
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32140 W76.28282	B-	D-	N-	Y-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32140 W76.28282	B-	D-	N-	Y-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32140 W76.28282	bb	D-	N-	Y-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32140 W76.28282	bb	D-	N-	Y-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32145 W76.28166	bb	D-	N-	Y-
Tarapoto - Yurimaguas km56.8 (Puente Yurayacu)	S6.32145 W76.28166	B-	D-	N-	Y-
Tunel	S06.45551 W076.28430	B-	dd	nn	yy
Tunel	S06.45673 W076.28445	B-	dd	nn	yy
Tunel	S06.45673 W076.28445	B-	dd	nn	yy
Tunel	S06.45673 W076.28445	B-	dd	nn	yy
Tunel	S06.45673 W076.28445	B-	dd	nn	yy
Tunel	S06.45673 W076.28445	B-	dd	nn	yy
Tunel	S06.45673 W076.28445	B-	dd	nn	yy
Tunel	S06.45673 W076.28445	B-	dd	nn	yy
Tunel	S06.45551 W076.28430	B-	D-	nn	yy
Tunel	S06.45551 W076.28430	B-	dd	nn	yy
Tunel	S06.45551 W076.28430	B-	dd	nn	yy
Tunel	S06.45551 W076.28430	B-	dd	nn	yy
Tunel	S06.45673 W076.28445	B-	dd	nn	yy
Tunel	S06.45464 W076.28688	B-	dd	nn	yy
Tunel	S06.45464 W076.28688	B-	dd	nn	yy
Tunel	S06.45464 W076.28688	B-	dd	nn	yy
Tunel	S06.45464 W076.28688	B-	dd	nn	yy
Tunel	S06.45464 W076.28688	B-	dd	nn	yy
Tunel	S06.45330 W076.28609	B-	dd	nn	yy
Tunel	S6.45344 W76.28764	B-	dd	nn	yy
Tunel	S6.45289 W76.28637	B-	dd	nn	yy
Tunel	S6.45289 W76.28637	B-	dd	nn	yy
Tunel	S6.45289 W76.28637	B-	dd	nn	yy
Tunel	S6.45289 W76.28637	B-	dd	nn	yy
Tunel	S6.45289 W76.28637	B-	dd	nn	yy
Tunel	S06.45288 W076.28638	B-	dd	nn	yy
Tunel	S06.45288 W076.28638	B-	dd	nn	yy
Tunel	S06.45551 W076.28430	B-	dd	nn	yy
Tunel	S06.45551 W076.28430	B-	dd	nn	yy
Tunel	S06.45551 W076.28430	B-	dd	nn	yy
tunel	S06.45551 W076.28430	B-	dd	nn	yy
Tunel	S6.45629 W76.28430	B-	dd	nn	yy
Tunel	S6.45588 W76.28398	B-	dd	nn	yy

Tunel	S06.45330 W076.28609		B-	dd	nn	yy
Tunel	S06.45464 W076.28688		B-	dd	nn	yy
Tunel	S06.45330 W076.28609		B-	dd	nn	yy
Tunel	S06.45330 W076.28609		B-	dd	nn	yy
Tunel	S06.45330 W076.28609		B-	dd	nn	yy
Tunel	S06.45464 W076.28688		B-	dd	nn	yy
Tunel	S06.45464 W076.28688		B-	dd	nn	yy
Tunel	S06.45464 W076.28688		B-	dd	nn	yy
Tunel	S06.45464 W076.28688		B-	dd	nn	yy
Tunel	S06.45464 W076.28688		B-	dd	nn	yy
Urahuasha	S06.46234 W076.33530		B-	dd	nn	yy
Urahuasha	S06.46234 W076.33530		B-	dd	nn	yy
Urahuasha	S06.46234 W076.33530		B-	dd	nn	yy
Urahuasha	S06.46234 W076.33530		B-	dd	nn	yy
Urahuasha	S06.47588 W076.34748		B-	dd	nn	yy
Urahuasha		S06.465440 W076.335373 ±500m	B-	dd	nn	yy
Urahuasha		S06.465440 W076.335373 ±500m	B-	dd	nn	yy
Urahuasha		S06.465440 W076.335373 ±500m	B-	dd	nn	yy