ClineFit v. 2.0-β1 – beta version
User's Manual

DRAFT:

20 August 2013

Adam Porter
Department of Biology
University of Massachusetts
Amherst MA 01003

aporter@bio.umass.edu

Introduction

Geographic clines and their multi-trait equivalents, hybrid zones, are regions with relatively sharp changes in allele and genotype frequencies, and phenotypic means of quantitative traits. Several processes acting individually or in concert cause and maintain these clines, so clines are of considerable interest to evolutionary biologists. Multi-trait clines can also act as partial barriers to gene flow even to neutral traits and are therefore highly relevant to studies of how new species originate.

There are many models that address cline shape, from statistically descriptive models to models that interpret clines organically from their underlying shape-generating mechanisms, gene flow and selection. The latter are particularly valuable because their parameters include dispersal rates and selection strengths, so that in estimating cline shape, it is often possible to obtain estimates of those causal parameters. Knowledge of dispersal rates is extremely valuable in interpreting evolutionary processes affecting other traits. Moreover, armed with estimates of both dispersal and selection, it is straightforward to estimate such parameters as the extent that introgressing neutral alleles are held back by correlated background selection in the genome as a whole.

ClineFit uses likelihood fits the classical equilibrium cline models of Szymura & Barton (1986, 1991) to genotypic and phenotypic data. I originally developed it to analyze the data in Porter et al. (1997) and people have been using it on and off since then. Parts of the original code have become obsolete on most available compilers, the general code library has evolved tremendously since then, and the needs of the community have evolved as well.

This new version of ClineFit is a complete overhaul. It is now able to fit traits with inheritance patterns beyond the autosomal codominant loci in the original models, including autosomal markers, sex-linked codominant and dominant markers, and cytoplasmic markers inherited through females and through males. It also handles continuously distributed quantitative traits. This new version includes much more sophisticated hypothesis-testing methods based on model selection, quite a bit more control over how cline models are constructed, and how varying parameters are shared or not among clines for different traits.

In addition to fitting cline shapes per se, ClineFit estimates linkage disequilibria across the cline and following Szymura & Barton (1986, 1991), uses these to estimate dispersal rate. Dispersal in turn is used to estimate the strength of selection in the cline. Those together are used to estimate several additional parameters, including barrier strength to introgressing, neutral alleles, and selection on individual loci in the data set.

Classical cline models include a sigmoidal *tanh* shape describing the location and steepness of the central region of the cline. Additional modifying parameters may be added, including the mean phenotype or allele frequencies on either side of the center, and the shapes of the left and right tails of the cline away from the center. Shapes may further differ among traits or groups of traits. ClineFit permits users to specify combinations of these parameters, for example a common width but separate centers for multiple traits, and subsequently use model-selection methods to obtain the best fitting models.

Language & Platform:

ClineFit is written in C++ using the Xcode compiler and is only available for machines running Max OS X. The output files for generating the graphics can be imported into *Mathematica*, which runs on most platforms, and functions are available here to do that. The statistical package *R* could also be used to create graphics, but code for that isn't written.

Core functions:

First some terminology regarding cline orientation. A 'high-right cline' has a higher frequency or value on the right side than the left, and a 'high-left cline' is the opposite.

Classical clines:

Classical cline models use a *tanh* function to describe the shape (Szymura & Barton, older Barton, Slatkin). This function is not arbitrary, but is derived from mechanistic models relating dispersal to selection either across an ecotone or between populations that show hybrid inviability (Nagylaki), and where dispersal distances are normally distributed (i.e., a Gaussian dispersal kernel). When estimated using more than one genetic marker, these same models allow the inference of dispersal rates and selection strengths from shape information and gametic disequilibrium.

The model for a basic, 2-parameter cline shape is

$$p_x = \tfrac{1}{2}\left(1 + \tanh\left[\frac{2(x-c)}{w}\right]\right)$$

where p_x is the expected allele or haplotype frequency at location x, the parameter c is the center of the cline (i.e., $p_{x=c}$ =0.5) and the parameter w is the width, defined as w=1/*slope* at the center. A *tanh* curve ranges from –1 to 1, and the scaling ensures that p ranges between 0 on the left and 1 on the right. The cline flips from high-right to high-left when the slope is negative and w<0. (This is academic; ClineFit uses the scaling scheme below in place of a negative width). When w=0 the shape is undefined (or if you like, the slope is vertical) and a step cline is appropriate. (ClineFit doesn't implement step clines yet, though. It will instead indicate a very narrow width as an approximation.)

Marker frequencies may not necessarily be fixed at the ends of the clines, and scaling is required. This is always the case for morphological and other quantitative traits where 'fixed' has no meaning. If unfixed, it is necessary to include the asymptotic frequencies at one or both ends of the cline. The model is

$$p_x = p_L + \tfrac{1}{2}\left(p_R - p_L\right)\left(1 + \tanh\left[\frac{2(x-c)}{w}\right]\right),$$

where p_L is the asymptotic frequency on the left side of the center and p_R is that for the right side. For an allele or haplotype frequency cline, $0 \leq p_L \leq 1$ and $0 \leq p_R \leq 1$.

The cline orientation depends on the sign of $p_R - p_L$. When $w > 0$, high-right clines have $p_R - p_L > 0$ and high-left clines have $p_R - p_L < 0$. (The orientations flip again when $w < 0$ but these are redundant and omitted from consideration. That is, in these types of clines, ClineFit always treats the slope as w>0 for both cline orientations, and uses the sign of the difference between p_L and p_R to determine the orientation. This is true even when the ends are fixed; p_L and p_R are set to their fixed values on their respective sides of the cline.)

Morphological and other quantitative traits: Traditionally people use different parameter symbols for quantitative traits. The models, however, are the same. Define y_x as the phenotype at location x; it has arbitrary range:

$$y_x = y_L + \tfrac{1}{2}\left(y_R - y_L\right)\left(1 + \tanh\left[\frac{2(x-c)}{w}\right]\right).$$

Cline orientation also follows the same rules.

An important point about interpreting scaled clines:

A scaled cline is no more than an unscaled cline stretched or compressed along the trait axis. As you stretch a cline in this direction, the actual slope steepens. However, when you do, the value of the cline width parameter *w* remains unchanged, even though it is defined as *w* = 1/*slope* at the steepest point. That means that the parameter *w* ≠ 1/[*slope of the cline*], but instead w = 1/[*slope of the cline after it is standardized to the 0-1 range*].

In other words, *w = effective width*, not necessarily the measured width.

The consequence is that two clines can differ quite obviously in steepness when viewed on their respective measurement axes. However, they may very well be concordant once they are viewed on the standardized scale. Such clines will have the same effective shapes (if their tails – see below – are also congruent), and not differ significantly in statistical comparisons.

The relationships between dispersal and selection that drive cline shape apply to the effective slopes, not the actual slopes.

Tails of introgressing markers:

Concordant clines have the same centers and approximate widths. However, they collectively have somewhat steeper widths than they would if only a single trait in the population showed a cline, a consequence of the assumption that selection is multiplicative and all loci contribute. It works like this: Near the center of the zone, traits tend to be correlated within individuals because each generation, gene flow brings together relatively differentiated individuals from opposite sides of the cline. These correlations (disequilibria) create an indirect selection on all loci. These disequilibria also decay with recombination and therefore distance from the center, weakening the effect of indirect selection. Out in the tail of its cline, a given introgressed allele copy may have taken several generations to travel to there from the center, and recombination over that period will have freed it from a significant fraction of the loci it was in disequilibrium with. To that extent, indirect selection from other loci is weakened and the allele experiences mostly the selection acting directly upon it. That is, the tail reverts toward the shallower tanh cline shape it would have had were it not for the added indirect selection from other loci.

Discrete traits:

Barton & Szymura (1886,1991) account for this by approximating the shapes of the introgressing tails separately. Since a tanh function describes the conjunction of inverted exponential decays, it is roughly equivalent to use simple exponential decay functions in each tail. These functions have (on the log scale) asymptotes and slopes rather than centers and widths. Barton & Szymura (1986,1991) use the scalar θ to express the slope of the tail as a proportion of the steeper slope ($1/w$) of the central cline. It is likewise convenient to characterize the asymptotes as being shifted a distance $|z|$ from the center of the central cline. Different sides can scale independently so that separate z's and θ's are needed for each tail. The models for the tails of the right-high clines are

$$p_{x<c} = \exp\left[\frac{4\left(x-(c+z_L)\right)\sqrt{\theta_L}}{w}\right] \qquad p_{x>c} = 1-\exp\left[\frac{-4\left(x-(c-z_R)\right)\sqrt{\theta_R}}{w}\right].$$

These require $w > 0$. For left-high clines, ClineFit keeps $w > 0$ and uses the complements

$$p_{x<c} = 1-\exp\left[\frac{4\left(x-(c+z_L)\right)\sqrt{\theta_L}}{w}\right] \qquad p_{x>c} = \exp\left[\frac{-4\left(x-(c-z_R)\right)\sqrt{\theta_R}}{w}\right].$$

Cline frequencies may not be asymptotally fixed on either side of the cline, making it is necessary to include in the model the asymptotic frequencies p_L and p_R on the left and right sides of the cline. (See my earlier comments for interpreting widths in this situation.) The adjustments for right-high clines are

$$p_{x<c} = p_L + (p_R - p_L)\exp\left[\frac{4(x-(c+z_L))\sqrt{\theta_L}}{w}\right]$$

$$p_{x>c} = p_L + (p_R - p_L)\left(1-\exp\left[\frac{-4(x-(c-z_R))\sqrt{\theta_R}}{w}\right]\right),$$

and for left-high clines,

$$p_{x<c} = p_L + (p_R - p_L)\left(1-\exp\left[\frac{4(x-(c+z_L))\sqrt{\theta_L}}{w}\right]\right)$$

$$p_{x>c} = p_L + (p_R - p_L)\exp\left[\frac{-4(x-(c-z_R))\sqrt{\theta_R}}{w}\right].$$

The central regions of the cline are represented by the tailless cline models in the previous section.

So what is *θ*? In the central region of the cline, the width *w* is steep because it represents the combined effects of direct and indirect selection on each locus. *θ* removes the indirect-selection component of width, since disequilibrium is negligible in the tails and shape there depends only on direct selection. $\sqrt{\theta}$ is used in place of *θ* only because it makes the equation for estimating selection in the tails (which involves a square-root term) a little cleaner.

Morphological and other quantitative traits

As above for the central cline, the notation is different but the model is the same; simply replace *p* with *y*. Again, see my earlier comment on interpreting cline width for scaled clines.

Gametic disequilibria:

Pairwise gametic disequilibria are calculated using maximum likelihood. The likelihood estimators depend on the inheritance patterns of the genetic markers. The full details of these estimators are at the end of the manual.

In multilocus clines, disequilibria reach their peaks at the center and decay away to zero at the ends. In classical cline models, the most interesting disequilibrium value is at its peak in the cline center. The disequilibrium is maximal there and represents a dynamic equilibrium between dispersal into the center and recombination. Individuals from outside the center tend to have genetic markers more similar to the 'pure' multilocus genotypes on the respective ends of the cline. As these individuals arrive in the steeper, central portion of the cline, these multilocus genotypes are correlated within individuals: they have high linkage disequilibrium in proportion to the rate that they arrive in the center. As they interbreed and offspring contain increasingly mixed genotypes, pairwise disequilibria decay in proportion to the extent of linkage between them. Further from the center, disequilibrium is even lower, for two reasons: first, it takes more generations for a pair of linked loci to spread to the edges of the cline, giving more time for disequilibrium to decay; second, disequilibrium itself is sensitive to allele frequency so that the further the frequencies are from 50%, the less disequilibrium is possible. Thus, the spatial distribution of disequilibrium across the cline stabilizes into a monotonic curve maximized at the cline center and reaching zero at some distance from the center. The maximum is proportional to the dispersal rate of individuals across the cline center.

ClineFit estimates the shape of this monotonic curve using two key parameters describing the height (D) and center (D_c) of the disequilibrium distribution, with additional parameters that describe the shape of the curve away from the center.

Currently, the only option ClineFit offers to describe the curve away from the center is a scaled Normal distribution. The parameter D_{sd} describes the standard deviation of the spread of the distribution, in the same units as cline location along the transect.

Simulations of disequilibrium in clines using a diversity of inheritance types indicate that a normal distribution provides a reasonable approximation of this curve. However, more work needs to be done to see if other distributions might provide a better fit, particularly under fat-tailed dispersal kernels.

Recombination rates

Recombination breaks up disequilibria while dispersal builds it up, so estimating dispersal from disequilibrium requires information on the recombination rate among loci. The recombination of interest isn't that between just the loci scored in the data, rather it's the average recombination among all pairs of loci in the genome. Recombination breaks up disequilibria very slowly when loci are closely linked, so the most appropriate way to determine average recombination is to use the harmonic mean among loci. Within a chromosome, the harmonic mean depends on map distance. Since recombination rate is 1/2 for loci on different chromosomes, the harmonic mean also depends on the number of chromosomes, and will be higher if chromosomes are numerous.

r is the harmonic mean recombination rate. It is calculated in two pieces, the first for within a single chromosome, and the second for among chromosomes:

$$\frac{1}{r} = \frac{2C}{R^2} \int \ln\left[\frac{\exp(2y) - 1}{2r_0} \right] dy + 2\frac{C-1}{C}$$

This approximation requires the chromosomes be the same size, and C is the effective number of chromosomes, the number that would give the true harmonic mean recombination rate if all the chromosomes were all the same size. It requires the map length within a single chromosome, R, and r_0, the distance between adjacent loci. ClineFit assumes that $r_0 = 10^{-4}$ Morgans and the chromosome length is R = 1 Morgan.

ClineFit requires the user to set C, the effective number of chromosomes. How do you do that? Basically, you guess. If you know they are all about the same size, use the chromosome number. If they vary dramatically in size, reduce your estimate since most recombination will be within that large chromosome. Since ClineFit assumes a chromosome is 1 Morgan, an especially long chromosome may count as two.

Remember, your estimates of dispersal and all the biologically interesting parameters you get from it depend on how good your guess is about effective chromosome number. If there are few chromosomes of very different sizes, that guess might matter a lot. If there are many chromosomes, it won't matter very much. You'll probably want to do a sensitivity analysis once you're satisfied that you've found your best-fitting model. Do some repeat runs varying the effective chromosome number to see the effect on dispersal, etc.

Derived parameters:

A derived parameter is one that is calculated from the estimated parameters. Just like the estimated parameters, their likelihoods are determined by the likelihood of the shape they're associated with. Derived parameters therefore have ML estimates and support limits.

Here are the derived parameters that ClineFit estimates.

x_L, x_R

In a cline with tails, x_L is the location along the transect where the left tail meets the center portion of the cline, and x_R is the intersection point on the right. During fitting, prospective cline shapes are rejected if x_L or x_R are undefined for them, which happens when the tail doesn't intersect the central region. ClineFit determines these numerically, within 0.001%.

$p[x_L]$ and $p[x_R]$

$p[x_L]$ is the allele frequency (or continuous trait value) at point x_L, and $p[x_R]$ is that value at x_R.

$\Delta p = |p[x_R] - p[x_L]|$

Δp (deltaP in the output files) is the difference in marker frequency or continuous trait means between where the tails of a cline meet the center section. The sign doesn't change if the cline orientation is inverted. If there is only one tail, Δp is the distance from the intersection to the asymptotic frequency on the opposite side.

For example, say a cline has a tail on the left side, meeting the central section at $p[x_L]$=0.2. However, it has no tail on the right side, but is polymorphic there and reaches an asymptotic frequency of p_R=0.9. In that case, $\Delta p = |p_R - p[x_L]| = 0.7$. If the allele were fixed on the right, then Δp=0.8.

σ

$$\sigma^2 = Drw^2/(1+r)$$

(DispersalSigma in the output)

σ is the dispersal in one generation for a continuously distributed species, and is the gene flow parameter on the transect. Formally, it's the standard deviation of the distances between the birthplaces of parents and offspring and has units of distance/$\sqrt{\text{time}}$ (rather than distance *per se*; see below). It assumes dispersal occurs by a discrete-time approximation of diffusion: individuals move randomly and without regard to where they are on the transect, reproducing along the way or afterwards. σ depends width and *D*, the peak disequilibrium in the cline center, as well as *r*, the harmonic mean recombination rate.

(Why units of distance/$\sqrt{\text{time}}$? Under diffusion, variance increases linearly with time, so the standard deviation increases linearly with $\sqrt{\text{time}}$. Happily, since time is measured in generations and we are interested in dispersal in a single generation, the denominator resolves to 1 and we can overlook the weird units. We can think of dispersal as having units of distance along the transect, as our intuition would prefer.) Since the cline is assumed to have reached an equilibrium shape and spatial disequilibrium distribution, it's not strictly necessarily that individuals move following a Gaussian dispersal kernel (that they take random steps from a normal distribution), but the kernel probably shouldn't have an extremely fat tail (there shouldn't be too much of an excess of especially long steps).

*s**

$$s^* = \sqrt{k\sigma/w}$$

(*s**[hets], *s**[ecotone] and *s**[domEcotone] in the output)

*s** is the effective selection strength in the central region of the cline, and represents the combined effects of direct and indirect selection there. Its value depends on cline width, dispersal rate, and a scalar *k* that depends on the underlying fitness model and the inheritance pattern of the traits. In heterozygote-inferiority fitness models where selection acts against heterozygotes wherever they exist, *k*=2. In ecotones where fitness depends on location, *k*=3. For dominant traits in ecotones, *k*=3.175. In fitting cline models that include sets of shapes with more than one width, ClineFit uses the mean width for *w*.

Effective selection strength? Because of the pervasiveness of indirect selection acting through genome-wide disequilibrium in the center, all traits share a very similar central region shape. This can be treated as if it was a steep single-locus cline, and indeed that's what the central region of a classical cline model represents. *s** is the direct selection strength in a hypothetical single-locus cline that would produce the same equilibrium shape as we see in the center region of a multilocus cline.

β_L and β_R

$$\beta_L = \frac{\Delta p}{\exp\left[\frac{4\sqrt{\theta_L}\left(x_L - \left(c + z_L\right)\right)}{w}\right]\frac{\sqrt{\theta_L}}{w}}$$

$$\beta_R = \frac{\Delta p}{\exp\left[\frac{4\sqrt{\theta_R}\left(x_R - \left(c - z_R\right)\right)}{w}\right]\frac{\sqrt{\theta_R}}{w}}$$

(distanceBarrierL and distanceBarrierR in the output files) When more than one cline shape is specified in customized settings, β is reported for each shape separately, and as the mean β over shapes.

The β's represent the strengths of barriers to gene flow at the cline center, measured in units of distance along the transect, to an allele traveling through a multilocus cline. When loci are in disequilibrium, which is highest at the cline center, selection on one creates indirect selection on the others, steepening the clines of all. Away from the center, disequilibrium and indirect selection on the trait diminish, and the cline reverts toward the shallower, single-locus shape it would have had if there were no indirect selection. The realized cline for each locus is distorted rather than smooth, with a steepened kink in its center caused by indirect selection. Barrier strength is a measure of this distortion -- how far along the transect would you have to stretch the center of the cline in order to remove the kink and bring it back to its smooth, shallower shape? β is a linear approximation that uses the points where the tails of the clines join the central region. β_L extends the slope of the left tail from the junction at point {x_L, $p[x_L]$} to the point {$x_L + \beta_L$, $p[x_R]$}. β_R does the same, from the right side. The barriers can be different on either side if the tails have different shapes. To the extent that the tail joins the central region too far from the center ($p[x_L]$ is near 0) or the tail is especially flat, then the linear approximation becomes problematic and β is biased upward, sometimes absurdly.

T_L and T_R

$T_L = \sqrt{\beta_L/\sigma}$ and $T_R = \sqrt{\beta_R/\sigma}$

(timeBarrierL and timeBarrierR in the output files) These represent the strength of barriers to gene flow in terms of generations. *T* represents the generations it would take an allele copy on at the cusp of one tail to diffuse the distance β through the hybrid zone, expressed as a proportion of how long it would take to diffuse the same distance if direct or indirect selection didn't exist. It's calculated from dispersal rate σ and barrier strength β, and therefore relies on information from disequilibrium *D* and the slopes of the cline tails and where they join the central region. As for β, these barriers can differ on the two sides of the cline if the tail shapes differ. And as for β, if the tails are too flat or join the central region too far away from the cline center, *T* can be biased upward, sometimes to the point of absurdity.

W_h/W_p and S

$W_H/W_P = (\beta/w)^{1/r}$ and $S = -2\ln(W_H/W_P)$

(Wh/Wp and S in the output)

W_H/W_P is the ratio of the fitness of F_1 or genetically equivalent hybrids to the fitness of pure genotypes, and *S* is the corresponding selection against those hybrids. These are relevant in cases where the cline is maintained by selection against heterozygotes wherever they are, rather than by adaptation to an ecotone where different genotypes are favored in different places. β is the average of β_L and β_R, so in practice, interpreting these requires that the tails be relatively symmetrical.

s

$s_L = \theta_L s*$ and $s_R = \theta_R s*$

(sPerMarkerLocusL and sPerMarkerLocusR in the output)

s is the strength of selection experienced on a locus out in the tails of the clines, where indirect selection is negligible. In this calculation, *s* = s* vs. heterozygotes* under a model of heterozygote disadvantage, and assuming that there is an additional weak selection coefficient on each individual locus. You can make a spreadsheet column and replace it with *s* ecotone* if you believe an ecotone fitness model is more appropriate for your locus.

s is reported for each individual cline shape in fitting customized models, as well as an average over tails.

Out in the tails of the cline, indirect selection becomes negligible and the shape there depends mostly on direct selection acting on the trait itself. *s* measures that on an individual marker provided you've estimated a unique tail for it, or as some sort of average if the shape is co-estimated among markers. It is only estimable when tails are detected. Values can differ on the left and right if the tails have different shapes.

n*_*S

$$n_s = -S/\ln(1-s)$$

(NumSelectedLoci in the output files)

This represents the number of loci in the genome under selection across the cline. *s* in the denominator is the mean selection per marker locus.

It assumes that loci with major effects don't play much of a role in determining fitness in the cline, in addition to the hierarchy of assumptions needed to derive from the average of β (acting through S), and that the tails of the cline are relatively symmetrical.

Hybrid index

For the user's convenience, ClineFit produces a file of hybrid index scores with the suffix *_HI.txt*. ClineFit doesn't directly use hybrid indices in any of its calculations. The HI value in the output includes all the traits not specifically skipped in the data file. To generate hybrid indices for subsets of traits, use data files specifying which traits to skip.

An individual's hybrid index (HI) is a score representing how closely it resembles phenotypically "pure" individual of one type vs. another. Standardized, it ranges from 0 to 1.

Calculation:

For genetic markers:

HI is a simple additive genetic trait. At each locus, an allele from one side of the cline is given the score 0 and the alternate allele is 1. Each individual's allele copies are summed and divided by the total number of scored allele copies:

$$HI = \tfrac{1}{n}\sum_{i}^{loci}\sum_{a}^{alleles} x_{ia}$$

where x_{ia} is the score (0 or 1) of allele copy *a* of locus *i*, and *n* is the number of allele copies tallied across loci (i.e., the sums of the ploidies of each locus).

When dominant loci are included, their *x*'s represent recessive (0) or dominant (1) phenotypes rather than allele-copy scores. Be aware that dominant loci contribute only half as much information to HI as do codominant loci.

For quantitative traits, with a warning

Each individual's HI score must still be confined to the 0-1 range, which can be accomplished by simply scaling the trait's value to the pure phenotypes on either side of the cline. However, because environmental variance contributes to the phenotype, there is no way to determine what counts as a genetically 'pure' phenotype on either end. We need instead to establish what might be called 'purity thresholds' for each trait. The choice of thresholds is arbitrary and idiosyncratic, but the choice you make can have profound effects on the outcome.

What ClineFit does by default is choose the mean phenotype at each end of the cline as the 'purity threshold,' and then give you the option to reset it as you see fit. Any individual with a phenotype beyond the threshold is scored as having the threshold value, and any individual between the thresholds keeps its score. Then, the range is scaled to 0-1, so that each individual ends up with a score between 0 and 1 inclusive. When there is more than one quantitative trait, ClineFit takes their average.

The choice of the thresholds can have a profound effect on the hybrid index, and subsequently, the outcome of any analysis that uses hybrid indices. The closer the thresholds are to one another, the more individuals are scored as 'phenotypically pure.' Under ClineFit's default, half the individuals at each the end of the cline will be assigned 'hybrid' ancestry in that their hybrid indices won't be respectively 0 or 1. Is that sufficient or appropriate for your case?

Combining genetic and quantitative traits:

The score for each quantitative trait is included in the genotypic hybrid index sum, and treated as one observation in the calculation of *n*.

Cline orientation:

When clines are oriented so that their left side is higher, then the complementary hybrid index is reported. The result is that the hybrid index is higher on the left side, mirroring the trait values that produced them.

It's not unusual that clines of different traits have different orientations. In that case, the orientation of the hybrid index mirrors the commonest orientation. Each trait's contribution to the overall hybrid index is determined by the orientation of that trait's cline relative to the commonest one. For traits with the less common orientation, complementary trait values are used in the computation.

Maximum likelihood estimation

Maximum likelihood estimators of cline shape:

The cline models above express p_x and y_x as functions of sets of shape parameters, which we can specify slightly further as frequency p_x[…] and quantitative trait value

y_x[…], where the ellipsis represents the list of parameters of the appropriate cline model. Estimating the shape parameters involves treating p_x[…] and y_x[…] as expected values (or if you like, predicted mean values) that vary depending where you are along the transect, then choosing the appropriate probability models to describe error variation on the vertical axis around those values. Different probability models are necessary for estimating cline shapes for discrete genetic markers vs. continuously distributed quantitative traits.

With discrete markers having two alternative alleles, sampling error follows a binomial probability distribution (reducing to a Bernoulli distribution when sample size n=1 at a locality). For a diploid, codominant locus, in a population of size n individuals sampled at location x with expected allele frequency p_x (we're now dropping the […] notation of p_x[…] and just using p_x for simplicity), the probabilities of getting an observed frequency p_{obs} of A alleles for diploid and haploid loci, respectively, is

$$\Pr\left(p_{obs}\middle|p_x,n\right)=\begin{pmatrix}2n\\2np_{obs}\end{pmatrix}p_x^{2np_{obs}}\left(1-p_x\right)^{2n\left(1-p_{obs}\right)}$$

$$\Pr\left(p_{obs}\middle|p_x,n\right)=\begin{pmatrix}n\\np\end{pmatrix}p_x^{np_{obs}}\left(1-p_x\right)^{n\left(1-p_{obs}\right)}.$$

Defining **N** as a list of the sampled numbers of each allele (here, $\mathbf{N}=\{N_A,N_a\}$), the ln-likelihood model for each is

$$\ln L\left(p_x\middle|\mathbf{N}\right)=N_A\ln p_x+N_a\ln\left(1-p_x\right)+C\ .$$

The factorial terms depend entirely on the data and are absorbed into the constant C, which is irrelevant for maximizing the log-likelihood.

For continuously distributed quantitative traits, sampling error follows a normal (Gaussian) distribution instead of a binomial. At location x, the probability of getting a particular observed phenotype y_{obs} is

$$\Pr\left(y_{obs}\middle|y_x,V_P\right)=\frac{1}{\sqrt{2\pi V_P}}\exp\left[\frac{-\left(y_{obs}-y_x\right)^2}{2V_{P,x}}\right]$$

where $y_x = y_x$[…] is the expected phenotypic measure and $V_{P,x}$ is the expected phenotypic variance at location x. The corresponding *ln*-likelihood model is

$$\ln L\left(y_x,V_{P,x}\middle|y_{obs}\right)=-\ln V_{,xP}-\frac{\left(y_{obs}-y_x\right)^2}{2V_{P,x}}.$$

V_P is often a biologically uninteresting 'nuisance parameter' that must be estimated but can be difficult to interpret in the context of clinal variation. For some questions, however, these variances can get interesting. See the section on interpreting cline parameters to appreciate its limitations.

Maximum likelihood estimators of disequilibrium:

The goal is to estimate disequilibrium *D* in the cline center based on the genotypic data, but to do so we also need to estimate the nuisance parameters D_c and D_{sd} that describe the disequilibrium curve across the entire transect.

It's easier to follow the likelihood model structure if we break it into components. The overall log-likelihood model to estimate all three parameters from the data set is

$$\ln L\left(D,D_c,D_{sd}\left|\mathbf{N}\right.\right)=\sum_x \ln L\left(D_x\left|\mathbf{N}_x\right.\right)$$

where *x* is the transect location, **N** is the complete set of pairwise genotypic frequencies in the entire data set and $\mathbf{N}_x$ is the subset of pairwise genotype frequencies specifically in the population at location *x*. Disequilibrium drops off away from the cline center, so D_x is the expected disequilibrium specifically at transect location *x.* D_x depends on the three global disequilibrium parameters scaled to each population's location following normal distribution, so

$$D_x=D_x\left(D,D_c,D_{sd}\right)=D\frac{-\left(x-D_c\right)^2}{2D_{sd}^2}.$$

This scaling reflects the fact that a normal distribution is scaled so that the area under it is 1.0, which restricts the shape so that the height at the center is constrained to be inversely proportional to the spread. We are only interested in the height and spread of the curve as independent parameters and don't want to employ that constraint.

On the right-hand side of the likelihood distribution, we need to estimate the likelihoods of each population-specific D_x parameter. We want to use all pairs of loci, such that

$$\ln L\left(D_x\left|\mathbf{N}_x\right.\right)=\sum_i^{loci}\sum_{j\neq i}^{loci}\ln L\left(D_{xij}\left|\mathbf{N}_{xij}\right.\right)$$

where D_{xij} is the disequilibrium between loci *i* and *j*, and $\mathbf{N}_{xij}$ is the set of observed genotype frequencies for those loci, in population *x*.

However, the appropriate model for $\ln L\left(D_{xij}\left|\mathbf{N}_{xij}\right.\right)$ differs depending on the inheritance and dominance patterns of each locus. I've listed these functions in the appendix.

On background: Support limits vs. confidence limits

Here's a little heuristic on the relationship between confidence limits and support limits.

Maximum likelihood expresses statistical uncertainty using *support limits*, which are analogous to confidence limits in parametric statistics. The broader the support limits, the less certain you can be about the true value of your parameter, and like confidence limits, support limits are influenced by the sample size and the model you are trying to fit to your data. Here's a grossly oversimplified and superficial bit on their rationale.

Background: Parametric and likelihood-based statistical methods use the same probability equations to account for the effects of chance; the difference is in their perspectives on how best to employ those distributions to explain the world. To emphasize the differences, each has a different jargon. Support limits are part of the likelihood lexicon.

Parametric statistics addresses question: If this were the true parameter value, what is the distribution of possible outcomes from data I might collect? If μ were the true mean, how unusual would it be to collect data having a mean as different from that as $\bar{x}$, the mean I've calculated from my data? The null hypothesis is that μ is the true mean, and you decide it's probably not (you reject that null hypothesis) if the probability was too low of sampling some data set that happens to have the mean equally or more extremely different than the mean that you measured in yours (if $\bar{x}$ was outside of the 95% confidence limits of a normal distribution, for example). Once you've decided how low of a probability is too low, the confidence limits describe the range of 'common-enough' data sets the true mean μ could have produced. The cutoff for 'common enough' data sets is usually set at '>5% of the time is too unusual.'

Likelihood addresses the mirror image of that question: Of all the possible value a parameter could take, which would have the best chance of having produced the data I've already collected? Which true mean would have the best chance of having generated my data? And, what range of possible true means could have generated my data with a reasonable plausibility? That range of possible truths is described by the *support limits*, once you've decided what your cutoff for 'reasonable plausibility' is. The conventional level of 'reasonable plausibility' is to use 2 ln$L$ units to find the support limits around the maximum likelihood estimate, although you could use any number of units.

To compare support limits and confidence limits, we can start with another question from the likelihood perspective: How much more plausible is the maximum-likelihood value compared to one that's 2ln$L$ units away? Say value v is the parameter value out at either of the support-limit edges, and $v_{\max}$ is the ML estimate. With a little algebra to convert $\ln L(v) = \ln L(v_{\max}) - 2$ from the log-likelihood ln$L$ scale to the original likelihood scale L, $L(v_{\max})/L(v) = e^2 = \sim 7.39$ times better. That's true regardless of whether we use a normal probability (=likelihood) distribution or any other.

We can then ask the same question from the perspective of parametric statistics: How much more probable is the true mean value μ compared to one out at either end of the 95% confidence limits? Rephrased, what is the probability at the top of the distribution, compared to the probability at either end of the 95% ci? We'll use the standard normal distribution as an example. The probability (not the cumulative probability) at the top of the normal distribution is ~0.3989, and either 95% c.i. it is ~0.054. The mean value is therefore ~0.3989/~0.054 = ~7.39 times better. In fact, if you do this algebraically, the answer is exactly e^2 here as well.

So, in the context of the normal distribution, the span of the 2-unit support limits in likelihood analysis is identical to the span of the 95% confidence limits in the parametric analysis. The comparison isn't quite as perfect for other probability distributions, but it's still a reasonable approximation.

On background: Model selection

Model selection involves finding the underlying model that best corresponds to the data you have. It is well covered by Burnham and Anderson (2002), and Edwards (1991) provides a good introduction to likelihood estimation (although he trashes Bayesian methods in the process). Here's a brief, verbal and therefore oversimplified abstract:

Model selection is based in maximum likelihood (or Bayesian) estimation, and ultimately in information theory. The constraint is that it can only consider the options that you've thought to compare.

Central to model selection is the fact that the more parameters you include in a model (the more things you take into account when you try to explain your data), the closer you can come to explaining every single data point you have. But, all data are affected by some degree of randomness, so you can invent causes for measurable differences within your data that are really only attributable to the luck of which individuals you happened to sample. How much is too much, and how much is good enough?

Model selection adjusts the log likelihood by including a penalty factor for every parameter you include. At some point, the improvement gained by adding a parameter is small enough to be overwhelmed by the penalty for having added it. The adjusted log-likelihood score is called Akaike's Information Criterion, AIC, and

$\mathrm{AIC} = -2(\ln L - k)$

where $\ln L$ is the log likelihood and k is the number of parameters you've included in the model. So, where your best estimate of your parameters' values lies at the model's maximum log likelihood value (maximizing a negative value), you minimize the AIC score to find the best values. AIC will continue to drop as informative parameters are added to the model, but will begin to rise again as uninformative parameters are added.

However, your sample size also plays a role in determining the best parameter values, and best models, and smaller sample sizes tend to draw the maximum $\ln L$ value (and therefore the AIC value) away from the optimal parameter values. An adjustment for this sample-size bias is necessary, analogous in principle to the degrees-of-freedom adjustment (n/(n–1)) used in parametric statistics. AICc adjusts AIC for an interaction between sample size and the number of parameters,

$$AICc = AIC + \frac{2k(k+1)}{n-k-1}.$$

For the kinds of sample sizes you'll be using for cline analysis, whether you use AIC or AICc during model selection won't make any difference. But, you can't go wrong with AICc so you should use it.

On background: Complex interdependencies: an underappreciated aspect of likelihood estimation

The parameters you estimate, and therefore their values and support limits, depend on the model you use. That also means your estimates are correct only to the extent that the model you choose is appropriate. At some level that's obvious, but it hides some subtleties about deeper interdependencies among a model's variables. Even minor variations of the same general model, perhaps with an added or relaxed constraint, can have a big impact on your estimates.

Consider, for example, the relationship between a model with and without disequilibrium. Cline shape requires an estimate for the cline center *c*, and disequilibrium requires an estimate for the disequilibrium center D_c, which is where *D* reaches its maximum. We can set up a model where $c = D_c$, forcing the cline center and the peak disequilibrium to coincide (and that's sensible enough to be ClineFit's default). However, in a data set where the peak disequilibrium is actually to (say) the left of the shape's center, the maximum-likelihood estimate of *c* will be shifted left compared to a model that excluded disequilibrium. The estimated cline width will be broadened to make up for it. At the same time, the estimated breadth of the spatial distribution of disequilibrium, represented D_{sd}, will increase to accommodate the rightward shift of D_c, and the peak disequilibrium estimate *D* will be reduced. Those parameters in turn determine your estimates of dispersal and selection, so they will be affected as well. Run a model without disequilibrium, or simply allow *c* and D_c to vary independently, and the whole thing shifts back. The bottom line is that parameters that would seem entirely unrelated in these models (e.g., *D* and *w*) are all in fact interdependent and pull one another around as a result of idiosyncrasies in the data itself, and of what constraints you put on yet other parameters. Your inferences about the underlying biology are subject to these interdependencies.

How big of a problem will that really be for ClineFit users? Maybe not so big. But it's very important to appreciate how the biological model you choose, and the constraints you put on it, can have a large effect on the kinds of conclusions you draw. That kind of insight is very important in helping you decide effects of the biases and limitations on your inferences about the underlying biological processes that generated the genetic and phenotypic data you've collected.

On background: Causal vs. descriptive models

Just because a model fits your data well doesn't mean it helps you to figure out what would cause your data to have that pattern. Besides finding the best-fitting model, from a scientific perspective it's more interesting to fit a model that explains your data from a cause/effect perspective. Alternatively, you could try to fit a model derived using underlying (in this case) biological variables that interact to produce the cline shape. Fitting such a model would entail finding the values of those built-in biological variables that, in combination, best fit your data. This might not matter quite as much if you are able to experimentally manipulate cline shape, but it seems critical for understanding naturally occurring clines measured *in situ*.

In the classical *tanh* cline models, the *tanh* shape (as opposed to some other sigmoidal function) arises from the underlying conflict between dispersal and selection, once the interaction has settled to its equilibrium state. The width *w* of a single-locus cline is a simple function of dispersal rates and selection strengths, with a scaling term *k* that depends on the type of inheritance and underlying source of selection. The relationship is $w=\sqrt{2\sigma/s}$, where 2 is the scaling factor for clines with heterozygote disadvantage. Using that, we can make the algebra of the cline models more dense by replacing *w* with $w=\sqrt{2\sigma/s}$ in the model,

$$p_x=\tfrac{1}{2}\left(1+\tanh\left[\frac{2(x-c)}{\sqrt{2\sigma/s}}\right]\right)$$

However, in these particular models, dispersal and selection are confounded and enter only as their ratio — a given width can result from weak selection and weak dispersal, or strong selection and strong dispersal, but we can't tell the difference from shape alone. Absent additional information about dispersal or selection, the best we can do is estimate the ratio, so we might as well just estimate the width.

But, if we did have additional information on dispersal or selection, then we could use it to resolve the other parameter in the ratio. If there is more than one locus, disequilibrium at the center of the cline gives us information on dispersal by the relationship $\sigma^2=Drw^2/(1+r)$, where *r* is the recombination rate. This resolves to $\sigma=2\,Dr/s(1+r)$ when we set $w=\sqrt{2\sigma/s}$. We can substitute $\sigma=2\,Dr/s(1+r)$ into the cline model and do a little algebra to get

$$p_x=\tfrac{1}{2}\left(1+\tanh\left[\frac{s(x-c)}{\sqrt{Dr/(1+r)}}\right]\right).$$

Now the same cline shape previously expressed in terms of width is expressed in terms of selection strength and disequilibrium. Fitting the cline model entails finding maximum-likelihood estimates of *c*, *D* and *s* (*r* is a constant that we have to measure elsewhere). We could use those estimates and support limits to back-calculate estimates and support limits for σ and w, and any other parameters that depend on them.

This is what the models fit by ClineFit do. The only difference is that models with disequilibrium are fit using w instead of s, mainly for consistency with models that don't include D (and therefore are constrained to measuring w). It's easy to calculate any parameter from combinations of the others, so in terms of reporting cline-associated parameters, it matters little which algebraic form of the model is fit.

Parameter estimation algorithms:

ClineFit uses numerical maximization to estimate all cline shape and disequilibrium parameters. Specifically, it implements a Metropolis-Hastings simulated annealing algorithm to zero in on the best parameter combinations, then uses a modified Markov Chain Monte Carlo (MCMC) algorithm to sample the parameter space around it to estimate support limits. Both methods use a random walk through parameter space. The annealing step entails sampling within an initially broad range of parameter values, accepting new parameter combinations with higher likelihoods, then gradually narrowing that range until the global maximum is found. In order to avoid being trapped in local rather than global optima, it also accepts a fraction of combinations with lower likelihoods, in proportion to how much less likely those new combinations would be. This permits it to 'walk downhill' on the likelihood surface and escape the confines of spurious local optima.

Sampling for support limits uses the same approach, and saves only parameter combinations within 2 ln*L* of the maximum. It also records a new maximum if one is found. The algorithm records the success rate and adjusts the sampling range so that each parameter has about a 50% chance of falling within the 2-unit range. The support limits are chosen as the extremes of these saved values for each parameter separately. The user sets the number of samples to save.

Instead of a standard MCMC method, the algorithm implements a modified Gibbs sampler. With a Gibbs sampler, one full step of the random walk involves sampling each of the cline and disequilibrium parameters once, in random sequence. This is more efficient when there are many parameters to estimate at once. My modification is that a chance also exists for parameters to be sampled more than once in a cycle. This helps escape unlucky chains of increasingly low-likelihood combinations that occasionally arise within Gibbs cycles of highly parameterized models. It slows the search down when there are few parameters, but those searches go fast anyway.

Limitations to interpretation

There are many and I've included a few prominent ones here. They might be categorized as model-dependent, statistical, and data-dependent.

The equilibrium assumption: The classical cline models are derived on the assumption that the opposing forces of dispersal and selection have reached a dynamically stable state so that cline shape and disequilibria are constant. The relevant issue regarding equilibrium isn't in the estimation of cline shape parameters themselves, which simply describe the shape in the context of the model, rather it's their biological interpretation. The interpretation is therefore informed by how fast a cline shape re-equilibrates after a perturbation. These issues haven't been well explored theoretically or quantitatively and I can offer only some predictions.

In general, the slope of a cline at the center will probably recover relatively quickly because dispersal brings together the most disparate genotypes there, and realized selection on a trait is strongest there because in includes significant indirect selection via disequilibria with other traits.

However, the shape in the tails seems more likely to be sensitive to perturbations, and the effect may depend on the type of perturbation. Out in the tails, realized selection on an individual trait is weaker because disequilibrium is low there even though dispersal is (by the classical model's assumption) constant throughout. Thus, dispersal will probably play a stronger role in reestablishing equilibrium. Thus, if a perturbation were to cause a decrease in frequency in the tails, dispersal should be able to introgress the trait back to its equilibrium distribution. However, were a perturbation to cause the frequency to increase in the tails, the recovery would be driven primarily by weakened selection, and relictual frequencies may persist much longer. Following this rule, consider a set of clines that has recently moved to the right along a transect. This would leave behind a long, high tail on the left, and it may persist a long time. On the right, the moving cline would overtake the tail, creating an especially steep shape on the right side. This may tend to reach its new equilibrium shape in proportion to the dispersal rate, potentially much faster than the left tail decayed away.

In general, biological parameters such as dispersal and selection that depend on the shape at the center of a multi-trait cline are probably relatively robust to perturbation once the disruption has ended, but parameters that depend on the shapes of the tails, such as barrier strength or selection on individual traits, may be quite sensitive. Check the Derived Parameters section to remind yourself which parameters are associated with tail shape.

There is no substitute for careful thinking about the biology behind the assumptions.

Support limits on disequilibrium and parameters that depend on it: ClineFit's disequilibrium estimates are optimistically narrow because they do not take into account sampling errors in allele frequency estimation. ClineFit uses observed allele frequencies as if they were constants in the calculation of *D*. Including estimation of frequencies would blow up the analysis beyond computational practicality, requiring estimation of a frequency parameter for each locus in each sample location.

Interpreting Vp: Presently, ClineFit's only estimation model for phenotypic variance in the cline is that Vp for each trait is constant across the cline. But, the biological expectation is that Vp will increase in the center because genetic variance is higher there. It would be possible to imagine a bell-shaped function for Vp, effectively adding another parameter to the estimation, if Vp were interesting. Even that's problematic because, beyond analytical convenience, there's no reason a priori to assume Vp will be equal on opposite sides of the cline. I hope to include nonlinear Vp models in the future.

Many of the derived parameters involve the adoption of a belief system that might be beyond what many users would be willing to entertain.

Widths of scaled clines: See my comments on effective vs. measured widths of clines in the section on classical cline models. Clines of scaled parameters, particularly of quantitative traits, may appear dramatically wider or narrower than the width measurement *w* that ClineFit reports. That's because *w* is really the standardized *effective width*, representing the width of a corresponding standardized cline. What looks blatantly incongruent the trait-measurement scale may well be effectively congruent if you don't notice this distinction.

On centers and widths: Clines of individual traits might not have the same centers even if they have the same widths. If we estimate a model that constrains all clines to the same center (ClineFit's default), then agglomerated data will be spread further around that center and the estimated width will be broader. Estimate separate centers and whether or not you estimate separate widths, the estimated width will narrow. That will reduce the dispersal estimate and increase the selection estimates, if you estimate *D* in the model. The numbers probably won't change outside the support limits, but it's something to consider. See the On Background section about complex interdependencies among parameters.

Designing your sampling scheme

Cline shape estimates, and the measures of dispersal and selection that depend on them, depend a whole lot on how thoroughly the middle region of the cline is sampled. From an analytical perspective alone, I don't believe one can sample too thoroughly. From a biological perspective, though, sampling can disrupt cline shape. There are three sampling issues: the number of sites, the number of individuals per site, and the number of traits/markers.

Sites in the center: The estimate of width depends greatly on how many populations are sampled in the steepest, central region of the cline. More individuals are always better (but leave individuals out there – reducing the density too much in the center will disrupt the cline, steepening it, by increasing the disequilibrium in the next several generations while it fills back in.) Far from the center, where frequencies change relatively little, greater spacing between sites is OK.

Sampling in the tails: Parameters associated with introgression, and with selection on individual traits, are highly dependent on getting good estimates of tail shape. To determine if a significant tail exists at all (that is, whether the shape of the cline can be adequately described by a single central *tanh* model, vs. whether adding a tail produces a better fit), closely spaced sites near the base of the prospective tail are required.

On top of that, for genetic markers there is greater statistical error to overcome in estimating tail shape. Allele frequencies are closer to 0 or 1, so diagnostic alleles are rarer, and a correspondingly greater number of individuals is needed measure those differences. Many individuals in many populations are needed for each tail, particularly near its base where it meets the central-region's tanh cline.

And on top of that, we expect direct selection to differ for each trait in the tail (because indirect selection through disequilibrium is negligible there), so we expect their tails to have different shapes. We potentially need far larger sample sizes for all markers in the tails in order to distinguish the small differences between those shapes.

I don't know of any simulation or theoretical work done to help optimize sample sizes and spacing. Anyway, most studies are limited by time and effort, so people sample what they find wherever they can find it. These issues can only serve as rough guides to circumscribe one's expectations of what resolution for various parameters might be feasible from their study.

Data files:

If you've been using prior versions of ClineFit, the data file format has changed in a few places to facilitate the reading of data from diverse inheritance types. ClineFit now handles more than one transect at a time. But, if you have separate transects, you will set up the data differently, and that's explained below.

An abbreviated, representative data file looks like this:

```
Title of the data file
NPOP=5, NTRAITS=14, MAXALL=2, NGROUPS=1, ALLELELEN=1
id      sex     c_1     c_2     d_1     xc_1    xd_1    cyf_1   cym_1   phenotype_1
SKIP=1, c_2
Population from my back yard at location 0
1       F       AA      BB      0       AA      0       1       0       0.61
2       M       AB      BA      0       B       1       1       0       0.63
NEXT
Population from my friend's yard at location 6.23
etc.
NEXT
etc.
NEXT
END
```

On the 2nd line:

> **Notice that there is no white space around the '=' signs.** ClineFit can't read this properly if you include white space there. White space after the commas is OK.
>
> **NPOP** is the number of populations. ClineFit will only read this number of populations from the data. So, if you ever want to skip populations, just move them to the end and reset NPOP.
>
> **NTRAITS** is the number of traits, and now includes the 'id' and 'sex' columns. (The earlier data format didn't include these columns in the count.) Include all the traits in the data, even if you will skip some in a later command.
>
> **MAXALL** is an ignored legacy variable and you can give it any value.
>
> **NGROUPS** is for hierarchical analyses that ClineFit doesn't do. Always set NGROUPS=1.
>
> **ALLELELEN** gives the number of characters that represent a single haplotype. You should score your data so that ALLELELEN=1.

The 3rd line is for trait names. Each trait name has to be a single word in the sense that it has no white space (spaces, tabs) or commas in it. ClineFit uses those characters as name separators. Trait names can continue onto subsequent lines if that's convenient for you. Just make sure there are as many names as there are NTRAITS.

> You need to label your first trait 'id' or 'ID', which is a keyword. If you include a column for sex, then label it 'sex', another keyword. The sex-data column can be in anywhere, but the ID column needs to go first.

Of course, I have to use generic trait names in a manual. You should use names with meaning.

The 4th line gives the traits you will omit from analysis, if any. If you don't want to omit any of them, then use SKIP=0. Otherwise, list them on the same line as in the example (where the c_2 trait is skipped) and separate them by tabs or commas. You can put these on several lines if you have a lot.

The 5th line begins the population blocks. Each block has the following format:

> **The title line** gives the population's name and any info you want to associate with it. In ClineFit, the last 'word' must the population's location on the transect.
>
> ```
> Population from my back yard at location 0
> ```
>
> **Each subsequent line represents a single individual.** The first column should be the individual's ID. See below for how to represent traits.
>
> **The last line of the block is NEXT.** This is a case-sensitive keyword. Every population block ends with NEXT, even the last population.

The order of the population blocks doesn't matter. Clinefit will sort them by location when it needs to. That means you can move populations around in the data file. If you have 10 populations and want to omit one, you can move it to the end and set NPOP=9, keeping your data set intact.

The last line of the file is END, another case-sensitive keyword. It might be good to add one more blank line after this.

The data:

Separate the columns using tabs. The order of the columns doesn't matter, except for the first. It's fine to have missing data, but the gaps need to be separated by tabs.

In the representative data file above, traits 'c_1' and c_2' are codominant, 'd_1' is dominant, 'xc_1' is sex-linked codominant, 'xd_1' is sex-linked dominant, 'cyf_1' is a cytoplasmic, female-inherited marker (mtDNA or chloroplast DNA, usually), 'cym_1' is a male-inherited marker (such as a Y chromosome), and 'phenotype_1' is a continuous (quantitative) trait. ClineFit doesn't notice these prefixes – they're just for the manual – and you can give your traits any name.

The ID trait can include anything, of any length. This column always goes first.

The sex trait should be scored F, f, M or m. If you don't have sex-relevant traits, just put any sex in the column.

If females are heterogametic, do you swap the names of the sexes (i.e., score males as females and vice versa)? No. You'll tell ClineFit which sex is heterogametic when you run the analysis.

Codominant traits require two characters, which can be letters or numbers. (When you use numbers, beware of editing in Excel – see the section on Data-Formatting Hassles.) The format is case-sensitive. The order of the characters doesn't matter: AB is just as valid as BA, but Ab is a different genotype than AB.

Dominant traits require a single character, whether a letter or a number. I recommend that you always score the recessive phenotype as 0, and the dominant phenotype as something else. This will be critical if you want to estimate disequilibrium, because phenotype=0 is assumed to be the recessive trait.

Sex-linked codominant traits: treat these like codominant traits, except that females (when they are the homogametic sex) have two characters and males have only one, as in the example. Codominant loci in haplodiploid systems will go here.

Sex-linked dominant traits: score these just like dominant traits, with 0 as the recessive phenotype. Dominant loci in haplodiploid systems will go here.

Cytoplasmic female markers: give these any alphanumeric symbols. There are two categories of these markers, sorted separately. They may be present in both sexes (such as mtDNA or cpDNA), or present in only one sex (such as sex-determining chromosomes in species where females are heterogametic sex). The distinction is that they are maternally inherited.

Cytoplasmic male markers: give these any alphanumeric symbols. There are two categories of these markers, sorted separately. They may be present in both sexes (such as male-inherited mtDNA in *Mytilus* mussels), or present in only one sex (such as Y chromosomes in male-heterogametic species). The distinction is that they are paternally inherited.

Continuous traits can take any numerical value.

Threshold traits: Use any alphanumeric character. ClineFit will read these data but can't analyze them yet. Keep them in the skipped group for now.

When you have more than one transect, or are comparing sexes, life stages, etc. within a transect:

ClineFit doesn't have a simple way to handle these data yet and you need a workaround. The way to do it is to make separate columns for each marker, labeled by transect, so that populations have transect-specific markers. The populations are otherwise set up the same.

> For example, imagine you have two transects, which we'll call transect J and transect K in the manual, and you've genotyped them at two loci. Instead of labeling these (say) Loc1 and Loc2, you should code them with the transect name, perhaps JLoc1, KLoc1, JLoc2 and KLoc2. (Clinefit doesn't notice these prefixes, but they'll help you keep straight which trait is which when you set up the analysis.) (The sample locations will probably differ between

transects and that's OK. Use the same measurement scales on both axes, though.)

Then your data might look like this:

```
A data set with two transects
NPOP=5, NTRAITS=6, MAXALL=2, NGROUPS=1, ALLELELEN=1
id      sex     JLoc1   JLoc2   KLoc1   KLoc2
SKIP=0
First population from transect J at location 0
1       F       BB      AA
2       M       BB      AA
NEXT
2nd population from transect J at location 7.3
1       M       AB      BA
2       M       BB      AA
NEXT
etc.
NEXT
First population from transect K at location 0
1       F                       BB      AA
2       F                       BB      AA
NEXT
2nd population from transect K at location 10.1
1       M                       BB      AB
2       F                       AB      BB
NEXT
etc.
END
```

Look in the Hypothesis Testing section to see how to set up these comparisons.

If you needed to test whether (for example) sexes or life stages within a single transect had different cline shapes, you should code your data the same way, as if those individuals were in populations on separate transects.

What if you have sampled entirely different traits in the two transects? That's OK too. Just give the markers their standard names and you can proceed to the Hypothesis Testing section.

For example, on transect J you sampled a genetic marker, and on transect K you sampled a continuous trait. Perhaps you need to know if the continuous-trait cline on transect K has the same shape as the genetic-marker cline on transect J. Set up the data as above: populations on transect J will have missing data for the continuous trait, and those from transect K will be missing the genetic marker.

Spreadsheet formats:

These aren't supported yet.

Data-formatting hassles:

The most devious one involves hidden formatting differences for standard text files. See the second section on Excel issues.

For the most part, you can create and organize a data set on Excel, then save it as a tab-delimited text file. Then open it in a text editor and make any last tweaks to get the format right.

Spreadsheets might rescore your data: If you have any numerically scored codominant markers scored as 00, 01, 02, etc., Excel will helpfully presume those are numbers and reformat them to simply 0, 1, 2, etc. If you are creating the file in a spreadsheet, you need to code these traits alphabetically instead. However, if you are importing a numerically scored file into Excel, you need to do the conversion beforehand:

> Before I open such a data file in Excel, I open it in a text editor. I then do a global Find/Replace, changing 00-->AA, 01->AB, 10-->BA and 11->BB. This will correct the sex-linked codominant columns as well, but only for the homogametic sex. You need to fix the heterogametic sex.
>
> Then I open Excel and correct the heterogametic genotypes in the sex-linked codominant loci. For those columns only, I change the 0's to A's and the 1's to B's. For example, with sex-linked codominant data in column D, in a new temporary column use the formula
>
> =IF(OR(D6="",ISBLANK(D6)),"",IF(D6=0,"A",IF(D6=1,"B",D6))).
>
> This copies the homogametic genotypes into that column unaltered. Then, select the column's contents, Copy, and use "Paste Special: Values" to overwrite the original column D. Finally, delete the temporary column with the equation.

Insidiously cryptic file formats: Second, when you save the file in text-only format (.txt), Excel often evilly formats the file with the wrong end-of-line character. Your old data files might have been formatted this way as well, as it is Apple's legacy format. This character is invisible so it's not obvious when it happens. ClineFit identifies the problem but you have to fix it yourself. There are a few ways to do it.

> One is to find a copy of BBEdit, a text-based html editor; there are free versions around. You can open your data in BBEdit, and depending on the version, there will be a little menu someplace (at the bottom of the page in my version) that lets you choose the file-encoding format. Choose 'Unix (LF)', which puts the right end-of-line character in, and save.
>
> Another is to find TextWrangler or some other free text editor that lets you see normally invisible characters like tabs, ends-of-lines, etc. Find the menu option that lets you view these characters. Copy the symbol for the last character of a line, then open Find/Replace. Paste the end-of-line character into the find box, put \n (backslash-n) into the replace box, do a replace-all, and save.

If you open and edit the file again in Excel, you'll probably have to repeat this process.

Data formatting details: ClineFit is very literal when it reads data files. People can hit curbs and roadblocks when their data formats don't adhere strictly to the example format above. Although I've tried to anticipate some of the problems and give appropriate error messages, it's perfectly typical that you'd end up with an error message that is a complete non sequitur. (You can find more on error messages in the Output section of this manual.) Your first hypothesis in this event is that there's a formatting inconsistency someplace.

Legacy versions: ClineFit doesn't pool or rescore alleles anymore:

The prior version required you to attach a code to each locus name specifying which allele(s) were at highest frequency on the right side of the cline. This code is no longer necessary but you can leave it in if you're using an older data set; it will be ignored.

Why? ClineFit now handles clines of either orientation, so rescoring the data for a right-high orientation is superfluous. ClineFit instead guesses at the cline orientations using your data, and gives you an opportunity to override it.

> *An example of what's different:* in the older format, imagine a codominant trait named Loc1, having 2 alleles, A & B. ClineFit required you to append a code to the locus name Loc1(B) to indicate B was the most common allele on the right side. If you had 3 alleles, A, B & C, then the code would also indicate which alleles to pool. The code Loc1(B;C) would recode your data to pool the B & C alleles as most common the right side, whereas the code Loc1(B) would pool the A and C alleles on the left.
>
> Now you can just ignore all that and name your marker Loc1. That also means that if you have multiple alleles, you now have to pool them by hand to create 2-allele loci when you create in the data file.

Running the program:

Preliminaries:

A convention: In the examples below, I'll use a '>' character to represent the beginning of an input line, even though that symbol probably won't be present on your screen. You should never type the symbol if you're following along.

A convenience: Throughout, when given a choice of options, you can usually use the first letter of the option instead of typing the whole thing. For example, given:

```
[Y/N/default/done]
```

You can always use 'd' or 'D' for 'done.'

Sample data file: Examples of output will correspond to the file <SampleClineData1.txt> with the random number seed set to 1. That file was produced by simulation and has a simple sigmoid cline shape, so estimation of the tails of the distribution won't yield a significantly better fit.

Where to put your data file

Put your data file in the 'User area.'

When you log onto a Mac, you actually log into a 'User' area that has the same name as your login screen, and when you open a new window, it opens into this 'User'-level folder. From there, you would typically navigate down through folders to find your favorite file.

ClineFit requires that you put your data file into your User folder. The files that ClineFit creates will be placed in this User folder as well. This rule only applies to input/output files; it doesn't matter what folder you keep the ClineFit program in.

Rename your output files, and move them

Clinefit will overwrite any output file, present the User folder, that has the same name. Every time I do a run, I at least move the output files into a different folder with an informative name. I usually make a copy of the input file and store it in that new folder as well, and take complete notes on the runs I do.

Starting:

Just so you know it: If you type in something inappropriate before the estimation process starts, ClineFit will normally ignore it or give you feedback, but sometimes, for example if you accidentally paste in something with multiple lines, it will go berserk and start spewing nonsensical text. Quit and start over (and check for data-file errors if it happens again at the same place).

You'll be led through a series of setup options before your analysis runs. Here are the key steps:

The first screen:

```
    What is the name of the data file you would like to read?  The file must be
a text (.txt) file. The maximum length of the name, including the extension, is
32 characters.
? data file name =
```

Enter the input file name. You can paste it in if that's convenient.

```
>SampleDataFile1.txt
```

ClineFit then tries to open and read the file, and interpret the data. If you have input-file errors, this is where many of the problems show up. Others might wait until later.

The second screen:

```
From the arrangement of the data and some guesswork, traits have been
preliminarily sorted.  In this step, you have a chance to edit these
categories.
Do you prefer to display categories by trait name, or by trait number
(following the order in the data file)?
Edit by ?=   [name, number, both]
```

ClineFit will try to figure out the inheritance patterns of your traits based on how they are scored. Here it asks you how you want to display its efforts, whether by name or number.

```
>both
```

The numbers will correspond to the original data columns, before the skipped data are accounted for. So, for example, in the sample data file, the trait listed 4th, c_2, is skipped. Even so, trait d_1 will still be called trait number 5.

The third screen:

```
 0. ID and unclassified traits:
    1(id)
 1. Sex identification trait:
    2(sex)
 2. Haploid inheritance (nuclear loci):
    none
 3. Codominant inheritance:
    3(c_1)
 4. Dominant inheritance:
    5(d_1), 7(xd_1), 8(cyf_1)
 5. Codominant sex-linked or haplodiploid inheritance:
    6(xc_1)
 6. Dominant sex-linked or haplodiploid inheritance:
    none
 7. Cytoplasmic female-transmitted inheritance (but found in both sexes):
    none
 8. Cytoplasmic male-transmitted inheritance (but found in both sexes):
    none
 9. Cytoplasmic female-transmitted inheritance (found in females only):
    none
10. Cytoplasmic male-transmitted inheritance (found in males only):
    9(cym_1)
11. Quantitative (continuous) traits
    phenotype_1, phenotype_2
12. Quantitative (continuous) threshold traits
    none
Location to sort a trait into =?  [0-12, done]
```

This presents ClineFit's preliminary attempt to sort the traits and gives you the opportunity to change them. It will assume that any trait having both sexes represented by one character is an autosomal dominant trait, even though it might really be cytoplasmic or sex-linked dominant. (The exception is that if you code a trait 'mtDNA,' 'cpDNA,' 'COI' or 'COII,' ClineFit will sort these into the category 7.)

To fix that, choose the inheritance type that the trait really is. The next screen will ask you which traits should be sorted into this inheritance type. For example, to move the sex-linked dominant marker, enter

```
>6
```

```
You can type in the names or the ordinal numbers of these traits. Type in names
exactly as they appear in the data file. If entering more than one name, separate
them with commas or white space (e.g., "trait1,trait3").  If entering numbers,
separate them with commas or white space too; you can enter a sequence of numbers
using a hyphen with no white space (e.g., "1-3,5,8").  Don't mix trait names and
numbers.

Sorting traits into:  (6) Dominant sex-linked or haplodiploid inheritance

       ? trait names or numbers =
```

Enter the name or the number, but not both. (I usually use numbers.) That is, use

```
>xd_1
```

or

```
>7
```

but not

```
>7(xd_1)
```

Continue until you're satisfied.

Fourth screen:

```
You need to specify the heterogametic sex.

  heterogametic sex =?  [M or F]
```

This is necessary if you have sex-associated markers, but otherwise just enter something.

4.5th screen

ClineFit generates a hybrid index file using all the markers. If you have continuous traits, they contribute to the hybrid index. You have a chance here to modify how continuous traits are scored. Read the Hybrid Index section to see some of the implications of your choices.

```
The hybrid index ranges between 0 and 1, and estimates the extent that an individual has
a recombinant genotype.  The index is calculated by scaling an individual's phenotype to
its relative position within the span of the phenotypic scores assigned to 'pure'
populations.  For genetic markers, the minimal phenotypic score is 0 (absent) and the
maximum is 1; heterozygotes fall in between.
  For continuous traits, where even 'pure' genotypes have variable phenotypes, these
boundaries must be set manually.  By default, the limits are set to the means of the
populations at the ends of the transects.  However, unless unreasonably extreme
boundaries are chosen, individuals may have phenotypes outside the set boundaries.  In
that case, phenotypes above the maximum are scored as having the maximum value, and those
below the minimum are given the minimum value.
You can adjust those limits here.
1) phenotype_1:
  left : 2.93111      (default)
  right: 7.1905       (default)
2) phenotype_2:
  left : 9.75944      (default)
  right: 3.815 (default)

Choose the trait you would like to reset.  [1-2, done]
```

If you want a hybrid index file without all the traits (say, only genetic markers), then do a new run and skip the traits you don't want. The hybrid index file is generated before parameter estimation starts, so you can quit on the next screen.

Fifth screen:

```
The program's current settings are:
     1 Input file name: SampleClineData1.txt
     2 Base output file name: SampleClineData1Out
     3 Set orientations of clines:  as surmised from data
     4 Estimation model: none
     5 Turn off/on estimation of some parameters:  select estimation model first
     6 Turn off/on co-estimation of parameters among traits:  select estimation
model first
     7 Starting parameter values to facilitate convergence:  select estimation
model first
     8 Estimating the model only, or also sampling for support limits:  finding
the best parameter estimates only

Choose the setting you want change.
? (1-9/done)
```

Trait numbering scheme changes: From this point on, ClineFit numbers traits according not to the sequence in the original locus list as on the 2nd screen, but in the sequence of the data being analyzed, and after dropping the ID, sex and skipped traits.

For example, c_1 becomes trait 1, d_1 becomes trait 2, etc.

This screen sets up the actual analyses and confirms your input and output file names. They are:

```
1   Input file name: SampleClineData1.txt
```

You can change your data set here.

```
2   Base output file name: SampleClineData1Out
```

ClineFit creates several data files (see the Output Files section) and this will form the first part of those file names. ClineFit will append suffixes for each of the file types. You can change that here.

```
3   Set orientations of clines:  as surmised from data
```

ClineFit guesses at your cline orientations by comparing the marker frequencies (for genetic markers) or trait means (for continuous traits) from the first and last scored populations. That guess could be wrong if the end populations have anomalous patterns.

Choosing `>3`, you get this screen:

```
Clines may be oriented so that the right side is higher the left, or vice
versa.  Here you can specify the orientations.
Traits with right side higher:
1,2,3,4,5,6,7
Clines with left side higher:
8
Type in the list of traits for which you would like to switch orientations.
Separate numbers by commas, but no white space. For example, if you have 5
```

```
traits in your data, you might enter '2,4,5'.  If you enter '2, 4, 5', it will
not be understood properly.
? trait numbers (#,#,#/done) =
```

This toggles the orientations: entering

```
>1,2
```

would (erroneously) put those traits into the left-side-high group.

4 Estimation model: none

You have to set this before you can set options in (5), (6) and (7).

Go back and read the model section to see what the parameters are in detail. But quickly, c=center and w=width of the central sigmoid cline, tL and zL determine the left tail shape, tR and zR determine the right tail shape, and pL and pR are the asymptotic allele frequencies (for genetic markers) or phenotypic means (for continuous traits) at respectively the left and right ends of the cline. D is disequilibrium at the center. If you have continuous traits, each will include a phenotypic variance parameter Vp, even though that parameter won't show up in the lists.

The choices of models you get here will depend on the inheritance types in your data set. For example, if you include any continuous traits, you only get models that include pL and pR among the parameters. Or, if you had less than 2 genetic markers, your options wouldn't include estimation of D.

Some of the model types never show up; for example, there is no model 1. That's because I hope to include additional models but I haven't coded them yet.

```
You may fit any of several cline models to your data.  Here you choose the base model
type. If different traits will be fit to different models, then choose the model having
the all the parameters you will estimate. Later, you'll have options to specify fixed
values or to turn on/off estimation for some of these parameters, whether for the whole
model or just for particular traits.
Your data include quantitative traits, so regardless of the model you choose here, a
phenotypic variance model will be included for those traits implicitly. By default,
variances are assumed to be constant for the entire cline. You can change that later.

  Your choices of available models are:
      5. simple sigmoid cline: c, w, pL & pR -- ends fitted or specified
      6. simple sigmoid cline: c, w, pL, pR & D -- ends fitted or specified, with
disequilibrium
     10. tailed step cline: c, tL, tR, zL, zR, pL & pR -- ends fitted or specified
     11. tailed sigmoid cline: c, w, tL, tR, zL, zR, pL & pR -- ends fitted or specified
     12. tailed sigmoid cline: c, w, tL, tR, zL, zR, pL & pR & D -- ends fitted or
specified, with disequilibrium

? base model =
```

You should choose the model with the maximum range of parameter types that you will want to estimate among the traits. For example, if you want to estimate a left tail for one trait but simple shapes for the rest, pick option 10. You will be able to customize and restrict the model in option (5) after you pick your global model here.

Since the file <SampleClineData1.txt> includes continuous traits, you only get model options that include the ends of the cline, pL and pR. If you skip these traits, you can see the other options.

For the sake of demonstration, let's imagine that you picked model 6, which includes a disequilibrium parameter:

```
>6
The program's current settings are:
      1 Input file name: SampleClineData1.txt
      2 Base output file name: SampleClineData1Out
      3 Set orientations of clines:  as surmised from data
      4 Estimation model:     6. simple sigmoid cline: c, w, pL, pR & D -- ends fitted, 
with disequilibrium
        4a     Effective number of chromosomes: 24
      5 Turn off/on estimation of some parameters:  default
      6 Turn off/on co-estimation of parameters among traits:  default
      7 Starting parameter values to facilitate convergence:  default
      8 Estimating the model only, or also sampling for support limits:  finding the best 
parameter estimates only

Choose the setting you want change.
? (1-8/done)
```

Whenever you choose a model with disequilibrium, you'll get a new option, (4a).

```
>4a
      Several parameters can be estimated from cline shape and disequilibrium provided 
that the genomic harmonic mean recombination rate is known.  This can be calculated from 
the effective number of chromosomes.  The effective number is roughly the number of 
chromosomes there would be if they were all the same length.
      Enter your best estimate of the effective number of chromosomes
```

The new option (4a), the effective number of chromosomes, represents a parameter that is required for estimating biologically interesting values such as dispersal rate and selection strength from the shape and disequilibrium. You can think of this as a rough estimate of the number of chromosomes there would be if all the chromosomes were the same size. If you don't know the number of chromosomes, then you will weaken your confidence in biologically interesting values. The idea is that dispersal builds up disequilibrium and genome-wide recombination reduces it, and the more chromosomes there are, the higher the genome-wide recombination rate. If you know D and the effective recombination rate, then you can estimate dispersal.

The number you choose can end up mattering a lot. Read the Recombination Rate section to get a sense of that. You can try different values in new runs, or twiddle with them on the output spreadsheet, and get a sense of the sensitivity of the estimates to chromosome number.

For no reason except that my organisms tend to have numerous chromosomes, the default is 24. Here we'll change the effective number to 5.

```
>4a
>5
>done
```

5 Turn off/on estimation of some parameters: default
6 Turn off/on co-estimation of parameters among traits: default

These two options are essential to customizing your cline model. They are covered in the sections on Model Customization and Hypothesis Testing.

If you're working through this using <SampleDataFile1.txt>, skip options 5 & 6 for now.

7 Starting parameter values to facilitate convergence: default

ClineFit assumes that the steep part of your cline is somewhere near the center of the transect. If it's not, you might want to move the initial center closer to its proper location to make the fitting go faster. It's probably not necessary to modify the others, but you can.

8 Estimating the model only, or also sampling for support limits: finding the best parameter estimates only

ClineFit proceeds in two stages. The first is to generate a maximum-likelihood estimate of the best-fitting parameter values for the model you've chosen. The second is to determine the extent of error around those estimates — the support limits — and therefore the uncertainty in cline shape. Sampling for support can take a long time, especially for parameter-rich models.

> If you are in a model-testing stage of analysis where you are trying to determine which cline model gives the best fit, or if you are testing competing hypotheses about cline shapes, then you probably aren't interested in determining the support limits. Use the default setting.
>
> If you've already determined your best-fitting model(s), then you do want to sample for support. This step will also give you the information you would need plot an error distribution around your cline.

Choosing (8) will toggle between turning on or off the support-limit step, and immediately pop you back to screen 5.

However, to demonstrate, we'll turn on sampling for support:

```
>8
The program's current settings are:
     1 Input file name: SampleClineData1.txt
     2 Base output file name: SampleClineData1Out
     3 Set orientations of clines:  as surmised from data
     4 Estimation model:     6. simple sigmoid cline: c, w, pL, pR & D -- ends fitted,
with disequilibrium
       4a     Effective number of chromosomes: 24
     5 Turn off/on estimation of some parameters:  default
     6 Turn off/on co-estimation of parameters among traits:  default
     7 Starting parameter values to facilitate convergence:  default
     8 Estimating the model only, or also sampling for support limits:  finding the best
parameter estimates only

Choose the setting you want change.
? (1-8/done)

>8
The program's current settings are:
     1 Input file name: SampleClineData1.txt
     2 Base output file name: SampleClineData1Out
     3 Set orientations of clines:  as surmised from data
     4 Estimation model:     6. simple sigmoid cline: c, w, pL, pR & D -- ends fitted,
with disequilibrium
       4a     Effective number of chromosomes: 24
     5 Turn off/on estimation of some parameters:  default
     6 Turn off/on co-estimation of parameters among traits:  default
     7 Starting parameter values to facilitate convergence:  default
     8 Estimating the model only, or also sampling for support limits:  fitting the model
and sampling for support limits (error bounds)

Choose the setting you want change.
? (1-8/done)


>done
```

Sixth screen:

Here you can adjust some settings that determine how quickly and completely ClineFit finds the best estimate, and how reliable your support limits will be.

```
     The numerical estimation algorithm will first try to narrow down to a best estimate
of cline shape parameters, then it will sample the parameter space to determine the
support limits.  You can use the default settings for this or override them.  In general,
higher values increase the time that the program runs, but also give more reliable
estimates.  The more parameters being estimated in the model, the higher these numbers
should be.  But for quick, qualitative looks at cline shape while you get the feel of
your data, low settings may be all you need.
     The current settings are:

     Narrowing for initial best estimate (the 'annealing' phase):
       1      26108   random number seed (chosen for you by default)
       2      5800    parameter tries per annealing step (at least 50/parameter; 200 is
better)
     Sampling for support limits:
       3      2000    replicates saved in garnering support (at least 2000)
       4      145     replicates between saves (at least 5/parameter; 100 is far better)
     The last helps ensure that successive saved replicates will be roughly independent.
More replicates yields better estimates of support limits but there's a tradeoff with
running time.

     Enter the parameter number you would like to adjust.  Enter 'done' when satisfied;
enter 'default' to restore the default settings.
? Setting to adjust (1/2/3/4/default/done)?
```

Your adjustments entail tradeoffs. The lower you set the values, the sooner the program finishes, but the less accurate the results could be. I've suggested some values but my suggestions are really only semi-educated guesses. ***I do not guarantee the default settings will give you the best answers!*** In other words, you are just as responsible for choosing the default settings as you are for changing them.

Going through the settings,

```
1    26108   random number seed (chosen for you by default)
```

Random numbers on computers are actually 'pseudorandom;' they are mathematical functions that yield numbers that are evenly distributed over the range of 0-1. The 'seed' is the starting point, and different starting points give different sequences. If you put in the same seed, you get exactly the same "random" number sequence.

ClineFit chooses its random number seed from the computer's clock, so it will be different on each of your runs. However, if for some reason you wanted truly identical runs (maybe to extend the sampling range for support limits?), you could set the same seed in each. I actually never reset this in practice, but it's useful if you are following along and want to repeat my sample analyses.

If you're following along, reset it:

```
>1
**************************
     Choose an integer between 1 and 32666 to seed the random number generator.  Using
the same seed twice gives identical results, useful if you want repeatability.  If you
enter 0, a unique seed will be chosen for you and recorded in the output file.

? seed =

>1
```

```
2   5400   parameter tries per annealing step (at least
50/parameter; 200 is better)
```

ClineFit sets the default value to 50/parameter.

You should read about how MCMC estimation works in another section of the manual. To summarize, parameters are randomly adjusted between iterations and the likelihood is calculated for each. Better ones are saved. At the beginning of the run, the random values vary over a wide range. Then, in successive 'annealing' steps, the range is narrowed. The rationale is that the maximum might be a long way from the starting point or there might be multiple local maxima, and sampling widely at the beginning is good. But once you begin to get into range of the best estimate, it's better to sample more narrowly and hone in on the best values.

This option lets you determine the rate that you hone in. If you set the number too low, then you may well focus in too hastily and miss the best estimate altogether. But if you set it too high, then you will waste time sampling widely over a region that already probably contains your best estimate.

> Many times you'll run models that are dramatically worse than the best one (based on their AICc scores). If you're comparing a lot of models, then you might consider reducing this value a little to weed out models are especially implausible. Then, when you think you have some models that might be reasonably close, say very conservatively within 15 AICc units, you could re-run them with higher settings to make sure you're getting the best comparison. It's a tradeoff between efficiency and accuracy, and you have to be careful. When in doubt, keep it towards the higher end.

```
3   2000   replicates saved in garnering support (at least 2000)
```

You get this option when you estimate support limits. Essentially, the more replicates you save, the better your estimate of the support limits will be. In another section of the manual, I'll give advice on how to decide whether you need to go back and increase your sample size.

Because this is a manual and all we need is to see how the program works and what the output looks like, we'll set this to 100 and reduce the wait for the program to finish.

4   135   replicates between saves (at least 5/parameter; 100 is far better)

If you are sampling for support and were to save every iteration that fell within the 2-unit support limits, then successive saves would tend to have autocorrelated parameter values. You actually want each saved iteration to be independent because each replicate should be a random sample of parameter values around the optimum. The more replicates you require between saved iterations, the more representatively random each save will be. I've actually just picked a minimum of 5/parameter out of thin air, and 100 might be excessive. Check the Analyzing your Analysis section to help figure out how many is good enough.

However, this value has a large effect on how long your error estimation will take. It will take twice as long to do 100 replicates as to do 50, and a parameter-rich run will take a long time even with a low replicate number. It's a tradeoff between the precision you want on your error bounds vs. your how tight your deadlines are.

Since we're just demonstrating how the program works, we'll set this to 10.

The settings overall are

```
     Narrowing for initial best estimate (the 'annealing' phase):
       1      1      random number seed (chosen for you by default)
       2      5800   parameter tries per annealing step (at least 50/parameter; 200 is
better)
     Sampling for support limits:
       3      100    replicates saved in garnering support (at least 2000)
       4      10     replicates between saves (at least 5/parameter; 100 is far better)
     The last helps ensure that successive saved replicates will be roughly independent.
More replicates yields better estimates of support limits but there's a tradeoff with
running time.

     Enter the parameter number you would like to adjust.  Enter 'done' when satisfied;
enter 'default' to restore the default settings.
? Setting to adjust (1/2/3/4/default/done)?
```

When you're done with this screen, the analysis starts.

Output while running:

Each time ClineFit finds a new best-fitting parameter combination, it will post that to the screen. This will include log-likelihood and AIC scores, as well as any derived parameters (parameters that are calculated directly from the primary parameters you've chosen for your model). You can watch it converge. You can also watch to see if it starts producing non-intuitive estimates, such as the center walking towards one edge of the transect while the width increases. If so, something's possibly funky with the input format, or a cline orientation is set wrong.

The first entry of the first line of each best estimate will be the iteration number. The end of the line will say ANNEALING FOR BEST VALUES. If you've chosen to calculate support limits, the message will switch to SAMPLING FOR SUPPORT when that process starts.

If you are sampling for support and encounter a new best estimate, the first line will end with the number of saved parameters out of the total you are saving.

When everything's done, you'll see the best estimates for each parameter. These will also be in the output files so you don't need to copy them down.

Parameter naming conventions

In setting up a customized run with more than one trait, you may have chosen an option to estimate a parameter independently for separate traits. For those cases, the traits that a parameter applies to will be given in brackets. Any parameters that apply to all traits won't have brackets.

> For example, say you're fitting a simple sigmoid cline and have two continuous traits in your data; you therefore must estimate their asymptotic mean trait values at each end of the cline. Otherwise, in your model the traits share the same widths and centers. The parameters will be named c, w, pL[1], pL[2], pR[1], pR[2], Vp[1] and Vp[2]. (The latter are phenotypic variances that must be estimated separately for all quantitative traits.) If you had opted to estimate their widths independently, then you would get w[1] and w[2] in place of w. If you had 3 traits, and customized your model so the last trait's width was independent, then the parameter names would be w[1,2] and w[3].

These conventions are followed for output on screen and in the output files.

If you quit the program before it's done running…

You won't get any partial output. You have to start over.

Output files generated:

ClineFit produces several output files. Let's say your input file is <**myPreciousData.txt**>. The output files will be named:

myPreciousData_sum.txt

This includes the summary statistics of the run, especially the maximum-likelihood parameter estimates and their support limits (if you've calculated them). The file is tab-delimited so you can open it in a spreadsheet program if you want.

myPreciousDataOut.txt

This is a tab-delimited spreadsheet produced when support limits are calculated. It includes all the saved replicates that fall within the 2-unit support limits, sorted by likelihood score with columns for likelihood statistics and parameter values, including values for derived parameters. Each replicate is a row, and the first row is the maximum-likelihood estimate.

myPreciousData_HI.txt

This includes the hybrid index values for each individual, as well as their location on the cline and the number of characteristics used to calculate the index. It is actually produced early in the run so you could stop the run early if all you wanted was HI scores. See the Hybrid Index section for how these are calculated.

myPreciousData_freqs.txt

These include allele frequencies and sample sizes for each genetic marker. They are generated soon after the data are read in, so you can stop the run early if all you want are these.

myPreciousData_pool.txt

This is ClineFit's interpretation of the input file, minus the skipped loci. Inspect it and see if it looks right. If not, try to find the error in your original file.

myPreciousData_shape1.txt
myPreciousData_shape2.txt
etc.

Use these for making your graphics. Each file includes shape parameter values for a unique cline shape in the analysis, with a list of the support-limit shapes. The best-fitting shape is in the first line. For custom analyses, one shape file is produced for each unique combination of parameters, and the parameters it includes are in the first line of the file.

> For example, say your customized model specifies unique widths for traits 1-3 vs. traits 4-6, with a common center for all of them. You will get two shape files. If you instead specified unique widths for traits 1-3 vs 4-6, but specified common centers for traits 1-4 vs 5 & 6, you would get three shape files. The first would be for traits 1-3, the second would be for trait 4, and the last would include traits 5 & 6.

Error messages

This is a beta version of ClineFit and it will inevitably produce errors under conditions that I haven't run. Other users and I depend on your input to help identify them.

ClineFit tries to recognize common errors and print those to the screen, sometimes with a suggestion on how to resolve the problem.

Sometimes you can get an inscrutable error message, though. Sometimes these are caused by data-formatting errors.

One common error message involves internal data arrays, where the size of the array is inconsistent with information used to access it:

```
Bad index:  sought item 3 in a 0-indexed ObjList of length 0
```

> These errors often trace to such things as inconsistencies in the data file, for example, specifying NTRAITS=10 when there are actually 9 data columns, using a 2-word trait name, or forgetting to use NEXT to end a population block, etc. Check carefully for such issues.

In some cases error messages may arise because your data have features that I hadn't anticipated, for example unique combinations of missing data for different traits. In those cases, I'll probably want to run your data set through my debugger to see what happened.

Analyzing your analysis

How do you know if your run really gave you trustworthy values? It depends on how you define trustworthy. Numerical methods like MCMC converge on the best answers, but they don't find the 'truth.' Moreover, for inscrutable reasons, occasionally they converge on a pretty wrong answer. (I've tried to design ClineFit so that it avoids these misdirections provided you've allowed enough replicates during the annealing process, but believing that it avoids all of them is tantamount to accepting a null hypothesis.) There are a few things you can do:

Rerun the analysis a few times with the same settings

Change the random number seed (which will be unique for each run anyway). Different runs should reach very similar answers. If they don't, then increase the number of replicates per annealing step.

Keep an eye on the progress of the best-estimate output on the screen during the run. If you are in the sampling-for-support stage and still get a lot of new best estimates, then you probably converged too fast in the annealing stage. (A few new ones are almost inevitable.) Likewise, if you get a run of new best estimates near the end of the annealing stage and especially if they seem to be crawling in one direction, you may have a problem. Start over, and increase the number of replicates per annealing step.

Plot your parameter space:

Do a run using the 'sample for support' option. When the run is done, the <**myPreciousDataOut.txt**> file (see Output Files section) will include an lnL column and a column for each estimated parameter. Plot lnL or AICc vs. your parameter. You should see a curve with a nice, smooth surface (filled in if more than 2 parameters are estimated in the model) that extends all the way down to the 2-unit bottom of the graph on both sides, or if you use AICc, all the way up to the edges. If one side doesn't extend all the way down, or if the coverage seems spotty especially near the maximum, or the curve seems too asymmetrical (indicating that it might still be walking towards the maximum – but remember that likelihood clouds can often be asymmetrical), then re-run the analysis. Increase the number of replicates per annealing step if the maximum seems awry, and increase the number of saved replicates if the ends of the curve aren't sufficiently filled in. Plot this for each estimated parameter.

If you have a slightly bad feeling and don't even know why (and you're not usually a worry-wort), just trust your intuition and re-run it.

Hypothesis testing

ClineFit fits models of cline shape to data. The models comprise the numbers and identities of the parameters combinations that you assign. Your hypotheses involve questions about whether and how cline shapes in different traits differ wholly or in part. To test hypotheses, you use 'model selection' methods that boil down to comparing how well alternative models fit your data.

ClineFit gives you full control over which types of parameters you include in the models you are concerned with, and which traits are allowed to differ within a given parameter type. Each run produces an AICc score, and to oversimplify a bit, the best model has the lowest score. Models more than 2 AICc units apart are significantly different from one another. Another section goes into a bit more depth on the rationale behind AICc.

Hypothesis testing entails creating models that are consistent with your alternative hypotheses and seeing which alternative gives you the best fit.

> For example, perhaps you have three markers and need to find out if the cline for the third marker has a width the same as, vs. different from, the other two, even though you believe they share the same centers. To test that, you would first run a model with a single width parameter shared by all markers (which is the default model). Then, you'd run a customized model where you include a second width parameter, so that one estimates the width for markers 1 & 2, and the other specifies a width for marker 3. (You set these conditions in option 6 of the 5th screen.) The model with the lowest AICc score is your best-supported hypothesis, but models within 2 AICc units aren't significantly worse.

Hypothesis testing within a single cline

What is the best-fitting overall shape?

Option (4) on screen 5 asks you to decide which model type you'd like to fit. The choices it gives you are determined by the type of data you have. If you have more than one trait, then the default model will for the most part represent the shape of the average cline.

> The exception involves data sets that include continuous traits. Each of these traits has unique mean phenotypes on each side and unique phenotypic variances, so these parameters must be estimated separately for each. By default, the remaining parameters, including center, width and the shapes of the tails, are shared among all traits by default, and so represent averages over all traits.

Thus, in the first pass, determining the best-fitting shape entails running each of the model options to find which has the lowest AICc value.

Customizing your model:

Eliminating parameters from a cline shape

Are clines asymmetrical? You may need to know if the shape of your cline is symmetrical, perhaps with a broad tail on the right but a simple sigmoid shape on the left. You would need to customize the model by specifying a tailed model and then turning off the left tail (thetaL and zL). Or, perhaps your genetic marker is fixed on the left side but remains polymorphic on the right side; you could fit a simpler model by turning off the estimation of the left asymptotic allele frequency (pL). If you have more than one trait, perhaps you wish to turn off parameters for some but not all of them.

ClineFit won't give you the option to turn on parameters that aren't already in the global model you selected. So, in option (4) on screen 5, first pick the global model type that includes every parameter you will want to include among the traits. Then, use option (5) to turn parameters off within that model type for selected traits.

```
5   Turn off/on estimation of some parameters:  default
```

The example continues from the section Running Your Data. We had chosen model `>6` for option (4) of the 5th screen. This included parameters c, w, pL, pR and D. We chose `>5` chromosomes for option (4a). The fifth screen reads

```
The program's current settings are:
     1 Input file name: SampleClineData1.txt
     2 Base output file name: SampleClineData1Out
     3 Set orientations of clines:  as surmised from data
     4 Estimation model:      6. simple sigmoid cline: c, w, pL, pR & D -- ends fitted, 
with disequilibrium
       4a     Effective number of chromosomes: 5
     5 Turn off/on estimation of some parameters:  default
     6 Turn off/on co-estimation of parameters among traits:  default
     7 Starting parameter values to facilitate convergence:  default
     8 Estimating the model only, or also sampling for support limits:  finding the best 
parameter estimates only

Choose the setting you want change.
? (1-8/done)


>5
**************************
By default, cline shape parameters are estimated using all the traits.  Here, you have an
opportunity to entirely omit from estimation the parameters of some traits.  For example,
you may wish to skip the estimation of introgression tails for one of your traits,
instead just estimating a 2-parameter model.  Or, you may wish to skip the estimation of
an introgression tail on (say) the left side for one of your traits.  Or, you may have
some traits that are fixed at 0 or 1 at the ends of the cline, and others for which you
need to estimate asymptotic frequencies.  By default, c & w are estimated for all traits,
and pL & pR are estimated independently for all quantitative traits.  When both genetic
markers and quantitative traits are present, by default the genetic markers are fixed at
0 and 1 on their respective sides of the cline.  Turn on pL and pR here for those traits
```

```
if you need to.
Current on/off settings:
1     c w      -      -      D      Dc      Dsd
2     c w      -      -      D      Dc      Dsd
3     c w      -      -      D      Dc      Dsd
4     c w      -      -      D      Dc      Dsd
5     c w      -      -      D      Dc      Dsd
6     c w      -      -      D      Dc      Dsd
7     c w      pL     pR     -      -       -
8     c w      pL     pR     -      -       -

     (Enter 'default' at anytime if you wish to return to the default condition.)
Do you have any parameters that you would like to turn on/off? (Y/N/default)
```

In this list, "–" means the parameter is turned off for that trait. By default, if your data include continuous traits and genetic markers both, ClineFit assumes that the ends of the clines are fixed at 0 and 1 for the genetic markers, even though pL and pR have to be fitted for the continuous traits. If you don't have continuous traits, the default will be to estimate pL and pR independently for each genetic marker.

To change the settings, choose

```
>Y
     First choose a parameter, then you will get a choice of traits for that parameter.
Enter 'done' when you are satisfied with your settings.
The parameters you can turn on/off are:
     pL       -- asymptotic allele frequency on the left side
       (if turned off, you assume pL=0 (or 1 for a left-high cline) asymptotically for
that trait)
     pR       -- asymptotic allele frequency on the right side
       (if turned off, you assume pR=1 (or 0 for a left-high cline) asymptotically for
that trait)
Which do wish to turn on/off? (pL/pR/done/default)
```

Your only viable options for model 6 are to change the pL and pR settings for the genetic markers. You cannot turn off the centers or widths for any traits. Further, you cannot turn off D for any of the genetic markers (1-6), or pL & pR for the continuous traits (7 & 8). If you had chosen, say, models 11 or 12, you would have the additional option of turning on or off the tails (theta and z) of any of the traits.

Let's change pL:

```
>pL
The current settings for parameter pL are:
trait# on/off  trait name
     1 off          c_1     (assumes pL=0.0 for this trait)
     2 off          d_1     (assumes pL=0.0 for this trait)
     3 off          xc_1    (assumes pL=0.0 for this trait)
     4 off          xd_1    (assumes pL=0.0 for this trait)
     5 off          cyf_1   (assumes pL=0.0 for this trait)
     6 off          cym_1   (assumes pL=0.0 for this trait)
     7 on           phenotype_1
     8 on           phenotype_2
     To change a setting, enter the trait name with the state you want it to show in
parentheses.  For example, if you want to turn off the estimation of trait 1, enter
```

```
'1(off)'.  To turn on the estimation at trait 2, you would enter '2(on)'.  If you want 1
and 3 through 5, enter 1,3-5(off).  Remember that you can't turn off pL or pR for
quantitative traits.  Don't put any white space between the characters -- '1 (off)' or
'1, 2(off)' won't be understood correctly.  To set all the traits back to the default
state for this parameter, enter 'default'.  When finished with this parameter, enter
'done'.
Enter the trait number and setting  [#(on)/#(off)/default/done]
```

Let's say the dominant markers don't seem to be fixed on the left side, so we'll try a run that estimates them. Change them and go back to the original screen

```
>2,4(on)
The current settings for parameter pL are:
trait# on/off     trait name
      1 off             c_1      (assumes pL=0.0 for this trait)
      2 on              d_1
      3 off             xc_1     (assumes pL=0.0 for this trait)
      4 on              xd_1
      5 off             cyf_1    (assumes pL=0.0 for this trait)
      6 off             cym_1    (assumes pL=0.0 for this trait)
      7 on              phenotype_1
      8 on              phenotype_2
      To change a setting, enter the trait name with the state you want it to show in parentheses.
For example, if you want to turn off the estimation of trait 1, enter '1(off)'.  To turn on the
estimation at trait 2, you would enter '2(on)'.  If you want 1 and 3 through 5, enter 1,3-5(off).
Remember that you can't turn off pL or pR for quantitative traits.  Don't put any white space between
the characters -- '1 (off)' or '1, 2(off)' won't be understood correctly.  To set all the traits back
to the default state for this parameter, enter 'default'.  When finished with this parameter, enter
'done'.
Enter the trait number and setting  [#(on)/#(off)/default/done]
```

```
>done
**************************
By default, cline shape parameters are estimated using all the traits.  Here, you have an opportunity
to entirely omit from estimation the parameters of some traits.  For example, you may wish to skip the
estimation of introgression tails for one of your traits, instead just estimating a 2-parameter model.
Or, you may wish to skip the estimation of an introgression tail on (say) the left side for one of
your traits.  Or, you may have some traits that are fixed at 0 or 1 at the ends of the cline, and
others for which you need to estimate asymptotic frequencies.  By default, c & w are estimated for all
traits, and pL & pR are estimated independently for all quantitative traits.  When both genetic
markers and quantitative traits are present, by default the genetic markers are fixed at 0 and 1 on
their respective sides of the cline.  Turn on pL and pR here for those traits if you need to.
Current on/off settings:
1      c w       -       -       D       Dc      Dsd
2      c w       pL      -       D       Dc      Dsd
3      c w       -       -       D       Dc      Dsd
4      c w       pL      -       D       Dc      Dsd
5      c w       -       -       D       Dc      Dsd
6      c w       -       -       D       Dc      Dsd
7      c w       pL      pR      -       -       -
8      c w       pL      pR      -       -       -

      (Enter 'default' at anytime if you wish to return to the default condition.)
Do you have any parameters that you would like to turn on/off? (Y/N/default)
```

You can go further or fix mistakes, but we'll stop here.

```
>N
**************************


The program's current settings are:
      1 Input file name: SampleClineData1.txt
      2 Base output file name: SampleClineData1Out
      3 Set orientations of clines:  as surmised from data
      4 Estimation model:      6. simple sigmoid cline: c, w, pL, pR & D -- ends fitted,
with disequilibrium
        4a      Effective number of chromosomes: 5
      5 Turn off/on estimation of some parameters:  done
      6 Turn off/on co-estimation of parameters among traits:  default
      7 Starting parameter values to facilitate convergence:  default
      8 Estimating the model only, or also sampling for support limits:  finding the best
parameter estimates only

Choose the setting you want change.
? (1-8/done)
```

6 Turn off/on co-estimation of parameters among traits: default

Here you can allow some parameters to vary freely among traits, while constraining others to vary in groups of traits. Option (6) will revert to the default state if you go back and change option (5).

```
>6
*************************
By default, common cline shape parameters are estimated using information from all the traits.  For
example, one common center location is co-estimated for all traits.  (The default exception is that pL
and pR are estimated separately for each trait.)  Here, you can allow sets of traits that remain
turned on to vary independently.  For example, you may wish to estimate separate centers for each
trait, or a width shared by traits 1 & 2 that may differ from the width estimated for traits 3-5.
      First you will choose the parameter you want to adjust, then you can assign identities to
control co-estimation among the traits.  Enter 'default' at anytime if you wish to restore the default
states.)

Parameters sharing the same numerical value in more than one trait will be co-estimated.  For example,
traits that share 0's for their first entries will be fitted to a single, shared center; those that
share 1's will be fitted to a second center, etc.
(Some parameters cannot be co-estimated and they are not shown here.  For example, the tail ends of
clines in continuous traits, as well as their phenotypic variances, must be estimated independently.)
Current settings for co-estimable parameters:
      c    w      pL     pR
1     0    0
2     0    0
3     0    0
4     0    0
5     0    0
6     0    0
7     0    0
8     0    0
D center:       tied to cline center

Do you have any parameters that you would like to adjust? (Y/N/default)
```

```
>Y
      First choose a parameter, then you will get a choice of traits for that parameter.  Enter 'done'
when you are satisfied with your settings.
The parameter co-estimation settings you can modify are:
      c  -- centers
      w  -- widths
      pL -- asymptotic allele frequency or phenotypic mean on the left side
      pR -- asymptotic allele frequency or phenotypic mean on the right side
      Dc -- The geographic center of the disequilibrium curve
                (currently co-estimated with the cline center)
Which do wish to adjust? (c/w/pL/pR/Dc/default/done)
```

In general, it would be silly to co-estimate pL, pR or Vp parameters among markers within a given transect, unless you can think of some biological reason why qualitatively different markers should reach the same frequencies or phenotypes. However, when comparing cline transects scored for the same traits, comparing runs that do vs. don't co-estimate pL, pR or Vp for those traits can be informative.

Hypothesis tests involving more than one transect, or subgroups of individuals within a transect

Data setup is described at the end of the Data Files section. You will need to customize your models to run these analyses, and those steps are described in the Customize section above. The difference between comparing separate clines within a transect and separate transects involve only your customization settings.

Comparing transect shapes

The main constraint in comparing shapes among transects is that different transects will typically have different centers. Your simplest comparisons, involving overall average shape differences, therefore need to include independently estimated centers for different transects.

> There is an alternative that reduces the number of parameters. You could estimate the average cline shapes independently first to determine their respective center locations. Next, reformat the locations in your data file so that each transect is zeroed to its center. Then, in the comparisons among transects, estimate a single center shared by traits on both transects.
>
> > For example, if the ML-estimated center of transect J is c_j and that of transect K is c_k, then rescore each location of transect to be $x–c_j$, and each location of transect J to $x–c_k$.

From here, your comparisons involve turning parameters on or off, and specifying whether parameters are co-estimated across clines. For example, do clines on transects J and K have different shapes? Assign all the traits of transects J a single co-estimation setting, and give those of transect K another.

Here is an example involving the data from <Sample2ClineData.txt>, which includes all data types. The transects have different lengths and the centers aren't zeroed. The genetic markers are fixed in the tails by default, and for the quantitative traits, pL, pR and Vp are all estimated independently.

Do the transects have different average shapes?

Choosing option 6,

```
    6 Turn off/on co–estimation of parameters among traits:  default
>6
```

the initial, default settings are

```
Current settings for co-estimable parameters:
      c w        pL       pR      Vp
1     0 0                                  Jc_1
2     0 0                                  Jc_2
3     0 0                                  Jd_1
4     0 0                                  Jxc_1
5     0 0                                  Jxd_1
6     0 0                                  Jcyf_1
7     0 0                                  Jcym_1
8     0 0        0        0       0        Jphenotype_1
9     0 0        1        1       1        Jphenotype_2
10    0 0                                  Kc_1
11    0 0                                  Kc_2
12    0 0                                  Kc_3
13    0 0                                  Kd_1
14    0 0                                  Kxc_1
15    0 0                                  Kxd_1
16    0 0                                  Kcyf_1
17    0 0                                  Kcym_1
18    0 0        2        2       2        Kphenotype_1
19    0 0        3        3       3        Kphenotype_2

Do you have any parameters that you would like to adjust? (Y/N/default)
```

Give the clines separate centers and widths following the instructions for this section, to get

```
Current settings for co-estimable parameters:
      c w        pL       pR      Vp
1     0 0                                  Jc_1
2     0 0                                  Jc_2
3     0 0                                  Jd_1
4     0 0                                  Jxc_1
5     0 0                                  Jxd_1
6     0 0                                  Jcyf_1
7     0 0                                  Jcym_1
8     0 0        0        0       0        Jphenotype_1
9     0 0        1        1       1        Jphenotype_2
10    1 1                                  Kc_1
11    1 1                                  Kc_2
12    1 1                                  Kc_3
13    1 1                                  Kd_1
14    1 1                                  Kxc_1
15    1 1                                  Kxd_1
16    1 1                                  Kcyf_1
17    1 1                                  Kcym_1
18    1 1        2        2       2        Kphenotype_1
19    1 1        3        3       3        Kphenotype_2
```

Then run the analysis. You can compare this to the case where the widths are all co-estimated (table setting=0). If you zero the population locations of the transects to their centers, you don't need to specify separate centers.

Do clines of the same traits on different transects have the same shapes?

I'll use just a subset of the traits for illustrative purposes. From the data-formatting protocol in the Data Files section, data columns labeled Jc_1 and Kc_1 are the same biological trait. Compare the cases where they are estimated independently to the case where they are estimated together:

```
Current settings for co-estimable parameters:
      c w        pL       pR      Vp
1     0 0                                  Jc_1
2     0 1                                  Jc_2
8     0 2        0        0       0        Jphenotype_1
9     0 3        1        1       1        Jphenotype_2
10    1 0                                  Kc_1
11    1 1                                  Kc_2
18    1 2        0        0       2        Kphenotype_1
19    1 3        1        1       3        Kphenotype_2
```

Here, the centers are transect-specific; the widths of Jc_1 and Kc_1 are co-estimated, etc.; and for the quantitative traits, the left and right trait means are co-estimated. The phenotypic variances are estimated independently for all quantitative traits. Run this model and get its AICc score.

Compare that to a model where widths are estimated independently:

```
Current settings for co–estimable parameters:
      c w       pL      pR      Vp
1     0 0                               Jc_1
2     0 1                               Jc_2
8     0 2       0       0       0       Jphenotype_1
9     0 3       1       1       1       Jphenotype_2
10    1 4                               Kc_1
11    1 5                               Kc_2
18    1 6       0       0       2       Kphenotype_1
19    1 7       1       1       3       Kphenotype_2
```

Run this model and get its AICc score, and compare it to the previous model; the lower score wins.

If this new model fits better, which traits are primarily responsible for the difference? Set various subsets of the traits to vary independently between transects and get their AICc scores to narrow it down. You can likewise fiddle with the pL or pR settings, including first turning on (option 5) pL or pR for particular traits, or on different transects for particular traits.

Trying all these combinations can get tedious especially if there are a lot of traits. It probably pays to graph the results from the separate transects first and identify the interesting follow-up comparisons by eye. Of course, in studies of particular traits, the biology itself will dictate the most interesting comparisons.

Comparing disequilibrium among transects: It's not currently possible to compare disequilibrium among transects. You can only compare transects using shape models that omit disequilibrium. (ClineFit fits a single set of disequilibrium parameters for all genetic markers at once. You can't presently customize the disequilibrium settings so that disequilibrium curves are estimated independently for different subsets of loci, which would be necessary to set up comparisons between transects.) Of course, you can still estimate D curves for transects individually.

Graphing your output

ClineFit isn't smart enough to handle graphics, so you're on your own. (It would be wonderful if somebody wrote some functions in R to make pretty plots from the ClineFit output files.)

I use Mathematica functions to plot my clines, and I've included a Mathematica notebook that you are free to use. The constraint is that Mathematica is expensive, although if you or a colleague are at a research university, you can probably track down somebody who has a copy.

Mathematica's graphics are low-resolution and choppy, and sometimes downright ugly when you paste them into other formats. For making publication-quality graphics, I paste them into an art program such as Canvas or Illustrator, trace over the cline shapes to get smooth curves, and overwrite axis labels.

References

This is a cursory, very superficial list that leaves out a wide swath of relevant literature that explains how clines work. I'll fill it properly in a less preliminary draft of the manual.

Burnham, K.P. & B.R. Anderson. 2004. Mutimodal inference: understanding AIC and BIC in model selection. *Sociological Methods & Research* 33: 261–304.

Edwards, A.W.F. 1991. *Likelihood*. Cambridge University Press, Cambridge.

Hill, W.H. 1974. Estimation of linkage disequilibrium in randomly mating populations. *Heredity* 33: 229–239.

Szymura, J.M. & N.H. Barton. 1986. Genetic analysis of a hybrid zone between the Fire-Bellied Toads (*Bombina bombina* and *B. variegata*) near Cracow in southern Poland. *Evolution* 40: 1141–1159.

Szymura, J.M. & N.H. Barton. 1991. The genetic structure of the hybrid zone between the Fire-Bellied Toads *Bombina bombina* and *B. variegata* between transects and between loci. *Evolution* 45: 237–261.

Porter, A. H., R. Wenger, H. J. Geiger, A. Scholl, & A. M. Shapiro. 1997. The *Pontia daplidice-edusa* hybrid zone in northwestern Italy. *Evolution* 52: 1561-1573.

Appendix:

Details of the disequilibrium estimators

Notation: Loci **A** and **B** each have two alleles {A, a} and {B, b}. The sample size is *N*, subscripted by genotype; for example, N_{AaBb} represents the double heterozygote. For cytoplasmic or males of haplodiploid/heterogametic loci, the subscripts would follow the form N_{AB}, in this case the number of AB genotypes. The set of genotype numbers in a population is **N**. For loci with dominance, x represents the unknown allele of the dominant phenotype, for example N_{Axbb} is the observed number of individuals with the dominant phenotype at locus **A** and recessive phenotype at locus **B**. The expected gamete frequency is *f*, again subscripted by gamete genotype; e.g., f_{AB} represents the expected frequency of the AB gamete. The frequency of allele A at locus **A** is *p*, and *q* is the frequency of allele B at locus **B**. Disequilibrium is subscripted by the inheritance type of the loci involved: *c*: codominant; *d*: dominant; *xc*: sex-linked codominant; *xd*: sex-linked dominant; *cyf*: maternally inherited cytoplasmic locus; *cym*: paternally inherited cytoplasmic locus; *cyfo*: cytoplasmic locus present only in females; and *cymo*: loci present only in males (e.g., Y chromosomes). For example, $D_{c\times d}$ is the disequilibrium between a codominant and a dominant locus.

Maximum likelihood estimators: Given a set of expected gamete frequencies, the probability of a set of observed genotype frequencies follows a multinomial distribution conditioned on the inheritance pattern (Hill 1974). The expected gamete frequencies *f* are simple functions of the observed allele or dominant-phenotype frequencies and the expected D:

$$f_{AB} = pq + D$$

$$f_{Ab} = p(1-q) - D$$

$$f_{aB} = (1-p)q - D$$

$$f_{ab} = (1-p)(1-q) + D$$

where *p* and *q* are the observed allele frequencies. The likelihood of the gamete frequencies, and therefore the disequilibrium estimate, follows the same distribution.

Here the **N**'s represent sets of zygote genotype frequencies.

Allele frequencies for most inheritance types are estimated by direct count. For dominant markers, ClineFit uses $p = 1 - \sqrt{N_{aa}/n}$, with *n* the number of individuals. For sex-linked dominant markers,

$$p = \frac{2n_F\left(1-\sqrt{N_{aa}/n_F}\right)+N_A}{2n_F+n_M}$$

where n_F and n_M are numbers of females and males in the sample.

The estimators, expressed for simplicity in terms of the gametic frequencies (*f*'s) rather than *p*, *q* and *D*, and observed zygote frequencies (**N**) of a single population, are as follows. Throughout, the label 'female' refer to the homogametic sex and 'male' refers to the heterogametic/haploid sex; swap these labels if the homogametic sex is male.

$$\begin{aligned}\ln L\left(D_{c\times c}\left|\mathbf{N}\right.\right) = {} & 2N_{AABB}\ln f_{AB} + N_{AABb}\ln\left(f_{AB}f_{Ab}\right) + 2N_{AAbb}\ln f_{Ab} \\ & + N_{AaBB}\ln\left(f_{AB}f_{aB}\right) + N_{AaBb}\ln\left(f_{AB}f_{ab} + f_{Ab}f_{aB}\right) + N_{Aabb}\ln\left(f_{Ab}f_{ab}\right) \\ & + 2N_{aaBB}\ln f_{aB} + N_{aaBb}\ln\left(f_{aB}f_{ab}\right) + 2N_{aabb}\ln f_{ab} + C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{d\times d}\left|\mathbf{N}\right.\right) = {} & 2N_{AxBx}\ln\left(f_{AB}\left(2 - f_{AB}\right) + 2f_{Ab}f_{aB}\right) \\ & + N_{Axbb}\ln\left(f_{Ab}^{2} + 2f_{Ab}f_{ab}\right) + N_{aaBx}\ln\left(f_{aB}^{2} + 2f_{aB}f_{ab}\right) + 2N_{aabb}\ln f_{ab} + C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{xc\times xc}\left|\mathbf{N}\right.\right) = {} & 2N_{AABB}\ln f_{AB} + N_{AABb}\ln\left(f_{AB}f_{Ab}\right) + 2N_{AAbb}\ln f_{Ab} \\ & + N_{AaBB}\ln\left(f_{AB}f_{aB}\right) + N_{AaBb}\ln\left(f_{AB}f_{ab} + f_{Ab}f_{aB}\right) + N_{Aabb}\ln\left(f_{Ab}f_{ab}\right) \\ & + 2N_{aaBB}\ln f_{aB} + N_{aaBb}\ln\left(f_{aB}f_{ab}\right) + 2N_{aabb}\ln f_{ab} \\ & + N_{AB}\ln f_{AB} + N_{Ab}\ln f_{Ab} + N_{aB}\ln f_{aB} + N_{ab}\ln f_{ab} + C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{xd\times xd}\left|\mathbf{N}\right.\right) = {} & 2N_{AxBx}\ln\left(\tfrac{1}{2}f_{AB}^{2} + f_{AB}\left(1 - f_{AB}\right) + f_{Ab}f_{aB}\right) \\ & + N_{Axbb}\ln\left(\tfrac{1}{2}f_{Ab}^{2} + f_{Ab}f_{ab}\right) + N_{aaBx}\ln\left(\tfrac{1}{2}f_{aB}^{2} + f_{aB}f_{ab}\right) + 2N_{aabb}\ln f_{ab} \\ & + N_{AB}\ln f_{AB} + N_{Ab}\ln f_{Ab} + N_{aB}\ln f_{aB} + N_{ab}\ln f_{ab} + C\end{aligned}$$

$$\ln L\left(D_{cyf\times cyf}\left|\mathbf{N}\right.\right) = N_{AB}\ln f_{AB} + N_{Ab}\ln f_{Ab} + N_{aB}\ln f_{aB} + N_{ab}\ln f_{ab} + C$$

$$\ln L\left(D_{cym\times cym}\left|\mathbf{N}\right.\right) = N_{AB}\ln f_{AB} + N_{Ab}\ln f_{Ab} + N_{aB}\ln f_{aB} + N_{ab}\ln f_{ab} + C$$

$$\ln L\left(D_{cyfo\times cyfo}\left|\mathbf{N}\right.\right) = N_{AB}\ln f_{AB} + N_{Ab}\ln f_{Ab} + N_{aB}\ln f_{aB} + N_{ab}\ln f_{ab} + C$$

$$\ln L\left(D_{cymo\times cymo}\left|\mathbf{N}\right.\right) = N_{AB}\ln f_{AB} + N_{Ab}\ln f_{Ab} + N_{aB}\ln f_{aB} + N_{ab}\ln f_{ab} + C$$

$$\begin{aligned}\ln L\left(D_{c\times d}\left|\mathbf{N}\right.\right) = {} & 2N_{AABx}\ln\left(f_{AB}^{2} + 2f_{AB}f_{Ab}\right) + 2N_{AAbb}\ln f_{Ab} \\ & + N_{AaBx}\ln\left(f_{AB}f_{aB} + f_{AB}f_{ab} + f_{Ab}f_{aB}\right) + N_{Aabb}\ln\left(f_{Ab}f_{ab}\right) \\ & + 2N_{aaBx}\ln\left(f_{aB}^{2} + 2f_{aB}f_{ab}\right) + 2N_{aabb}\ln f_{ab} + C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{c\times xc}\left|\mathbf{N}\right.\right) = {} & 2N_{AABB}\ln f_{AB} + N_{AABb}\ln\left(f_{AB}f_{Ab}\right) + 2N_{AAbb}\ln f_{Ab} \\ & +N_{AaBB}\ln\left(f_{AB}f_{aB}\right) + N_{AaBb}\ln\left(f_{AB}f_{ab} + f_{Ab}f_{aB}\right) + N_{Aabb}\ln\left(f_{Ab}f_{ab}\right) \\ & +2N_{aaBB}\ln f_{aB} + N_{aaBb}\ln\left(f_{aB}f_{ab}\right) + 2N_{aabb}\ln f_{ab} \\ & +N_{AAB}\ln\left(f_{AB}p\right) + N_{AAb}\ln\left(f_{Ab}p\right) + N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right) + N_{aab}\ln\left(f_{ab}\left(1-p\right)\right) \\ & +N_{AaB}\ln\left(f_{AB}\left(1-p\right) + f_{Ab}p\right) + N_{Aab}\ln\left(f_{Ab}\left(1-p\right) + f_{ab}p\right) + C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{c\times xd}\left|\mathbf{N}\right.\right) = {} & 2N_{AABx}\ln\left(\tfrac{1}{2}f_{AB}^2 + f_{AB}f_{Ab}\right) + 2N_{AAbb}\ln f_{Ab} \\ & +N_{AaBx}\ln\left(f_{AB}f_{aB} + f_{AB}f_{ab} + f_{Ab}f_{aB}\right) + N_{Aabb}\ln\left(f_{Ab}f_{ab}\right) \\ & +N_{aaBx}\ln\left(\tfrac{1}{2}f_{aB}^2 + f_{aB}f_{ab}\right) + 2N_{aabb}\ln f_{ab} \\ & +N_{AAB}\ln\left(f_{AB}p\right) + N_{AAb}\ln\left(f_{Ab}p\right) + N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right) + N_{aab}\ln\left(f_{ab}\left(1-p\right)\right) \\ & +N_{AaB}\ln\left(f_{AB}\left(1-p\right) + f_{Ab}p\right) + N_{Aab}\ln\left(f_{Ab}\left(1-p\right) + f_{ab}p\right) + C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{c\times cyf}\left|\mathbf{N}\right.\right) = {} & N_{AAB}\ln\left(f_{AB}p\right) + N_{AAb}\ln\left(f_{Ab}p\right) + N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right) + N_{aab}\ln\left(f_{ab}\left(1-p\right)\right) \\ & +N_{AaB}\ln\left(f_{AB}\left(1-p\right) + f_{aB}p\right) + N_{Aab}\ln\left(f_{Ab}\left(1-p\right) + f_{ab}p\right) + C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{c\times cym}\left|\mathbf{N}\right.\right) = {} & N_{AAB}\ln\left(f_{AB}p\right) + N_{AAb}\ln\left(f_{Ab}p\right) + N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right) + N_{aab}\ln\left(f_{ab}\left(1-p\right)\right) \\ & +N_{AaB}\ln\left(f_{AB}\left(1-p\right) + f_{aB}p\right) + N_{Aab}\ln\left(f_{Ab}\left(1-p\right) + f_{ab}p\right) + C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{c\times cym}\left|\mathbf{N}\right.\right) = {} & N_{AAB}\ln\left(f_{AB}p\right) + N_{AAb}\ln\left(f_{Ab}p\right) + N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right) + N_{aab}\ln\left(f_{ab}\left(1-p\right)\right) \\ & +N_{AaB}\ln\left(f_{AB}\left(1-p\right) + f_{aB}p\right) + N_{Aab}\ln\left(f_{Ab}\left(1-p\right) + f_{ab}p\right) + C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{c\times cymo}\left|\mathbf{N}\right.\right) = {} & N_{AAB}\ln\left(f_{AB}p\right) + N_{AAb}\ln\left(f_{Ab}p\right) + N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right) + N_{aab}\ln\left(f_{ab}\left(1-p\right)\right) \\ & +N_{AaB}\ln\left(f_{AB}\left(1-p\right) + f_{aB}p\right) + N_{Aab}\ln\left(f_{Ab}\left(1-p\right) + f_{ab}p\right) + C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{d\times xc}\left|\mathbf{N}\right.\right) = {} & 2N_{AxBB}\ln\left(\tfrac{1}{2}f_{AB}^2 + f_{AB}f_{aB}\right) + 2N_{aaBB}\ln f_{aB} \\ & +N_{AxBb}\ln\left(f_{AB}f_{Ab} + f_{AB}f_{ab} + f_{Ab}f_{aB}\right) + N_{aaBb}\ln\left(\tfrac{1}{2}f_{ab}^2 + f_{aB}f_{ab}\right) \\ & +2N_{Axbb}\ln\left(\tfrac{1}{2}f_{Ab}^2 + f_{Ab}f_{ab}\right) + 2N_{aabb}\ln f_{ab} \\ & +N_{AxB}\ln\left(\tfrac{1}{2}f_{AB}p + f_{AB}\left(1-p\right) + f_{aB}p\right) + N_{Axb}\ln\left(\tfrac{1}{2}f_{Ab}p + f_{Ab}\left(1-p\right) + f_{ab}p\right) \\ & N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right) + N_{aab}\ln\left(f_{ab}\left(1-p\right)\right) + C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{d\times xd}\left|\mathbf{N}\right.\right) = {} & N_{AxBx}\ln\left(\tfrac{1}{2}f_{AB}^2 + f_{AB}f_{aB} + f_{AB}f_{Ab} + f_{Ab}f_{aB} + f_{AB}f_{ab}\right) \\ & +N_{Axbb}\ln\left(\tfrac{1}{2}f_{Ab}^2 + f_{Ab}f_{ab}\right) + N_{aaBx}\ln\left(\tfrac{1}{2}f_{aB}^2 + f_{aB}f_{ab}\right) + 2N_{aabb}\ln f_{ab} \\ & +N_{AxB}\ln\left(\tfrac{1}{2}f_{AB}p + f_{AB}\left(1-p\right) + f_{aB}p\right) + N_{Axb}\ln\left(\tfrac{1}{2}f_{Ab}p + f_{Ab}\left(1-p\right) + f_{ab}p\right) \\ & +N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right) + N_{aab}\ln\left(f_{ab}\left(1-p\right)\right) + C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{d\times cyf}\left|\mathbf{N}\right.\right) = {} & N_{AxB}\ln\left(f_{AB}p+f_{AB}\left(1-p\right)+f_{aB}p\right)+N_{Axb}\ln\left(f_{Ab}p+f_{Ab}\left(1-p\right)+f_{ab}p\right)\\ & +N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right)+N_{aab}\ln\left(f_{ab}\left(1-p\right)\right)+C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{d\times cym}\left|\mathbf{N}\right.\right) = {} & N_{AxB}\ln\left(f_{AB}p+f_{AB}\left(1-p\right)+f_{aB}p\right)+N_{Axb}\ln\left(f_{Ab}p+f_{Ab}\left(1-p\right)+f_{ab}p\right)\\ & +N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right)+N_{aab}\ln\left(f_{ab}\left(1-p\right)\right)+C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{d\times cyfo}\left|\mathbf{N}\right.\right) = {} & N_{AxB}\ln\left(f_{AB}p+f_{AB}\left(1-p\right)+f_{aB}p\right)+N_{Axb}\ln\left(f_{Ab}p+f_{Ab}\left(1-p\right)+f_{ab}p\right)\\ & +N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right)+N_{aab}\ln\left(f_{ab}\left(1-p\right)\right)+C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{d\times cymo}\left|\mathbf{N}\right.\right) = {} & N_{AxB}\ln\left(f_{AB}p+f_{AB}\left(1-p\right)+f_{aB}p\right)+N_{Axb}\ln\left(f_{Ab}p+f_{Ab}\left(1-p\right)+f_{ab}p\right)\\ & +N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right)+N_{aab}\ln\left(f_{ab}\left(1-p\right)\right)+C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{xc\times xd}\left|\mathbf{N}\right.\right) = {} & 2N_{AABx}\ln\left(\tfrac{1}{2}f_{AB}^{2}+f_{AB}f_{Ab}\right)+2N_{AAbb}\ln f_{Ab}\\ & +N_{AaBx}\ln\left(f_{AB}f_{aB}+f_{Ab}f_{aB}+f_{AB}f_{ab}\right)+N_{Aabb}\ln\left(f_{Ab}f_{ab}\right)\\ & +N_{aaBx}\ln\left(\tfrac{1}{2}f_{aB}^{2}+f_{aB}f_{ab}\right)+2N_{aabb}\ln f_{ab}\\ & +N_{AB}\ln f_{AB}+N_{Ab}\ln f_{Ab}+N_{aB}\ln f_{aB}+N_{ab}\ln f_{ab}+C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{xc\times cyf}\left|\mathbf{N}\right.\right) = {} & N_{AAB}\ln\left(f_{AB}p\right)+N_{AAb}\ln\left(f_{Ab}p\right)+N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right)+N_{aab}\ln\left(f_{ab}\left(1-p\right)\right)\\ & +N_{AaB}\ln\left(f_{AB}p+f_{aB}\left(1-p\right)\right)+N_{Aab}\ln\left(f_{Ab}p+f_{ab}\left(1-p\right)\right)\\ & +N_{AB}\ln f_{AB}+N_{Ab}\ln f_{Ab}+N_{aB}\ln f_{aB}+N_{ab}\ln f_{ab}+C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{xc\times cym}\left|\mathbf{N}\right.\right) = {} & N_{AAB}\ln\left(f_{AB}p\right)+N_{AAb}\ln\left(f_{Ab}p\right)+N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right)+N_{aab}\ln\left(f_{ab}\left(1-p\right)\right)\\ & +N_{AaB}\ln\left(f_{AB}p+f_{aB}\left(1-p\right)\right)+N_{Aab}\ln\left(f_{Ab}p+f_{ab}\left(1-p\right)\right)\\ & +N_{AB}\ln f_{AB}+N_{Ab}\ln f_{Ab}+N_{aB}\ln f_{aB}+N_{ab}\ln f_{ab}+C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{xc\times cyfo}\left|\mathbf{N}\right.\right) = {} & N_{AAB}\ln\left(f_{AB}p\right)+N_{AAb}\ln\left(f_{Ab}p\right)+N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right)+N_{aab}\ln\left(f_{ab}\left(1-p\right)\right)\\ & +N_{AaB}\ln\left(f_{AB}p+f_{aB}\left(1-p\right)\right)+N_{Aab}\ln\left(f_{Ab}p+f_{ab}\left(1-p\right)\right)\\ & +N_{AB}\ln f_{AB}+N_{Ab}\ln f_{Ab}+N_{aB}\ln f_{aB}+N_{ab}\ln f_{ab}+C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{xc\times cymo}\left|\mathbf{N}\right.\right) = {} & N_{AAB}\ln\left(f_{AB}p\right)+N_{AAb}\ln\left(f_{Ab}p\right)+N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right)+N_{aab}\ln\left(f_{ab}\left(1-p\right)\right)\\ & +N_{AaB}\ln\left(f_{AB}p+f_{aB}\left(1-p\right)\right)+N_{Aab}\ln\left(f_{Ab}p+f_{ab}\left(1-p\right)\right)\\ & +N_{AB}\ln f_{AB}+N_{Ab}\ln f_{Ab}+N_{aB}\ln f_{aB}+N_{ab}\ln f_{ab}+C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{xd\times cyf}\left|\mathbf{N}\right.\right) = {} & N_{AxB}\ln\left(\tfrac{1}{2}f_{AB}p+f_{AB}\left(1-p\right)+f_{aB}p\right)+N_{Axb}\ln\left(\tfrac{1}{2}f_{Ab}p+f_{Ab}\left(1-p\right)+f_{ab}p\right)\\ & +N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right)+N_{aab}\ln\left(f_{ab}\left(1-p\right)\right)\\ & +N_{AB}\ln f_{AB}+N_{Ab}\ln f_{Ab}+N_{aB}\ln f_{aB}+N_{ab}\ln f_{ab}+C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{xd\times cym}\middle|\mathbf{N}\right) = {} & N_{AxB}\ln\left(\tfrac{1}{2}f_{AB}p+f_{AB}\left(1-p\right)+f_{aB}p\right)+N_{Axb}\ln\left(\tfrac{1}{2}f_{Ab}p+f_{Ab}\left(1-p\right)+f_{ab}p\right)\\ & +N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right)+N_{aab}\ln\left(f_{ab}\left(1-p\right)\right)\\ & +N_{AB}\ln f_{AB}+N_{Ab}\ln f_{Ab}+N_{aB}\ln f_{aB}+N_{ab}\ln f_{ab}+C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{xd\times cyfo}\middle|\mathbf{N}\right) = {} & N_{AxB}\ln\left(\tfrac{1}{2}f_{AB}p+f_{AB}\left(1-p\right)+f_{aB}p\right)+N_{Axb}\ln\left(\tfrac{1}{2}f_{Ab}p+f_{Ab}\left(1-p\right)+f_{ab}p\right)\\ & +N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right)+N_{aab}\ln\left(f_{ab}\left(1-p\right)\right)\\ & +N_{AB}\ln f_{AB}+N_{Ab}\ln f_{Ab}+N_{aB}\ln f_{aB}+N_{ab}\ln f_{ab}+C\end{aligned}$$

$$\begin{aligned}\ln L\left(D_{xd\times cymo}\middle|\mathbf{N}\right) = {} & N_{AxB}\ln\left(\tfrac{1}{2}f_{AB}p+f_{AB}\left(1-p\right)+f_{aB}p\right)+N_{Axb}\ln\left(\tfrac{1}{2}f_{Ab}p+f_{Ab}\left(1-p\right)+f_{ab}p\right)\\ & +N_{aaB}\ln\left(f_{aB}\left(1-p\right)\right)+N_{aab}\ln\left(f_{ab}\left(1-p\right)\right)\\ & +N_{AB}\ln f_{AB}+N_{Ab}\ln f_{Ab}+N_{aB}\ln f_{aB}+N_{ab}\ln f_{ab}+C\end{aligned}$$

The 1/2 factors in the sex-linked dominant cases account for the expected sex ratio. It isn't possible to factor them out of the dominant-phenotype cases, but otherwise they are factored out and absorbed into the constant term *C*. They also reside in the constant *C* in sex-linked codominant cases. The sex-ratio parameter indeed factors out of all models except those involving sex-linked dominant inheritance, so unequal sex ratios would affect *D* estimation for only those. Nevertheless, for this reason, ClineFit assumes 1:1 sex ratios exist in the source populations. You can pretty much ignore that assumption if you don't have sex-linked dominant markers, but you might keep it in the back of your mind if you have quantitative traits.

Appendix 2: Sample data files

ClineFit comes with some sample data and output files. Some are used in the examples on in the Running the Program section. Some are used in the Mathematica notebook that has some graphics functions. Here are the contents and specs.

How sample clines were created:

All are produced by individual-based simulation using an ecotone to determine fitnesses. Each starts with individuals carrying unlinked loci of the appropriate combination of inheritance patterns, fixed for different alleles on opposite sides of the ecotone. Individuals experience selection as zygotes, disperse and mate randomly to produce the next zygote generation, and the parents die. The hybrid index for the genetic markers is calculated (see the Hybrid Index section) and used to determine fitness along the ecotone.

None of the sample clines has tails significantly different from the cline center shape.

Dominant traits are scored so that the recessive genotype has value 0 and the dominant phenotype has value 1. The actual genotypes are used during reproduction.

Each quantitative trait is produced using an underlying additive genetic model based on 10 codominant markers/trait, scaled to the 0-1 range. A random environmental deviation is assigned to each individual following a Gaussian distribution, using a trait-specific phenotypic variance with the mean at the individual's additive genetic trait value. After the simulation, the data set is manipulated to generate the data ranges and orientations appropriate for illustrating how ClineFit works. Quantitative traits are rescaled to an arbitrary range and inverted or not to determine their cline orientation. The codominant traits that underlie these quantitative traits are then deleted from the data set.

In all sample data files, the traits are given names that represent their inheritance patterns. Codominant traits have the prefix 'c,' dominant traits use 'd,' sex-linked codominant traits use 'xc,' sex-linked dominant traits use 'xd,' cytoplasmic female markers use 'cyf,' cytoplasmic male markers use 'cym,' and quantitative traits use 'phenotype.' These prefixes are for demonstration and ClineFit doesn't notice them, so it's not a format you need to use to construct your own data files.

In all sample data files, the cline center in the simulation generating the data is halfway between the ends of the transect.

The data files mostly differ in which traits are skipped, and may have different widths.

Each sample file comes with associated output files. The output files are produced using the default settings.

Data sets with 1 transect

SampleClineData1.txt

This data set contains two codominant and two quantitative traits, and single representatives of the other trait types. No traits are skipped.

SampleClineDataNoQ.txt

The same as SampleClineData1.txt, except the quantitative traits are skipped.

SampleClineDataQonly.txt

The same as SampleClineData1.txt, except the genetic markers are skipped.

SampleClineDataQ1only.txt

The same as SampleClineData1.txt, but skipping all traits except phenotype_1.

Data sets with 2 transects

Sample2ClineData.txt

Data from 2 transects, J and K. The transects are of different lengths and have different centers and widths. Transect K has an extra trait.

Sample2ClineDataNoQ.txt

The same as Sample2ClineData.txt, except the quantitative traits are skipped.

Sample2ClineDataQonly.txt

The same as Sample2ClineData.txt, except the genetic markers are skipped.

Output files used in constructing the graphics

SampleClineDataNoQOut_shape1.txt

SampleClineDataNoQOut_HI.txt

These are results from SampleClineDataNoQ.txt. The model fitted is

```
8.  tailed sigmoid cline: c, w, tL, tR, zL & zR -- ends fixed at 0 and 1
```

The cline was fitted using the default settings throughout, with the random number seed set to 1. So, parameters are co-estimated for all traits to produce a single shape, and the default settings are used to fit the parameters and collect support limits.

SampleClineDataQonlyOut_shape1.txt

SampleClineDataQonlyOut_shape2.txt

These are results from SampleClineDataQonly.txt. The model fitted is

```
5.  tailed sigmoid cline: c, w, pL & pR -- ends fitted
```

The cline was fitted using the default settings throughout. So, parameters are co-estimated for all traits to produce a single shape, and the default settings are used to fit the parameters and collect support limits.

SampleClineDataQ1onlyOut_shape1.txt

SampleClineDataQ1phenotypes.txt

The shape file results from SampleClineDataQ1only.txt. The model fitted is

```
8.  tailed sigmoid cline: c, w, tL, tR, zL & zR -- ends fixed at 0 and 1
```

The cline was fitted using the default settings throughout. So, parameters are co-estimated for all traits to produce a single shape, and the default settings are used to fit the parameters and collect support limits.

I constructed SampleClineDataQ1phenotypes.txt by hand from the data for trait 1. It takes the format of a _HI.txt file, except the scores are the values of phenotype1 for each individual.