Week 7 Computer Lab Introduction: Spatial Linear Models

VIDEO 1

Scene 1: Spatial Linear Models

Welcome to the Week 7 computer lab. This week, we want to test whether genetic diversity in this plant, *Dianthus carthusianorum*, is better explained by isolation-by-distance or isolation-by-resistance.

Genetic diversity may also be related to population size, because smaller populations have higher rates of genetic drift. Here we'll use grassland patch size as a proxy for population size. Finally, we want to know if the same factors explain genetic diversity across the study area.

We have a number of methodological challenges to address. This first video explains the neighbourhood-level analysis approach. We start by defining each connectivity hypothesis as a pairwise distance matrix. Then we use Hanski's connectivity index 'Si' to get a connectivity value for each focal patch, integrating over all its pairwise distances. Then we explain allelic richness as a measure of genetic diversity by the connectivity of each patch.

A second video shows how we can handle spatial autocorrelation in linear models.

Scene 2: Dianthus Data Set

Please see also the video by Yessica Rico that introduces the study system. On this map, the dark gray areas are forest, and the orange and purple patches are all 106 calcareous grassland patches in the study area. *Dianthus carthusianorum* occurred in 65 of them, so we have 65 local populations or demes, and from these about 1600 individuals have been genotyped at 11 microsatellite markers.

Population size varies a lot and we recorded it in four categories for practical purposes: less than 4 individuals means that we can't really do population-level analysis, less than 40 means that we can genotype all individuals, for up to 100 we take a sample but we can still more or less count the individuals, whereas populations with more than 100 individuals are too large to count. We also have patch size in hectares, and this was measured from orthophotos and thus varies continuously. In this species, we expect a relatively strong association between patch size and population size, and here we'll use patch size as a proxy for population size.

Since 1989, most patches have been connected in three shepherding systems. The idea is that large flocks of sheep can help seeds disperse among patches in the same grazing system. However, some patches remained ungrazed as we can see on this map. Also, some patches are grazed only late in the growing season or were only grazed for a few years, we call these

intermittently grazed. Consistently grazed here means that the sheep are herded through 3 - 5 times per year between spring and fall.

We want to test alternative hypotheses of connectivity. The simplest one is isolation-by-distance. We model this with pairwise geographic distances between patches. This pattern is unrelated to the sheep, it could arise either from wind-dispersal of seeds or from pollination. Both of these processes could be affected by intervening forest, so we have a second model that treats forest as a barrier.

The third hypothesis is an isolation-by-resistance model, where the rate of gene flow depends on seed dispersal by sheep. We consider three sub-models: the first assumes that seeds can get from any patch to any other patch within the same shepherding system and distance does not matter. The second assumes that the more patches sheep have to traverse from patch A to B, the less likely it is that a seed disperses from A to B. We thus count the number of patches traversed, and we give an arbitrary high number of 100 to patches outside of the shepherding system to indicate that dispersal is highly unlikely. The last model is almost the same, but it only counts those patches that are grazed consistently as defined before. Each of these competing hypotheses is quantified in a pair-wise distance matrix.

Scene 3: Connectivity Index Si

We want to analyze the data with a neighbourhood approach. Here's an example. We want to quantify connectivity for patch *a*, so this is our focal patch *i*. It has two neighbours, patches *b* and *c*, which can act as source patches *j* for *Dianthus* seeds. The probability of seeds dispersing from each source patch *j* to the focal patch *i* depends on the pairwise ecological distance *dij*, which we can model with any one of the five distance matrices from the previous slide. The predictor X then becomes the integration of seed dispersal from all neighbours, and we can use it to model our response variable, allelic richness.

We use this variant of Hanski's incidence function, where the connectivity 'Si' of focal patch 'i' is modeled as the sum overall its neighbours 'j'. For each neighbour, we multiply to terms. The first is the connectivity model, which is a negative exponential function of the ecological distance 'dij', scaled by a constant dispersal parameter 'alpha' that we can optimize from the data. The second term 'Pj' has to do with the source pool. We used three different variables here. 'Pj' is simply the species occurrence, it is 1 if *Dianthus* is present and thus seeds may be produced, and 0 if the species does not occur. 'Aj' is the area of the source patch, and 'Nj' is the population size category from the previous slide.

We can thus calculate 'Si' for each combination of five distance matrices and three source patch variables. Here's simply a matrix with the correlation between 'Si' and allelic richness, and we see that the second IBR model generally had high correlations, and models with 'Pj' as source patch variable also showed some high correlations, hence the combination of the second IBR model with 'Pj' might be a good model. This kind of comparison can be done more formally with the package 'MuMIn', which stands for multi-model inference. We would calculate a

regression of allelic richness on each of these 15 models, calculate AIC and a model weight for each model, where a higher model weight means that the model is better supported by the data. We'll see how this is done in the second video.

Imagine we replaced the correlations in this table by model weights. Then we could sum the model weights across each row for the connectivity models, and across each column for the source patch variables. These sums of model weights are shown here as a barplot. We see that among the connectivity models, the second and third IBR model are best supported, and among the source patch variables, 'Pj' is best supported. This suggests that genetic diversity is affected by sheep dispersing seeds from patch to patch, and this seems to work as long as the species is present, not only for large source populations.

The analyses on this slide will be included in the course R package as Week 7 Bonus Material.

Scene 4: Back to Regression!

What have we achieved by calculating the connectivity index Si? We started with a hypothesis about between-site characteristics: how the ecological distance between patches affects gene flow. We modeled this in a set of pairwise distance matrices, one for each hypothesis. We could relate these matrices to a matrix of genetic distances, in a Mantel test, partial Mantel test or regression on distance matrices. However, these methods may be affected by spatial autocorrelation in the data.

Here we took a neighbourhood-level approach, where we calculated patch connectivity for each focal patch over all its neighbours. Instead of pairwise distances, we now have one connectivity index value 'Si' for each patch and for each connectivity model. This brings us back into the familiar framework of regression analysis. Here we model a single patch-level response variable, allelic richness, as a function of patch connectivity 'Si', and we can add other variables like patch size that measure at-site characteristics. Now we can use established methods for handling spatial autocorrelation in a regression context. These will be explained in the second video.

Scene 1: Spatial Linear Models

This brings us to the end of the first video. The second video will assume that you are familiar with regression analysis, spatial statistics and linear mixed models that have been covered in the videos for Weeks 4, 5 and 6.

VIDEO 2

Scene 1: Spatial Linear Models

This second video shows how we can handle spatial autocorrelation in linear models, such as regression. We'll look at three different approaches: generalized least squares regression GLS, spatial regression with a simultaneous auto-regressive model, SAR, and spatial filtering with Moran eigenvector maps, MEM. Each method can be implemented in different ways, hence there is a step of model selection. This will be addressed in more detail in Week 12. Finally, we will use a variant of geographically-weighted regression to explore whether the regression results vary across the study area.

Scene 5: Generalized Least Squares (GLS)

In multiple regression, we model a response variable 'Y' as the sum of an intercept, the effect of one or more predictors 'X', and an error term or residual. We estimate the slope coefficient 'b' for each predictor and test whether it is statistically significant. However, these parameter estimates and p-values may not be valid if the residuals are spatially autocorrelated.

We can use the function 'lm.morantest' to test for spatial autocorrelation in the residuals. As with any effect, we should consider the direction, size and statistical significance of the autocorrelation: a positive Moran's I indicates positive spatial autocorrelation, which can inflate p-values in the regression. The magnitude of Moran's I gives us a rough idea of the strength of autocorrelation, as it is similar to a correlation coefficient, and we can compare it between variables measured at the same sampling locations. We should not compare it between studies, though, because different sampling designs have different maximum values that Moran's I could reach. The p-value of the Moran test tells us whether we can rule out chance in creating the observed magnitude of spatial autocorrelation. If the residuals show significant spatial autocorrelation, we want to account for it in the model.

Generalized Least Squares regression GLS uses a geostatistical approach for modeling the spatial autocorrelation in the residuals. We start with an empirical variogram, a plot of the semivariance or pairwise difference between observations against their distance in space. The code to do this is surprisingly simple: we use the function 'Variogram', specify the regression model that we already fitted, and with the argument 'form', we indicate that semivariance should be plotted as a function of the two-dimensional Euclidean distance calculated from the x- and y-coordinates.

Here we see a smooth curve that shows the overall pattern of the point cloud. The variogram has been standardized so that the expected value or sill is 1. We see reduced semivariance at short distances, up to the range. The intercept at zero distance would be around 0.65, that is the nugget effect. The variogram does level off at some point, which suggests that we don't have a large-scale trend in the data and we can go ahead and fit a variogram model to this empirical variogram.

Now we need to take two decisions: whether or not to fit a nugget effect, and which model family to fit. In this case, we definitely want to include a nugget effect. But, which model family would you choose? The choices are: a Rational Quadratic model 'corRatio', a spherical model

'corSpher', an exponential model 'corExp', a Gaussian model 'corGaus', and a linear model 'corLin' that shows a linear increase up to the sill.

What if we don't know? We can simply fit a model for each model family and let R find the best model. We fit each model with the function 'gls', specify the regression model with formula notation and the data set, and with the 'correlation' argument, we specify the model family, how the distances should be calculated, and whether or not to fit a nugget effect.

In this line of pseudo-code, we pass the names of six fitted models to the function 'model.sel', including the original regression model and the gls models fitted with corRatio, corSpher, corExp, corGaus and corLin.

Scene 6: Which Model Fits Best?

So, which of these models fits best? Here's a quick overview of model selection, which will be covered in more depth in Week 12. The function 'model.sel' ranks the models from best to worst. In this case, one model could not be fitted and was removed. The best-ranked model is the one with an exponential distance-decay of the correlation among residuals. For each model, we see the slope estimates. The intercept and the slope for patch size did not vary much, but the slope for IBR is quite a bit lower in this model than for some of the other models.

The models have different degrees of freedom. For each variogram, we had to fit two parameters, hence we used up two additional degrees of freedom compared to the original model. The more parameters we fit, the more likely the model is going to fit the data, hence we use a criterion like AIC to give a penalty for the number of parameters in a model and make the models more comparable.

Here, the criterion used is AICc, which includes a small-sample correction. The model with the smallest AICc is considered best. The delta value shows how much larger the AICc value is for each model compared to the best-ranked model. We see that the next two models have almost the same AICc value, whereas the last two models have much higher AICc values and thus much larger delta values.

This is reflected in the model weights. The weights sum to one, and they indicate the relative support for each model given all models in the set. Thus among these five models, the first three have similar levels of support and we can't really decide between them, whereas the last two have very little support and we can pretty much rule them out.

This is interesting, because the exponential model was ranked best, but the original model with rank 3 came quite close. We should probably go with the exponential model as we can better trust its p-values. This model shows a smaller slope and thus a weaker effect of IBR than the original model.

By the way, in the first video, we saw a barplot of model weights for the different connectivity models and for the source patch variables, now you know how these were calculated!

Scene 7: Spatial Regression with SAR

What if we think that a variogram is not a good way to represent the autocorrelation in our data? We can use so-called spatial regression with a simultaneous auto-regressive term, SAR, that is more compatible with a stepping-stone model of gene flow. Let's summarize what GLS did. GLS quantified the semivariance among any pair of observations as a function of their geographic distance. Internally, this is used to create a matrix of error covariances as some function of distance in space. The error covariance matrix has values of 1 on the diagonal, and the further away two observations are, the lower their covariance, and this distance-decay of covariance is defined by the variogram function that we fitted. Pairs further apart than the range of the variogram have a covariance value of 0, which means that we treat them as independent of each other.

With SAR, we use a different approach to come to the same kind covariance matrix. Here, we first define which pairs of sites are neighbours, then we assign weights to the neighbours. Internally, SAR converts these weights into a covariance matrix that has the same overall characteristics as the one from GLS but obviously will have different values, as these now reflect the neighbour weights.

Could we get the same result as GLS if we treated all pairs as neighbours and used the same distance-decay function to determine the neighbour weights? Interestingly, the answer is no! And this really matters, because it helps understand why we should not include too many neighbours.

Recall this figure from Week 5, where the blue line showed the autocorrelation among first, second, or third neighbours, and the red line shows the partial autocorrelation among second neighbours after accounting for the autocorrelation among first neighbors, and so on. GLS works from the blue line, the observed spatial autocorrelation. SAR works from the red line, assuming that once we account for the autocorrelation among first neighbours, the higher-order autocorrelations are zero. From this red-line model, SAR infers what blue line we would expect to find, and from there it creates the covariance matrix.

So if we defined neighbours based on the blue line, SAR would infer an observed autocorrelation structure more like the green line. This would make the SAR model perform not much better than the original regression model.

In summary, we don't want to include too many neighbours, and the Gabriel graph that we have seen in Week 5 is often a good compromise. Once we have defined the neighbours with the Gabriel graph, we can either give them each a binary weight of 1, or assign weights proportional to the inverse distance, as shown here for two neighbours of site 2. This makes sense if the data were not collected on a regular transect or grid. In either case, we then row-

standardize the weights of all neighbors so that the weights here of all neighbours of site 2 sum to 1.

The neighbour weights are stored in a 'listw' object, see Week 5. We fit a SAR model with the function 'errorsarlm', which takes as arguments the formula of the regression model, dataset, and the 'listw' object with the neighbour weights. Again, we can fit multiple models and select the best one with the function 'model.sel'. Here, the original model is compared to three SAR models based on the Gabriel graph: one with binary weights, one with inverse distance weights, and one with inverse squared distance weights.

Scene 8: Spatial Filtering with MEM

What if we expect a more complex spatial autocorrelation structure, maybe as a result of different processes acting over different spatial and temporal scales that we don't want to model individually? Spatial filtering can be used to capture any spatial structure in the response variable, it does not have to take a specific shape. The idea is then to account for this spatial structure when assessing the effect of our predictors 'X' on the response 'Y'.

The way it works is that we create a set of spatial filters SF that we include in the regression model. Spatial filters are bogus variables that vary in space. To create them, we start with a matrix of spatial weights like we've defined them before, for example with the Gabriel graph. Then we perform eigen analysis to extract the eigenvectors, which are called Moran eigenvector maps, or MEM. These are the bogus variables, and here we see some of them for the *Dianthus* data set plotted in space. The first spatial eigenvector shows the largest scale pattern and the scale of the pattern becomes smaller from one eigenvector to the next. In this example, there is a lot of East-West variation, so let's plot a moving average of each eigenvector as a function of the x-coordinate to illustrate those patterns. From West to East, the first eigenvector, 'sf1', shows a steady decrease. The second eigenvector, 'sf2', shows a more Gaussian pattern, 'sf6' shows a finer-scale periodicity, and 'sf17' has three peaks along the same distance.

Each pattern in itself is not meaningful, but each may capture some of the spatial variance in the allelic richness. We then use a stepwise selection algorithm to select those spatial eigenvectors that explain a significant portion of the spatial variation in allelic richness. These are then included as predictors in the regression model.

In multiple regression, the slope estimates and p-values that we see are estimated after accounting for all other predictors in the model. This means that when we include the set of spatial filters in the model, the effects of our predictors 'X' are estimated after accounting for the set of significant spatial eigenvectors SF. In essence, we partial out all spatially structured variance in the response and fit the predictors 'X' to the remaining non-spatial variation in allelic richness.

This is illustrated here: with the set of significant spatial eigenvectors, we predict these values on the left. Compared to the observed values of allelic richness on the right, this captures already a large part of the variation. Patch size and connectivity are then used to explain the remaining difference between the values on the left and on the right.

How do we implement this? The new package 'spmoran' makes it really easy. The function 'meigen' takes as argument either the spatial coordinates or a spatial weights matrix 'cmat' and extracts the spatial eigenvectors and their eigenvalues. The function 'esf' does the rest: it performs stepwise selection of spatial eigenvectors with a method defined by the 'fn' argument, here an option that is based on R-squared, and fit the model. There are two ways of fitting the model: the function 'esf' treats the spatial eigenvectors as fixed effects, the function 'resf' treats them as random effects and fits a linear mixed model like the ones we have seen in Week 6.

Scene 9: Which Method to Choose?

We've now seen three methods that do the same thing in different ways. Which one should you choose? This depends on what you want to build into your null model. Do you only want to account for isolation by distance, or for any spatial constraints on gene flow, which could be the result of isolation by distance, barriers, resistance, or any combination of them?

In the latter case, spatial filtering with MEM is appropriate. An example would be if you want to test the association between a SNP locus and some environmental gradient that you expect to create selection.

If you only want to account for isolation by distance, then the decision depends on your model of gene flow. If a stepping stone model is appropriate then I would choose spatial regression with SAR. This is the case in our example today, where the species occurs in discrete grassland patches and there presumably are no un-sampled populations within the study area.

For a random sample of individuals from a continuously distributed species, or if a random subset of populations in the study area have been sampled, I would use generalized least squares regression GLS.

So far so good. Once we have accounted for spatial autocorrelation in our data, we want to interpret the model, especially the slope parameter for each predictor 'X'. The final slide today falls into the category of amazing things I did not know I could do in R.

Scene 9: Spatially Varying Coefficients

With geographically weighted regression, we can see how the slope estimates vary across the study area. This means that we can check whether the relationship that we found holds across the study area. In practical terms, this could mean here whether connectivity by shepherding is

effective in restoring gene flow among grazed patches overall or whether there are specific areas where this is not the case?

The method we use here is called a spatially varying coefficients model, SVC, and we fit it almost the same way as we fitted the last model. The function 'resf_vc' of the package 'spmoran' fits a varying coefficients model to the spatial filtering model with random effects. Here, 'x' is the set of predictors whose slope parameters we allow to vary, and 'xconst' is an optional argument to include predictors whose slope should be kept constant across the study area, this could be a covariate we want to control for.

What does the method do? It uses a local neighbourhood around each site to estimate a local regression slope for each predictor. The map here shows how the slope for the IBR connectivity index varies across the study area. The larger the symbol, the larger the slope. Bluish symbols mark statistically significant slopes, and red ones are not significantly different from zero.
We also get a numerical summary of the distribution of parameter estimates, here on the left for the intercept and on the right the slope for the only predictor in this model, the IBR connectivity index. The lowest slope estimate was negative 0.2, the highest plus 0.4. The median was 0.25, which means that 50 % of sites had a slope estimate of 0.25 or higher.

Across most of the study area, the slope estimates seem quite constant. The smaller and often non-significant values occurred clustered in specific areas, including two valleys in the East and Southeast and two groups of sites at either side of the mouth of the valley in the West. These indicate areas where allelic richness is less associated with connectivity by shepherding. This map can help us generate new hypotheses. In the Southeast, for example, this could indicate a lack of seed dispersal, whereas in the West, this might reflect historic connectivity across the valley, or pollen-mediated gene flow as these sites are not far apart, although they belong to two different shepherding systems.

Scene 1: Spatial Linear Models

This brings us to the end of this week's two-part video. As you probably have noticed, we've relied quite heavily on topics covered in previous weeks: Regression in Week 4, Spatial statistics in Week 5, and Linear mixed models in Week 6, and you may want to brush up some of these as needed. Then you should be well prepared to move on to the worked example. The interactive tutorial will let you practice some related R skills so that you can tweak R code from this and future worked examples.