LPDynR: a new tool to calculate the Land Productivity Dynamics indicator. Supplementary Material

Xavier Rotllan-Puig[1, ✉], Eva Ivits[2], and Michael Cherlet[3]

[1] ASTER Projects. Barri Reboll, 9, 1r. 08694 Guardiola de Berguedà (Barcelona), SPAIN

[2] European Environment Agency. Geospatial Information Services Group. Copenhagen, DENMARK

[3] European Commission – Joint Research Centre (JRC). Directorate D – Sustainable Resources. Unit D6 – Knowledge for Sustainable Development & Food Security Unit. Via Enrico Fermi 2749. I-21027 Ispra (VA), ITALY

✉ Correspondence: Xavier Rotllan-Puig <xavier.rotllan.puig@aster-projects.cat>

Supplementary Material S1: Assessing the need of using the "PCAs step" to derive the Ecosystem Functional Types

For clustering exercises, usually it is necessary to standardise the data set, especially when variables have different units/scales. However, cases can be found often where this is avoided. This is why we checked whether the Ecosystem Functional Types (EFT) calculated with the variables derived from the PCAs step gave better results than when using the raw variables. We did that both by (1) analysing an evaluator of clustering performance and (2) by checking differences in the Land Productivity Dynamics (LPD) final map.

Analysing clustering performance

We included an index in the function *EFT_clust()* so that the user can have an idea of each clustering's performance. The index (i.e. compactness index or CI) is calculated by the ratio of *betweenss* (i.e. between-cluster sum of squares, or BSS) and *totss* (i.e. total sum of squares, or TSS), in percentage. It measures the compactness of the individuals (i.e. pixels) within the groups and it is expected to be as high as possible. In addition, the function implemented for the clustering (*stats::kmeans*()) accepts an argument to set the number of times (*n*) that the clustering is run using a different set of starting centroids. For each of these runs, the function makes an evaluation of the results and finally choose the best model. Therefore, increasing this *n* maximises the clustering performance.

We made the tests by setting *n* to 5, 10 and 15 runs and the results can be seen in Table S1.1. It can be seen how the clustering performed better when using the raw variables and that the process stabilized at 10 runs

*Table S1.1: Comparison of the compactness index (CI = (BSS/TSS)*100) when using the variables derived from the PCAs step versus when using the raw variables*

Number of runs (*n*)	CI using PCAs	CI using raw variables
5	82.77298	87.95956
10	82.77297	87.98955
15	82.77296	87.98984

Analysing Land Productivity Dynamics final map

For the LPD final map evaluation, it must be kept in mind that each clustering starts with a random selection of the starting centroids and, even if the process converges, the final EFTs can be slightly different. This fact could make that the final LPD maps showed some differences for this reason more than for a bad performance of one the methods (i.e. the variables used).

Figure S1.1 shows the LPD final maps produced using the EFTs derived with the "PCAs variables" (Map1) and with the raw variables (Map2), and also a "differences map" (Map1 - Map2). Summarising the number of pixels (Table S1.2), we saw that less than 4% of them changed only 1 LPD class (either positively or negatively), but not a single pixel changed 2 or more classes. Moreover, nearly none changed among classes 2 and 3 (either way). This means that that 4% of pixels only get changed inside its own positive or negative dynamics (i.e. either between declining to early signs of decline, or between increasing and stable but not stressed).

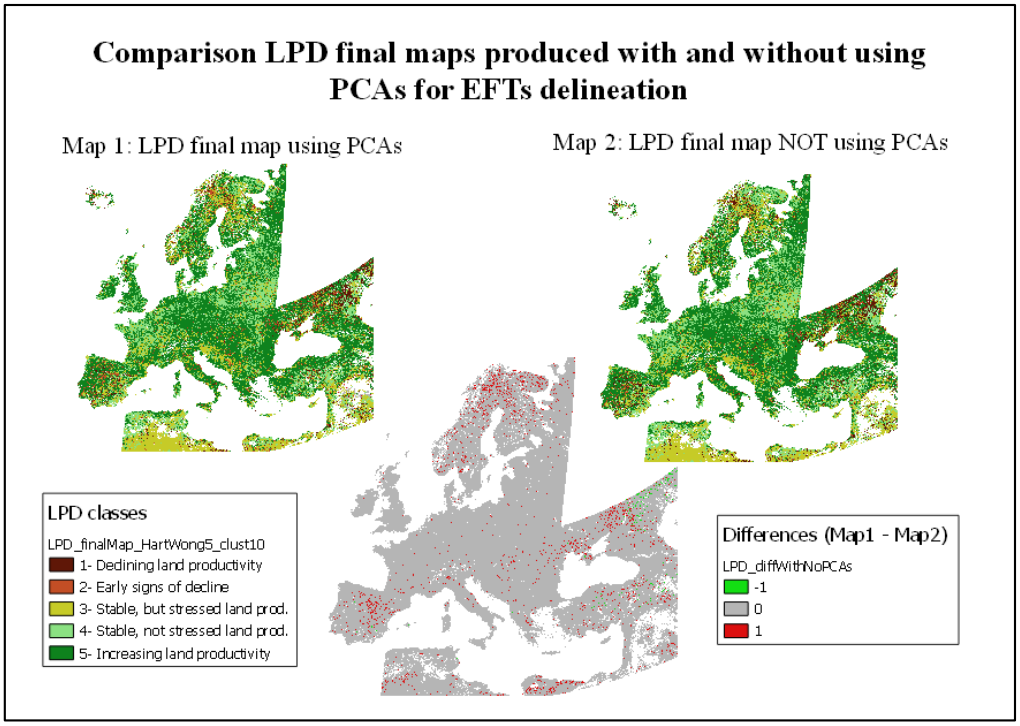

Figure S1.1: Comparison of Land Productivity Dynamics final maps produced using the Ecosystem Functional Types derived with the "PCAs variables" (Map1) and with the raw variables (Map2), and also a "differences map" (Map1 - Map2)

Table S1.2: Number and proportion of pixels which showed differences in the final Land Productivity map when using the Ecosystem Functional Types calculated with the variables derived with the PCA step and with the raw variables. LPD classes correspond to those in Figure S1.1

LPD class using PCAs	LPD class using raw variables	Number of pixels	% of total pixels
1	2	51565	0.14
2	1	699324	1.87
2	3	12	<0.0001
3	2	38	<0.0001
4	5	78660	0.21
5	4	644403	1.73

Conclusions

According to both analysis (i.e. evaluation of clustering performance and differences in the Land Productivity Dynamics final map, there are not many differences when using one or the other approach, therefore, the simplest approach should be used. In this case, then, the Ecosystem Functional Types should be calculated using the raw variables for the clustering process.

Supplementary Material S2: Multicollinearity of variables

In order to check and remove multicollinearity among the phenological/productivity variables, *LPDynR* uses the function *rm_multicol()*. This function allows the user to set up the minimum Pearson's correlation absolute value |r|, which can be modified by passing the argument *multicol_cutoff*. In the case study presented in the main article, it was established to be r = 0.7. Figure S2.1 shows the dendrogram produced by the function, in which it can be seen the three groups of correlation that were found.

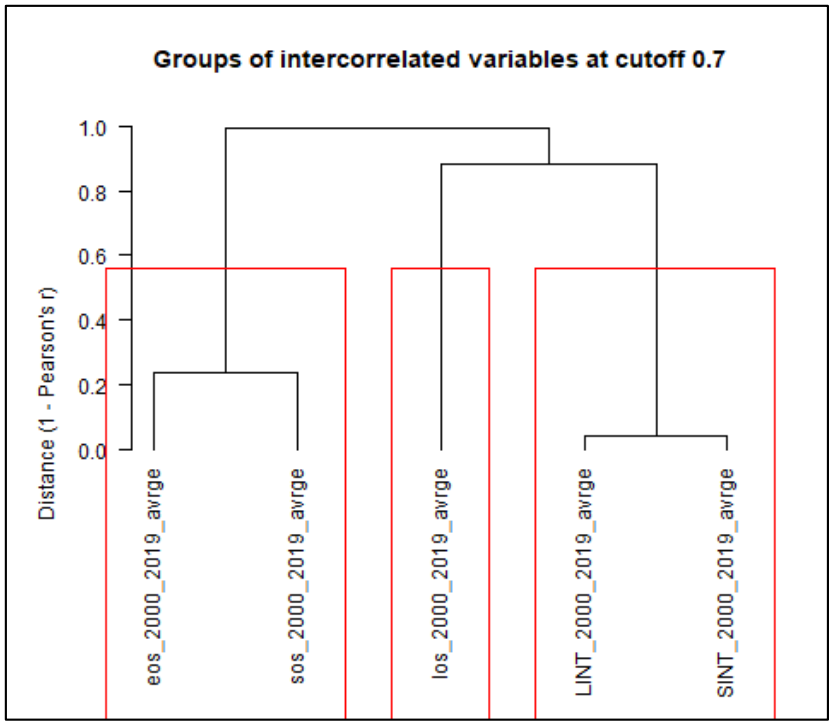

Figure S2.1. Dendrogram showing the groups of correlated variables

Supplementary Material S3: "Scree plot" to determine optimal number of clusters

One of the main limitations of K-means for clustering is that this method does not optimize the number of clusters by itself, but they need to be determined by the user. There are several methods to calculate the optimal number of clusters, such as the "scree plot", numerical methods, etc.

In the package LPDynR, the function *clust_optim()* produces a "scree plot" after running several K-means clustering with different number of clusters each, in order to assess how the quality of the models change with the number of clusters. In the case study, clustering was run with ten different number of clusters (5 to 50, with the increment of 5) to give a good amount of points to plot the curve, and the maximum number of iterations was set to 10. Figure S3.1 shows how after 15 clusters the total within-cluster sum of squares (i.e. TSS) does not improve substantially when increasing the number of clusters.

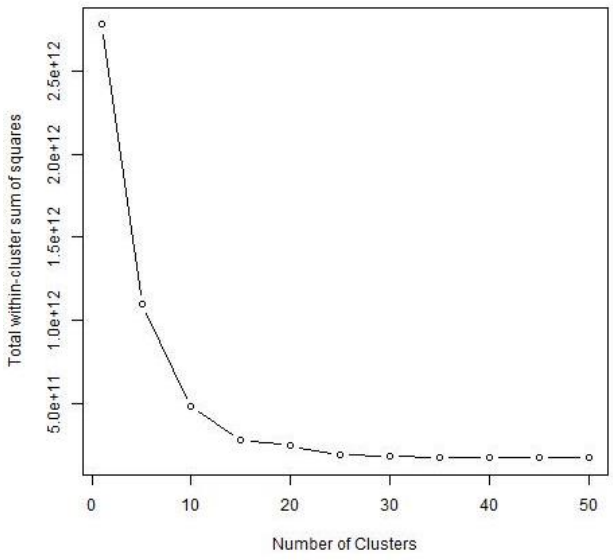

Figure S3.1. "Scree plot" method used to calculate the optimal number of clusters. The "elbow" indicates where the quality of the model (total within'cluster sum of squares, or TSS) no longer improves substantially as the number of clusters (model complexity) increases

Supplementary Material S4: Assessing number of iterations for Ecosystem Functional Types clustering

The Ecosystem Functional Types (EFTs) are calculated by means of a K-means clustering. We assessed how many iterations were necessary for the clustering to converge. To do that, we run the function *LPDynR::EFT_clust()* with the same rounded data set, algorithm and number of clusters, but with different number of iterations (i.e. 10, 50, 100, 250 and 500), 10 times each. The results (Table S4.1) showed that with 500 iterations the process achieved convergence with no issues (i.e. without exceeding execution time limits self-established by the algorithm) more often than the others, while for 10 iterations the convergence failed several times. However, clustering performance calculated with the compactness index (CI = (BSS/TSS)*100) seemed to be very similar far all the cases.

*Table S4.1: Number of clustering processes (repetitions) in which k-means converged within certain amount of iterations. "With issues" stands for processes which achieve convergence but the algorithm warned because of an excess of the limit of processing time. Compactness index (CI = (BSS/TSS)*100) measures clustering performance*

Number of iterations	Total number of repetitions	Total times converged (with issues)	CI
10	10	8(3)	87.98985
50	10	10(6)	87.98568
100	10	10(7)	87.98568
250	10	10(4)	87.98568
500	10	10(3)	87.98568