

# Comparison of silhouette-based reallocation methods for vegetation classification

Attila Lengyel<sup>1</sup>  | David W. Roberts<sup>2,3</sup>  | Zoltán Botta-Dukát<sup>1</sup> 

<sup>1</sup>Centre for Ecological Research, Institute of Ecology and Botany, Vácrátót, Hungary

<sup>2</sup>Swiss Federal Research Institute WSL, Birmensdorf, Switzerland

<sup>3</sup>Ecology Department, Montana State University, Bozeman, MT, USA

## Correspondence

Attila Lengyel, Centre for Ecological Research, Institute of Ecology and Botany, Alkotmány u. 2-4. H-2163 Vácrátót, Hungary.

Email: lengyel.attila@ecolres.hu

## Funding information

The work of AL was supported by the National Research, Development and Innovation Office, Hungary, Grant/Award Number: PD123997

Co-ordinating Editor: Otto Wildi

## Abstract

**Aims:** Vegetation classification seeks to partition the variability of vegetation into relatively homogeneous but distinct types. There are many ways to evaluate, and potentially improve, such a partitioning. One effective approach involves calculating silhouette widths which measure the goodness-of-fit of plots to their cluster. We introduce a new iterative reallocation clustering method – Reallocation of Misclassified Objects based on Silhouette width (REMOS) – and compare its performance with an existing algorithm – OPTimizing SILhouette widths (OPTSIL). REMOS reallocates misclassified objects to their nearest-neighbour cluster iteratively. Of its two variants, REMOS1 reallocates only the object with the lowest silhouette width, while REMOS2 reallocates all objects with negative silhouette width in each iteration. We test how REMOS1, REMOS2 and OPTSIL perform in terms of: (a) cluster homogeneity and separation; (b) the number of diagnostic species; and (c) runtime.

**Methods:** We classified simulated data with the flexible-beta algorithm for values of beta from  $-1$  to  $0$ . These classifications were subsequently optimized by REMOS1, REMOS2 and OPTSIL and compared for mean silhouette widths, misclassification rate, and runtime. We classified three vegetation data sets from two to ten clusters, optimized all outcomes with the three reallocation methods, and compared their mean silhouette widths, misclassification rate, and number of diagnostic species.

**Results:** OPTSIL achieved the highest mean silhouette width across the majority of the data sets. REMOS achieved zero or negligible misclassifications, outperforming OPTSIL on this criterion. REMOS algorithms were typically more than an order of magnitude faster to calculate than OPTSIL. There was no clear difference between REMOS and OPTSIL in the number of diagnostic species.

**Conclusions:** REMOS algorithms may be preferable to OPTSIL when: (a) the primary objective is to reduce the number of negative silhouette widths in a classification, as opposed to maximizing mean silhouette width; or (b) when the time efficiency of the algorithm is important.

## KEYWORDS

classification, clustering, flexible-beta, iterative, OPTIMCLASS, optimization, OPTSIL, REMOS, silhouette, validation

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2021 The Authors. *Journal of Vegetation Science* published by John Wiley & Sons Ltd on behalf of International Association for Vegetation Science.

## 1 | INTRODUCTION

Numerical classification methods are essential data analytical tools in vegetation ecology (De Cáceres et al., 2015) and several other scientific fields, including genomics, psychology, or sociology. Basically, classification algorithms can be divided into two groups. Hierarchical algorithms produce a perfectly nested hierarchy of clusters of objects, while the output of non-hierarchical methods is a partition in which each classified object is assigned exclusively to one cluster (or, in the special case of fuzzy clustering methods, non-exclusively to several clusters using fuzzy membership weights) at a single level. Hierarchical methods can be subdivided into agglomerative and divisive methods based on whether they initiate the clustering algorithm by treating each single object as a separate cluster, and then merge them until all objects are included in a single cluster at the highest hierarchical level, or they proceed in the opposite direction by dividing the entire sample iteratively into smaller and smaller subsets in a nested way. The diversity of numerical classification methods is reviewed by several authors, e.g. Kaufman and Rousseeuw (1990), Podani (2000), Legendre and Legendre (2012), Peet and Roberts (2013).

An advantage of hierarchical methods is that they do not need a pre-defined number of clusters; however, if a partition is the objective, as is commonly the case, a hierarchical classification requires a post-hoc assessment for choosing the “best” number of clusters. Moreover, a disadvantage of hierarchical methods is that earlier steps (either merging or division) constrain subsequent ones, hence the final solution may be suboptimal. In such a case the a posteriori reallocation of misclassified objects might be advantageous. A simplistic, more or less general workflow of reallocation-based algorithms can be summarized in the following basic steps (Podani, 2000 cit. Hartigan, 1975, Therrien, 1989). First, the initial partition is assessed by an appropriate index of “goodness” (also called a *cluster validity measure*). Second, an algorithm seeks for possible changes in the partition with which the index value can be increased (or decreased depending on scaling). If improvement is possible, the partition is changed accordingly. Then, the goodness of the updated partition is re-assessed, and new changes bringing further improvement are sought for. These steps are repeated until no further improvement is possible. In this way, reallocation-based methods optimize (i.e. minimize or maximize) the value of a particular index of goodness through iteratively changing the initial clustering.

The assessment of cluster validity can be approached from two perspectives, which Aho et al. (2008) distinguished as internal and external evaluators (i.e. indices of “goodness”). Internal evaluators are based on the species abundance or occurrence data, or distance/dissimilarity matrices derived from those data, whereas external evaluators are based on data not employed in the calculations, generally including environmental variables associated with the sample units. Because our focus is on cluster homogeneity and distinctness we emphasize internal evaluation based on pairwise sample unit dissimilarity.

Different reallocation algorithms may optimize different cluster validity measures. The widely known *k*-means method, for instance, minimizes the sum of squared Euclidean distances of objects from the centroid of their respective cluster. A similarly common alternative called *k*-medoid or partitioning around medoids method (Kaufman and Rousseeuw, 1987) minimizes average dissimilarity between each object and the central object (that is, the medoid) of their respective cluster. These two methods are successful in recognizing clusters with similar size and diameter. However, both sum of squared distances and mean dissimilarity provide limited information about the partition, and are thus minimally useful validity measures at the stage of in-depth evaluation of the classification result. In their basic form, they disregard the fit of individual sample units in the partition. They are not standardized measures and, therefore, are incomparable across different data sets. As a consequence, partitions resulting from *k*-means or *k*-medoid methods need to be assessed by validity criteria different from those which had been optimized. This might be disadvantageous if the criterion used for creating the partition and the one used for the detailed assessment are sensitive to different properties of the partition. Optimizing standardized validity criteria with broader applicability instead of sum of squared distances and mean dissimilarity seems straightforward. Such a criterion is the silhouette width index (Rousseeuw, 1987, Kaufman and Rousseeuw, 1990). Let *i* be a sample unit belonging to cluster *A*. Let *C* be a cluster not containing *i*. Then *a*(*i*) is defined as the average dissimilarity between *i* and all other sample units in *A*, while *c*(*i*,*C*) is the average dissimilarity between *i* and all sample units in *C*.

$$b(i) = \min\{c(i, C)\}, C \neq A$$

That is, *b*(*i*) is the average dissimilarity between *i* and the members of its closest neighbour cluster. The silhouette width, *S*(*i*), is defined as:

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

*S*(*i*) ranges between -1 and +1. Values near +1 indicate that sample unit *i* is much closer to other sample units in its assigned cluster than to sample units of the closest other cluster, implying a correct classification. If *S*(*i*) is near 0, the correct classification of the focal sample unit is uncertain, thus suggesting intermediate position between two clusters. *S*(*i*) values < 0 indicate poor fit, and such objects are often considered “misclassified” (Rousseeuw, 1987). Scaling up from the level of sample units, values of silhouette width can be averaged over individual clusters, or the entire sample, thus providing information about goodness of clusters separately, or the entire partition.

Roberts (2015) introduced two reallocation-based methods which can be used for improving already existing classifications. One of these two, called OPTSIL (for OPTimizing SILhouette widths), optimizes the mean overall silhouette width. In each iteration, OPTSIL evaluates how much the reallocation of any single



sample unit in the classification increases the sample-wise mean of silhouette width. It is done by re-assigning each sample unit from its current cluster to every other cluster, and then re-calculating the silhouette widths for all sample units. The reallocation which causes the highest increase in the sample-wise mean silhouette is accepted at each step, until no further improvement is possible. Because OPTSIL employs an anticipatory algorithm that calculates the consequences of reallocation before revising the classification, the optimization criterion is strictly monotonic increasing. Roberts (2015) concluded that OPTSIL is able to significantly improve an initial classification; however, it is slow to converge, and thus recommended for “polishing” of classifications made by other methods. R code for OPTSIL is provided in the R package *optpart* (Roberts, 2016).

In this paper, we present two new silhouette-based reallocation algorithms, called REMOS (REallocation of Misclassified Objects based on Silhouette width) version 1 and 2, which minimize the number of misclassified objects instead of the mean silhouette width. Using artificial and real data sets, we compare them with OPTSIL in terms of three criteria: optimization success, time efficiency, and interpretability.

## 2 | METHODS

### 2.1 | The REMOS algorithms

REMOS begins with an existing classification by calculating the silhouette width of every sample unit to its assigned cluster. It then identifies all sample units that have a negative silhouette width and reallocates them to the nearest-neighbour cluster that defines their silhouette width. Reallocation of a single object means moving it from its current cluster to a different target cluster. The target cluster is the nearest cluster to the object other than its current cluster. REMOS versions 1 and 2 differ in the number of reassignments made in each iteration. REMOS1 reallocates only the sample unit with the most negative silhouette width (i.e. the “worst classified” sample unit), while REMOS2 reallocates all sample units with negative silhouette width (i.e. all misclassified objects). Both algorithms stop if the lowest silhouette width reaches or exceeds a threshold  $L$ , or if no further improvement is possible. By default  $L$  is 0; i.e. no silhouette widths are negative. However, using different values between  $-1$  and  $0$  can control tolerance towards misclassifications. The steps of the algorithms are presented below:

1. Calculate the silhouette widths,  $S(i)$ , for the classified objects.
2. Are there any objects with  $S(i) < L$ ?
  - 2a. If no, then go to (5).
  - 2b. If yes, go to (3).
3. Update the classification by reallocating objects:
  - REMOS1: reallocate only the object with the most negative silhouette width to its neighbour cluster.

REMOS2: reallocate all the objects with  $S(i) < L$  to their respective neighbour clusters.

4. Go to (1).
5. End – no further optimization is possible.

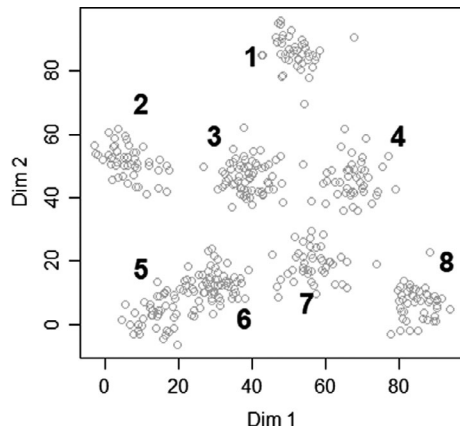
Preliminary analyses (not shown) demonstrated that both REMOS algorithms frequently converge into loops where the algorithm iterates repeatedly over a finite number of suboptimal solutions without finding any of them as a final solution. To break such a loop, the algorithm checks for repetitions and stops if two identical solutions occur. In this case, the solution with the lowest number of negative silhouette widths is selected from the previous iterations. In case of tied minimum negative silhouette widths, the solution giving the lowest absolute sum of negative silhouette widths (a surrogate for smaller “classification error”) is chosen as final. While in general this looping behaviour is undesirable it may help avoid local optima in a manner similar to genetic algorithms.

Not surprisingly, in most cases REMOS1 requires many more iterations than REMOS2. According to our pilot analyses with differently sized data matrices and different initial classifications, this can extend the computation time of REMOS1 in comparison with REMOS2. It is possible to set an upper limit to the number of iterations; however, as there is no standard value for this threshold, the default setting is infinity (that is, no limit).

An R script of the REMOS algorithms is provided in Appendix S1.

### 2.2 | Data sets

We compared the performance of the REMOS1, REMOS2 and the OPTSIL algorithms on three real and one artificial data set. The Shoshone data set is a random subset comprising 150 plots of 375 m<sup>2</sup> selected from a larger forest inventory database. This data set contains 368 species and represents coniferous forests of Shoshone National Forests (WY, USA). The Bryce data set was sampled in the Bryce Canyon National Park (UT, USA; Roberts, 1992). It includes 160 plots of ~400 m<sup>2</sup> (0.1 acre) where the presence of 169 vascular plant species (except trees) was recorded. The Grasslands data set is a subset of a larger sample of mesic grasslands of northern Hungary with 25 m<sup>2</sup> plot area (Lengyel et al., 2016). The size of the matrix is 55 plots by 269 species. For all real data sets, only the presence or absence of species was considered. For artificial data, we employed a simulated data set of 400 points in two dimensions. To generate this point pattern, in the first step, eight centroids were randomly selected around which point aggregations were to be prepared. Each of the 400 points was assigned randomly to one of the centroids. Then, the exact coordinate of each point was drawn from a two-dimensional normal distribution where the centroid was the respective group centroid and standard deviation was 5. As a result, points were aggregated into eight diffuse clusters, where sizes of the aggregations were similar, although unequal (Figure 1). The R code used for generating the point pattern is available in Appendix S2. For different test scenarios, random subsets of different size were used.



**FIGURE 1** The simulated data set containing 400 points in eight aggregations

## 2.3 | Data analysis

The performance of the REMOS and OPTSIL algorithms was evaluated from three aspects: optimization success on different initial classifications of artificial and real data, interpretability of the optimized classification of real data based on diagnostic (faithful) species, and dependence of computation time on sample size with artificial data.

For testing optimization success, initial classifications of random subsamples of the artificial data set containing 200 points were prepared using the flexible-beta classification algorithm (Lance and Williams, 1966) on the Euclidean distance matrix among sample points in the subsample. This method uses a parameter called beta which enables producing classifications with different degrees of “chaining” vs “grouping.” The beta is adjustable between  $-1$  and  $+1$ , with lower values emphasizing grouping (with a similar logic to the complete linkage method) while higher values emphasize chaining (similarly to single linkage). With beta near 0 flexible-beta clustering shows an intermediate behaviour, similar to the average linkage method (UPGMA). Several authors reported that the flexible-beta clustering method provides the most satisfactory classifications using  $\beta = -0.25$ . In practice, values  $> 0.0$  are rarely employed, and we focused our analyses on beta in the range between  $-1$  and  $0$  in steps by  $0.25$ . The hierarchical classifications were cut at the known correct eight-cluster level. The procedure was repeated on independent subsamples five times resulting in  $5 \times 9 = 45$  initial classifications. Each of them was optimized using the REMOS1, REMOS2, and OPTSIL algorithms. We compared the change of mean silhouette widths (MSW) and misclassification rate (the proportion of negative silhouette widths; MR) across beta values between the optimized classifications and the initial classification. In Appendix S4, we show some exemplary classifications.

For comparing time efficiency, we drew subsamples containing 50, 100, 200, and 300 points of the artificial data set in 20 replicates, and additionally, used also the entire sample of 400 points. Each of them was classified to eight clusters using the flexible-beta algorithm with

$\beta = -0.25$ , resulting in 81 initial classifications. These were optimized using REMOS1, REMOS2, and OPTSIL, and the time elapsed during the optimization process was compared among the three algorithms. The analyses were run on a virtual machine with 8 GB RAM, four CPU cores (virtualized by two Intel Xeon E5-2620 0 @ 2.00GHz processors by the host computer) and 312 GB HDD.

The real data sets were classified from two to twenty clusters using the flexible-beta algorithm (Lance and Williams, 1966) with  $\beta = -0.25$ . In all cases the dissimilarity measure was the Sørensen index (Legendre and Legendre, 2012). Each partition was optimized using the REMOS1, REMOS2, and OPTSIL methods. To assess differences in optimization success, mean silhouette width and misclassification rate were calculated and compared among reallocation methods, the original classification, and across numbers of clusters.

Lötter et al. (2013) argued that species fidelity should be a leading criterion in the evaluation of vegetation classifications. Therefore, we used the Optimclass1 index – the total number of faithful species across all clusters (Tichý et al., 2010) – as a proxy for interpretability of classifications. Faithful species were determined testing the null hypothesis that the species shows random distribution across clusters by Fisher's exact test at  $\alpha = 0.001$  significance level (Chytrý et al., 2002). Hence, we compared flexible-beta classifications optimized by REMOS1, REMOS2, and OPTSIL, as well as the initial flexible-beta classifications, in terms of the number of faithful species for each cluster solution across the number of clusters.

The data analysis was carried out in the R software environment (R Core Team, 2017) using the *vegan* (Oksanen et al., 2019) and the *cluster* (Maechler et al., 2018) packages. Source code for REMOS1 and REMOS2 is supplied in Appendix S3. OPTSIL was calculated using the *optpart* package (Roberts, 2016).

## 3 | RESULTS

### 3.1 | Simulated data

In all cases, the silhouette widths of the initial flexible-beta classifications were moderately high (Table 1, Figure 2). Across the five levels of beta, OPTSIL exhibited higher mean silhouette widths than REMOS1 and REMOS2; however, there was a large overlap between methods. REMOS1 exhibited the lowest misclassification rates, followed by REMOS2 and then OPTSIL (Table 1, Figure 3). All classifications mirrored the a priori point aggregations efficiently (Figures S4-1 to Figure S4-4). Classifications differed mostly in the assignments of transitional points.

There was a significant difference in computation time among the three optimization methods for the larger data sets (Figure 4). As expected REMOS2 was generally the fastest, with runtimes between 0.0006 s and 0.0010 s. REMOS1 was nearly as efficient (run times from 0.0015 s to 0.0030 s), while OPTSIL was slower by several orders of magnitude (0.2556 s to 24.4539 s).

### 3.2 | Real data

On the Grasslands data set, OPTSIL reached the highest MSW at all but two examined cluster sizes (Figure 5). At six and ten clusters, REMOS1 performed the best and was only slightly worse than OPTSIL in all other cases. Interestingly, REMOS2 gave the same MSW values as REMOS1 with two to five clusters (likely due to identical final solutions), but at finer resolutions it was much poorer. With six, seven, and nine clusters REMOS2 even decreased

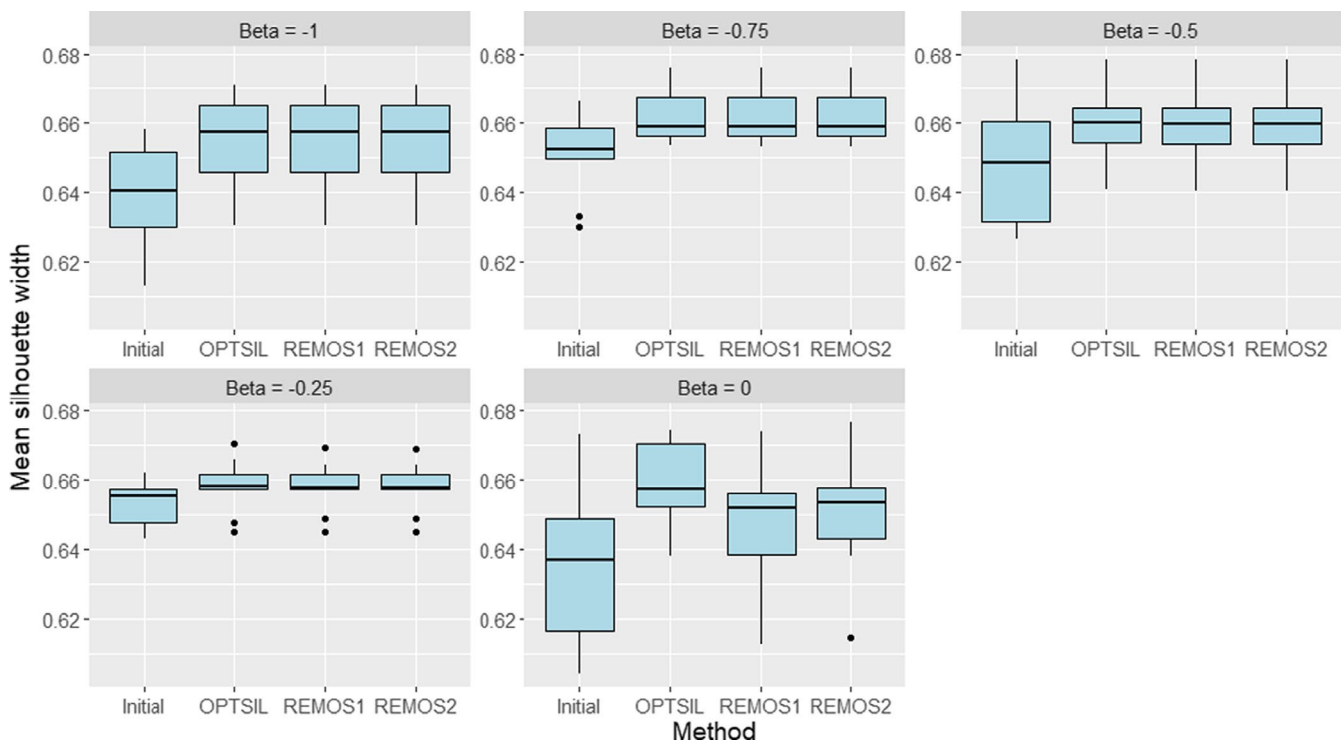
**TABLE 1** Minimum, median and maximum mean silhouette widths and misclassification rates of the initial (flexible-beta), OPTSIL, REMOS1, and REMOS2 classifications

	Initial (flexible-beta)	OPTSIL	REMOS1	REMOS2
Mean silhouette widths				
Minimum	0.6040	0.6302	0.6125	0.6143
Median	0.6496	0.6580	0.6573	0.6574
Maximum	0.6783	0.6783	0.6783	0.6783
Misclassification rate				
Minimum	0.0000	0.0000	0.0000	0.0000
Median	0.0150	0.0025	0.0000	0.0000
Maximum	0.0500	0.0200	0.0050	0.0100

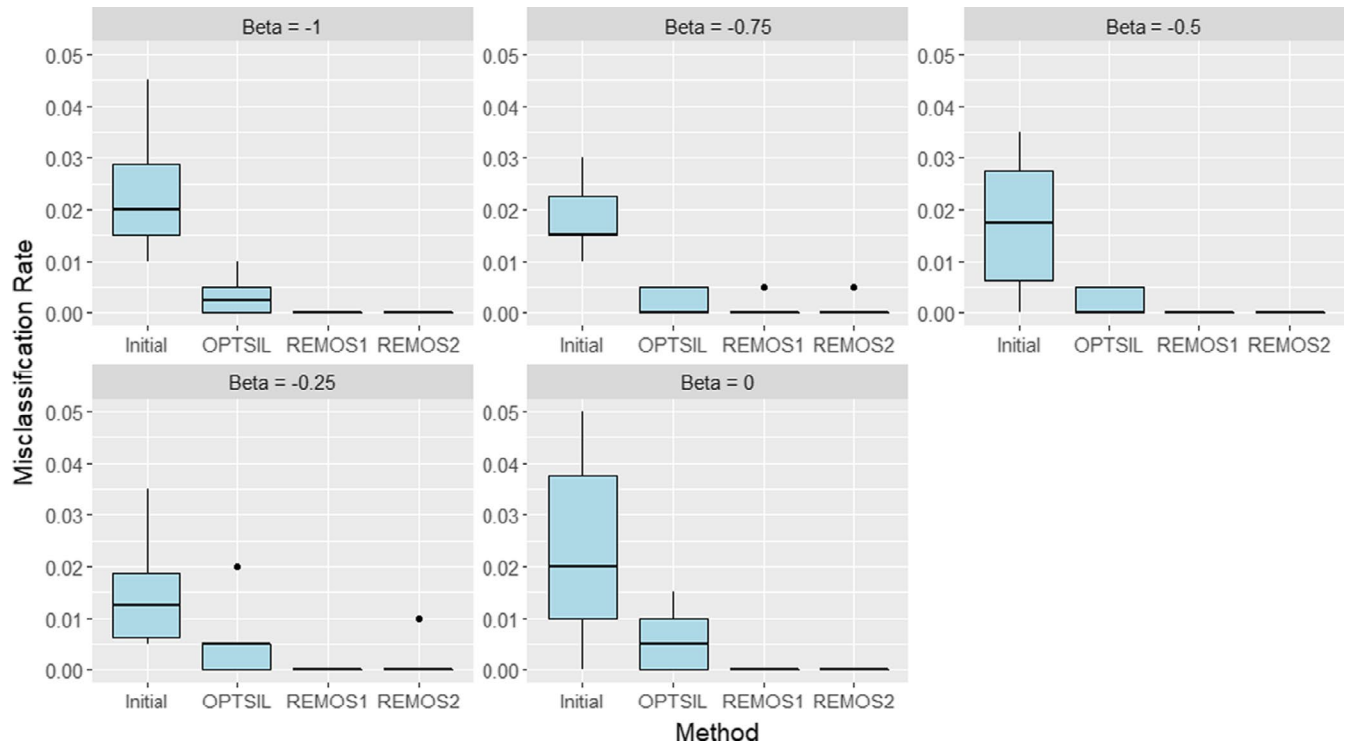
OPTSIL, OPTimizing SILhouette width; REMOS, REallocation of Misclassified Objects based on Silhouette width.

the MSW of the initial classification. REMOS1 achieved the lowest misclassification rate with no negative silhouette width values over all cluster solutions. As with MSW, from two to five clusters REMOS2 gave the same result, but the weak performance with six or more clusters was visible here, too. OPTSIL solutions ranked in intermediate position between REMOS1 and the initial classification, the latter being the worst in all but two cases. Between two to seven clusters all methods (including the initial classification) showed similar numbers of diagnostic species, while at finer resolutions REMOS2 was the best. Nevertheless, the Grassland data set is small, thus at this level the sizes of clusters are so small and the number of diagnostic species so low that these differences are probably insignificant.

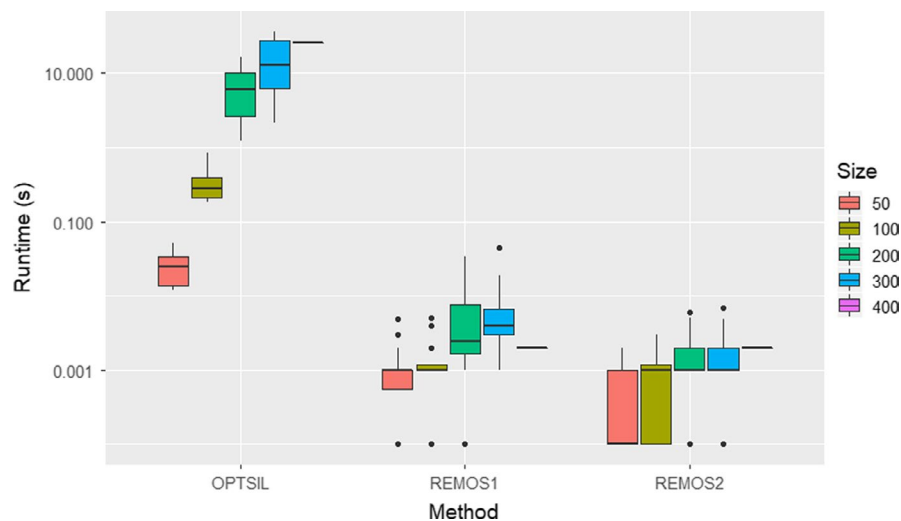
With the Bryce data set OPTSIL produced the highest MSW for all but the two-cluster solution (Figure 6). REMOS1 and REMOS2 had very similar, often identical performance. With a minimal difference they outperformed OPTSIL at two clusters. At three and four clusters they were slightly worse than OPTSIL but this difference increased with the number of clusters, and became striking for seven and more clusters. The initial classification had the lowest MSW across the tested numbers of clusters. REMOS1 and REMOS2 provided solutions with the lowest MR, most often with no negative silhouette widths at all. OPTSIL had MR between 0.02 and 0.07, while the initial classification had the highest MR in at all cluster numbers (MR between 0.048 and 0.15). OPTSIL performed the best in terms of diagnostic species at three, as well as at six and more clusters. Interestingly, at four clusters the initial classification had the most



**FIGURE 2** Differences in mean silhouette width between the initial classification (without optimization), REMOS1, REMOS2, and OPTSIL across different beta values of the flexible-beta classification. OPTSIL, OPTimizing SILhouette width; REMOS, REallocation of Misclassified Objects based on Silhouette width



**FIGURE 3** Differences in misclassification rate between the initial classification (without optimization), REMOS1, REMOS2, and OPTSIL across different beta values of the flexible-beta classification. OPTSIL, OPTimizing SILhouette width; REMOS, REallocation of Misclassified Objects based on Silhouette width

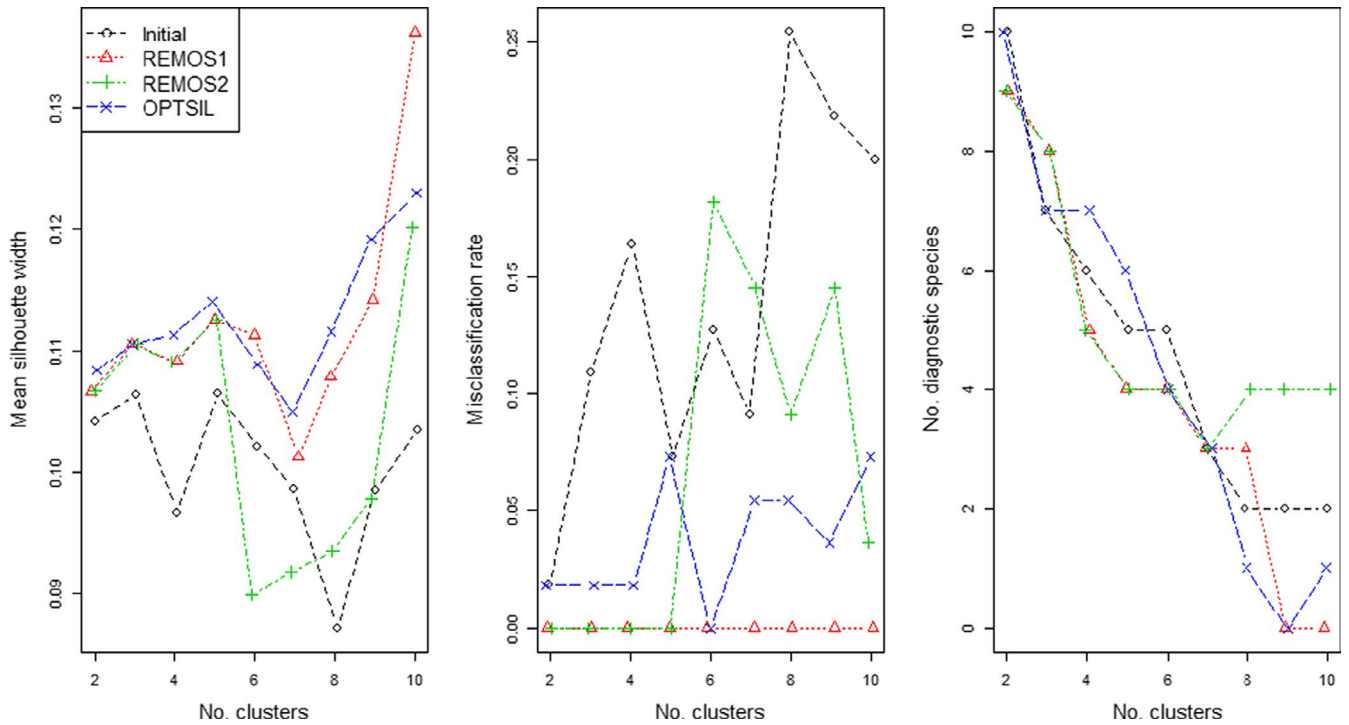


**FIGURE 4** Computation times with different sample sizes by REMOS1, REMOS2, and OPTSIL. Shortest computation times are truncated and replaced by 0.0001 s. OPTSIL, OPTimizing SILhouette width; REMOS, REallocation of Misclassified Objects based on Silhouette width

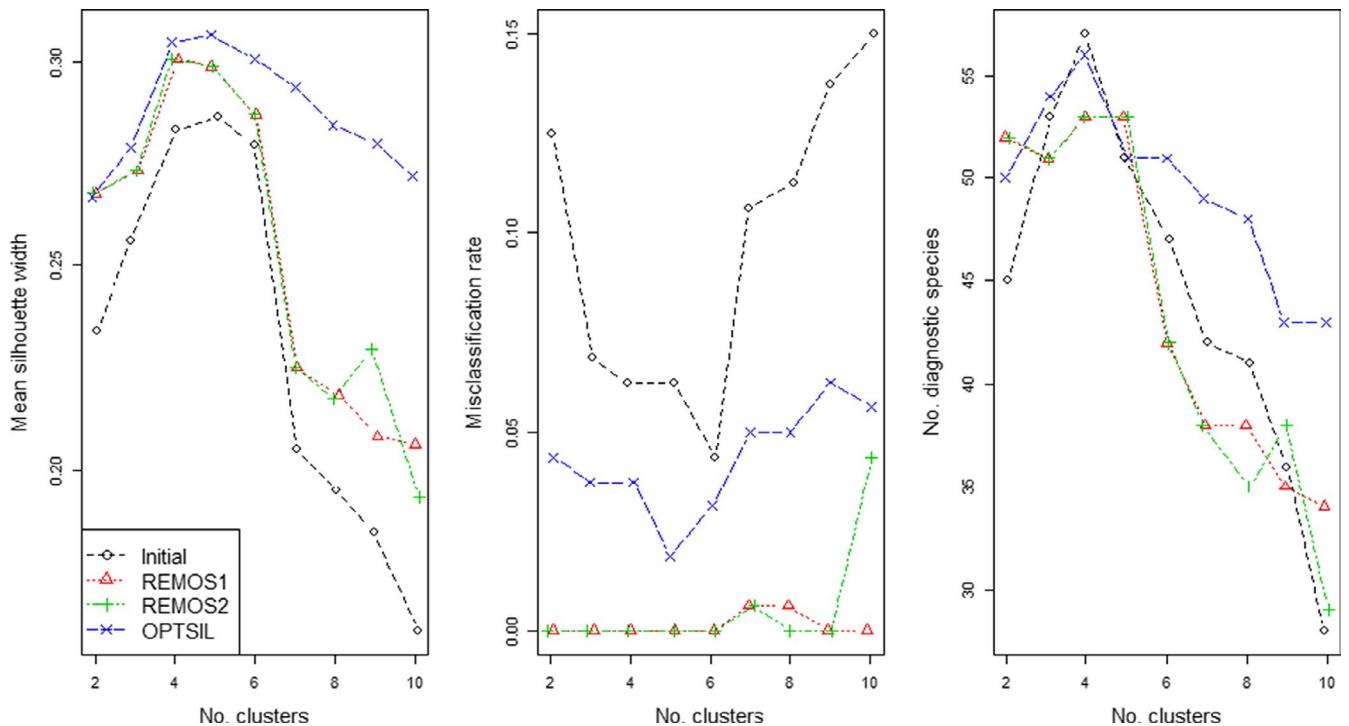
diagnostic species, while at two and five clusters REMOS algorithms reached the highest values.

On the Shoshone data set, OPTSIL reached the highest MSW across all cluster numbers, REMOS1 was the second best, showing similar (in a few cases identical) MSW values with REMOS2, and the worst was the initial classification (Figure 7). REMOS1 had the lowest MR again. This position was shared with REMOS2 between two and five clusters when both algorithms provided no misclassifications.

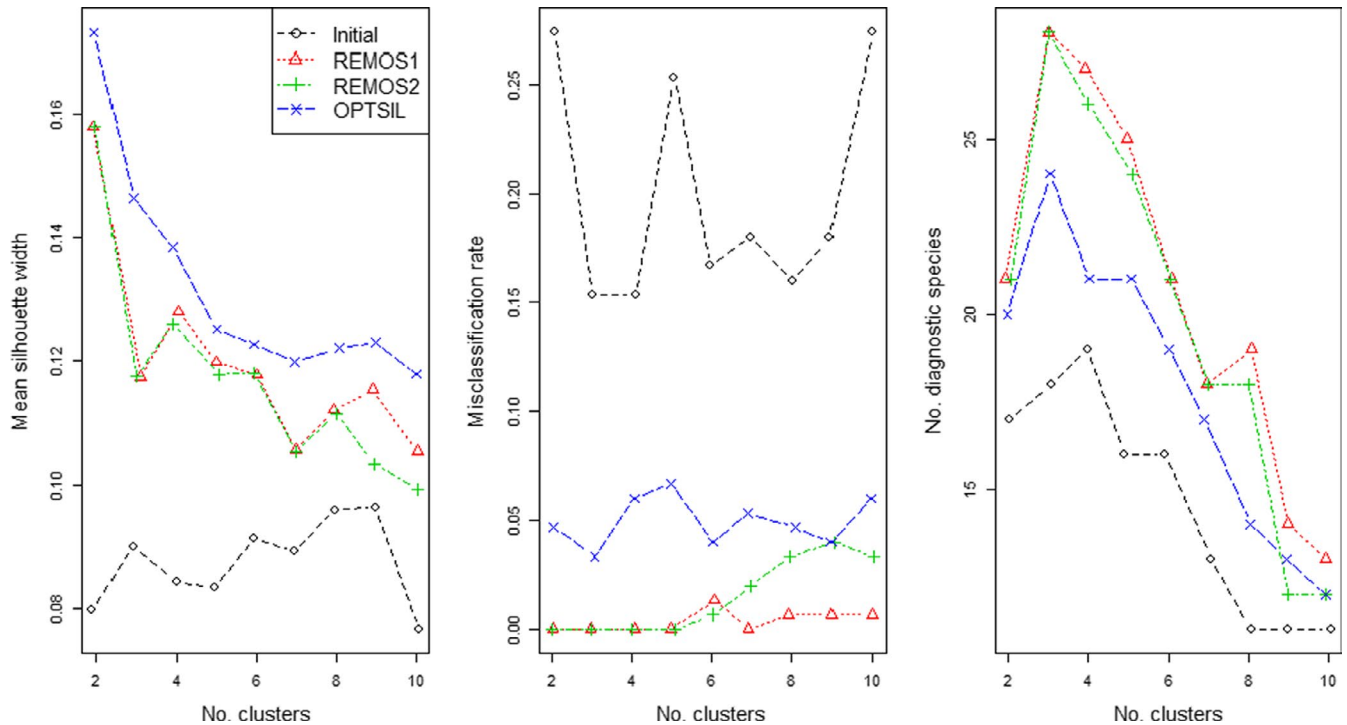
OPTSIL had MR between 0.04 and 0.07, which positioned it behind REMOS2 in all but two cluster numbers. The initial classification had the highest MR (between 0.15 and 0.27). Regarding the number of diagnostic species, the picture was different. REMOS1 exhibited the highest numbers, again in a few cases together with REMOS2, while OPTSIL was always inferior. The initial classification was again the worst in all cases, except for the 10-cluster level, where REMOS2 had the fewest diagnostic species.



**FIGURE 5** Differences in mean silhouette width, misclassification rate, and number of diagnostic species between the initial classification (without optimization), REMOS1, REMOS2, and OPTSIL solutions across the number of clusters on the Grassland data set. The initial classification was produced by the flexible-beta method ( $\beta = -0.25$ ). To avoid overlap, points are jittered in the horizontal direction on the graph. OPTSIL, OPTimizing SILhouette width; REMOS, REallocation of Misclassified Objects based on Silhouette width



**FIGURE 6** Differences in mean silhouette width, misclassification rate, and number of diagnostic species between the initial classification (without optimization), REMOS1, REMOS2, and OPTSIL solutions across the number of clusters on the Bryce data set. The initial classification was produced by the flexible-beta method ( $\beta = -0.25$ ). To avoid overlap, points are jittered in horizontal direction on the graph. OPTSIL, OPTimizing SILhouette width; REMOS, REallocation of Misclassified Objects based on Silhouette width



**FIGURE 7** Differences in mean silhouette width, misclassification rate, and number of diagnostic species between the initial classification (without optimization), REMOS1, REMOS2, and OPTSIL solutions across the number of clusters on the Shoshone data set. The initial classification was produced by the flexible-beta method ( $\beta = -0.25$ ). To avoid overlap, points are jittered in horizontal direction on the graph. OPTSIL, OPTimizing SILhouette width; REMOS, REallocation of Misclassified Objects based on Silhouette width

## 4 | DISCUSSION

Evaluating classifications with the silhouette width index makes it possible to assess the fit of individual sample units in their clusters on a standardized scale, and to upscale these index values to the level of clusters or of the entire sample. To achieve a partition which is optimized in terms of silhouette width it is straightforward to apply a clustering method which directly optimizes this criterion. Complementing an existing method with this objective, called OPTSIL, in this paper we introduced the REMOS algorithms which can be used for improving already existing classifications by reallocating misclassified objects using the silhouette width criterion. Two versions are available: at each iteration REMOS1 reallocates only the single object with the lowest silhouette width, while REMOS2 reassigns all objects with negative silhouette width to their respective closest neighbour cluster. We provide evidence on the high optimization success and time efficiency of the new algorithms through a comparison with OPTSIL.

Across the range of classifications evaluated, OPTSIL generally achieved the highest mean silhouette widths, especially on the real data. Given the optimization criterion, this was to be expected. However, in specific cases OPTSIL was outperformed by REMOS. In these cases, OPTSIL converged to a local minimum. OPTSIL is deterministic from an initial condition, and to avoid local optima would need the iteration to be started from numerous different initial

classifications. However, because OPTSIL is slow to converge this can be problematic.

REMOS1 generally achieved the fewest misclassifications. For the simulated data or for low cluster number solutions on the real data, REMOS2 often achieved the same solution exhibiting zero negative silhouette widths. However, for the real data with higher numbers of clusters REMOS2 tended not to reach such efficiency. It must be noted that different algorithms may reach the same value for MR, while their final solutions are not necessarily identical. REMOS1 and REMOS2 solutions sometimes contained no, or only very few misclassified objects, while their classifications were different. Even the number of clusters can differ between REMOS1 and REMOS2 despite equal MR (e.g., Figure S4-4). Such agreement in MSW is less probable due to its continuous scale.

In general, optimizing a single criterion results in trade-offs for other criteria, and OPTSIL and REMOS demonstrate this clearly. When comparing the optimization success of OPTSIL and REMOS on MSW and MR, it must be noted that OPTSIL directly maximizes MSW, a “higher-level” criterion of classification efficiency. MSW increases if the reallocation of a sample unit from its original to its neighbour cluster increases the separation and/or the average homogeneity of the two clusters. Therefore, the scope of MSW is at the level of clusters. On the contrary, REMOS reassigns misclassified objects to their neighbour cluster without controlling any higher-level index. Hence, REMOS has a more local perspective on classification efficiency and only implicitly optimizes related higher-level criteria.



Surely, MSW and MR correlate strongly, which is the reason why there was negligible difference between OPTSIL and REMOS when the initial classification was already efficient; that is, there was not much to optimize.

From the perspective of optimizing MSW an object with negative silhouette width is not misclassified if reallocating it to its nearest-neighbour cluster decreases MSW. Rather, a misclassification is an assignment that lowers mean silhouette width. However, as noted above, MSW cannot be high with many negative silhouette widths. Alternatively, the viewpoint that the best classification reflects strictly positive silhouette widths for as many objects as possible might be more straightforward than an “on-average correct” solution. This requires a decision from the investigator before choosing between these methods. Notably, the logic of REMOS seems advantageous for vegetation classification where classifying all sample units in the best possible way is often preferred over solutions with a number of very homogeneous and distinct clusters at the expense of some heterogeneous or inseparable ones.

An important property of REMOS2 and OPTSIL is that they are able to eliminate complete clusters from the initial classification, resulting in fewer clusters than the initial classification. This can be useful if the initial classification has more clusters than is optimal in terms of MSW or MR but is problematic if an a priori number of clusters is desired.

Tests on real data showed that OPTSIL combined with flexible-beta ( $\beta = -0.25$ ) is more efficient than REMOS algorithms in terms of MSW, although the difference is often small. In contrast, with respect to minimizing the proportion of negative silhouette widths REMOS1 provided consistently the best classifications. However, these differences may not affect interpretability the same way since we could not detect consistent differences between OPTSIL and REMOS algorithms in the number of diagnostic species. We suggest considering which cluster validity measure fits the research question the best, and then deciding between the methods discussed above.

We found clear differences in computation time between the three methods. REMOS algorithms were orders of magnitude faster than OPTSIL. This difference likely increases with sample size, number of clusters, and low MSW of the initial classification. In our tests for computation time we used rather small data sets (i.e. containing max. 400 objects) with clear cluster structure, and optimized initial classifications with relatively high MSW. Presumably, runtimes will be longer for larger and more complicated data sets, less efficient classifications, or more clusters. For modest data sets (e.g. fewer than 1,000 sample units), runtime is likely not a primary concern. However, with modern analyses in vegetation science now working with data sets with many thousands of sample units, runtime may be a major consideration. In a pilot analysis (not shown here in detail) with a data set of 1,000 points, OPTSIL needed more than eight days and six hours to reach a final solution, while REMOS1 finished in 2.356 s making a more than 300,000-fold difference in time demand. In addition, REMOS1 reached slightly higher MSW in this trial. With data sets of 5,000 plots, the runtime of OPTSIL exceeded

several months and was stopped manually before termination. If time efficiency of the analysis is crucial and the small difference in optimization success can be neglected, REMOS1 or REMOS2 should be considered instead of OPTSIL.

## 5 | CONCLUSIONS

We present REMOS1 and REMOS2 as new reallocation methods for the optimization of classifications and compare them with the related OPTSIL algorithm. Most frequently OPTSIL gives the highest final mean silhouette width; however, REMOS solutions are often only slightly worse. With respect to the proportion of misclassified objects, REMOS algorithms, especially REMOS1, provided better classifications than OPTSIL. REMOS algorithms are much more time-efficient to compute than OPTSIL. We found no systematic difference in the number of diagnostic species between vegetation classifications obtained by OPTSIL and REMOS algorithms.

As a closing remark, we advise deciding what properties a classification should hold prior to the analysis. The clustering method and the cluster validity index should be chosen in accordance with this decision. If clusters with more or less spherical shape and similar size are desired, group-forming hierarchical algorithms (e.g. flexible beta with  $\beta < 0$ , UPGMA, Ward's method) are generally suitable. At the next step, we advise exploring the possibility of improving the classification using reallocation methods examined in this paper. Suitability of a more global measure vs the fit of individual sample units as goodness criterion, as well as sample size, complexity, and computation capacity together affecting time demand should guide the researcher's choice among OPTSIL, REMOS1 and REMOS2. The resulting updated classification can be used in the same way as outputs from other numerical methods.

## AUTHOR CONTRIBUTIONS

AL raised the idea, wrote the scripts, did data analysis, led writing; DWR did data analysis, discussed results, contributed to the manuscript, responded to reviews; ZBD discussed results, commented on the manuscript. All authors critically revised the manuscript.

## DATA AVAILABILITY STATEMENT

The simulated and the Grassland data sets are available in the Appendix. Bryce and Shoshone data sets are available through the *labdsv* and *optpart* R packages respectively.

## ORCID

Attila Lengyel  <https://orcid.org/0000-0002-1712-6748>

David W. Roberts  <https://orcid.org/0000-0001-7128-6243>

Zoltán Botta-Dukát  <https://orcid.org/0000-0002-9544-3474>

## REFERENCES

Aho, K., Roberts, D.W. and Weaver, T. (2008) Using geometric and non-geometric internal evaluators to compare eight vegetation classification methods. *Journal of Vegetation Science*, 19, 549–562.

- Chytrý, M., Tichý, L., Holt, J. and Botta-Dukát, Z. (2002) Determination of diagnostic species with statistical fidelity measures. *Journal of Vegetation Science*, 13, 79–90.
- De Cáceres, M., Chytrý, M., Agrillo, E., Attorre, F., Botta-Dukát, Z., Capelo, J. et al (2015) A comparative framework for broad-scale plot-based vegetation classification. *Applied Vegetation Science*, 18, 543–560. <https://doi.org/10.1111/avsc.12179>
- Kaufman, L. and Rousseeuw, P.J. (1987) Clustering by Means of Medoids. In: Dodge, Y. (Ed.) *Statistical Data Analysis Based on the L1-Norm and Related Methods*. North-Holland, pp. 405–416.
- Kaufman, L. and Rousseeuw, P.J. (1990) *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley.
- Lance, G.N. and Williams, W.T. (1966) A general theory of classificatory sorting strategies. I. Hierarchical systems. *Computer Journal*, 9, 373–380.
- Legendre, P. and Legendre, L. (2012) *Numerical ecology*, 3rd edition. Amsterdam: Elsevier.
- Lengyel, A., Illyés, E., Bauer, N., Csiky, J., Király, G., Purger, D. et al (2016) Classification and syntaxonomical revision of mesic and semi-dry grasslands in Hungary. *Preslia*, 88, 201–228.
- Lötter, M.C., Mucina, L. and Witkowski, E.T.F. (2013) The classification conundrum: species fidelity as leading criterion in search of a rigorous method to classify a complex forest data set. *Community Ecology*, 14, 121–132.
- Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M. and Hornik, K. (2018) *cluster: Cluster Analysis Basics and Extensions*. R package version 2.0.7-1.
- Oksanen, J., Blanchet, F.G., Friendly, M., Kindt, R., Legendre, P., McGlenn, D. et al (2019) *vegan: Community Ecology Package*. R package version 2.5-6. <https://CRAN.R-project.org/package=vegan> (last accessed: Aug 31, 2020).
- Peet, R.K. and Roberts, D.W. (2013) Classification of natural and semi-natural vegetation. In: van der Maarel, E. and Franklin, J. (Eds.) *Vegetation ecology*, 2nd edition. Oxford: Wiley-Blackwell, pp. 26–62.
- Podani, J. (2000) *Introduction to the exploration of multivariate biological data*. Leiden, NL: Backhuys.
- R Core Team (2017) *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing URL: <https://www.R-project.org/>
- Roberts, D.W. (1992) *Plant Community Distribution and Dynamics in Bryce Canyon National Park: Final Report for Project PX 1200-7-0966*.
- Roberts, D.W. (2015) Vegetation classification by two new iterative reallocation optimization algorithms. *Plant Ecology*, 216, 714–758.
- Roberts, D.W. (2016) *optpart: Optimal Partitioning of Similarity Relations*. R package version 2.3-0. <https://CRAN.R-project.org/package=optpart>
- Rousseeuw P.J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53–65. [http://dx.doi.org/10.1016/0377-0427\(87\)90125-7](http://dx.doi.org/10.1016/0377-0427(87)90125-7)
- Tichý, L., Chytrý, M., Hájek, M., Talbot, S.S. and Botta-Dukát, Z. (2010) OptimClass: Using species-to-cluster fidelity to determine the optimal partition in classification of ecological communities. *Journal of Vegetation Science*, 21, 287–299.
- Rousseeuw, P.J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53–65. [http://dx.doi.org/10.1016/0377-0427\(87\)90125-7](http://dx.doi.org/10.1016/0377-0427(87)90125-7)

## SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section.

**Appendix S1.** The R code of REMOS

**Appendix S2.** The R code for the simulated data set

**Appendix S3.** The Grassland data set (in txt format directly readable to R)

**Appendix S4.** Exemplary classifications of the simulated data set

**How to cite this article:** Lengyel A, Roberts DW, Botta-Dukát Z. Comparison of silhouette-based reallocation methods for vegetation classification. *J Veg Sci*. 2021;32:e12984. <https://doi.org/10.1111/jvs.12984>