The Journal of Open Source Software

corrselect: Fast and flexible predictor pruning for data analysis and modeling

Gilles Colling [1]

1 Department of Botany and Biodiversity Research, University of Vienna, Austria

DOI: 10.21105/joss.09539

Software

- Review
- Repository
- Archive

Editor: Nikoleta Glynatsi

Reviewers:

- @dansmith01
- @danStich

Submitted: 10 September 2025
Published: 02 February 2026

Summary

`corrselect` (Colling, 2025) is an R package for reducing multicollinearity and redundancy in predictor sets. It provides two complementary approaches: (1) high-level pruning functions that return a single optimal subset, and (2) exhaustive enumeration of all maximal admissible subsets. The package handles both numeric and mixed-type data, supports forced inclusion of key predictors, and integrates with standard R modeling workflows including mixed-effects models.

Version 3.0 introduces `corrPrune()` for association-based pruning and `modelPrune()` for VIF-based model pruning, while retaining the original exhaustive enumeration functions (`corrSelect()`, `assocSelect()`, `MatSelect()`). A fast C++ greedy algorithm enables efficient pruning for large predictor sets (p $>$ 100), while exact graph-theoretic algorithms guarantee complete enumeration when exhaustive search is feasible.

Statement of Need

Collinearity among predictors is common in applied modeling and can degrade inference and prediction (Dormann et al., 2013). Popular utilities such as `caret::findCorrelation()` apply greedy, order-dependent filtering and return a single solution, typically removing the variable with the highest mean correlation at each step. This heuristic approach discards potentially useful subsets and provides no guarantee of optimality: where `caret` returns one subset, `corrselect` in exact mode might reveal a dozen equally valid alternatives. Having the full set of options helps when domain knowledge should guide final variable selection, or when researchers need to assess the sensitivity of their conclusions to predictor choice. Supervised filter methods such as FCBF (Yu & Liu, 2003) select features correlated with a target variable while removing redundancy, which is a different goal than reducing pairwise redundancy alone. Embedded and wrapper methods like the elastic net (Zou & Hastie, 2005) or recursive feature elimination (Witten et al., 2009) can be powerful but couple selection to a specific model and reduce transparency.

`corrselect` addresses these limitations through two interfaces. For routine workflows, `corrPrune()` and `modelPrune()` provide simple, deterministic pruning with a single function call. For exhaustive exploration, the package formulates a global admissible set problem: given variables $X_1, \dots, X_p$ and pairwise measures r_{ij}, find all maximal subsets S such that

$$|r_{ij}| \leq t \quad \text{for all } i \neq j \in S,$$

with a user threshold $t \in (0, 1)$. This is equivalent to finding all maximal cliques in the compatibility graph, a well-studied problem in computer science. Unlike greedy methods that return a single result, `corrselect` in exact mode enumerates *all* maximal admissible subsets, enabling researchers to explore the full solution space. This gives users both convenience and completeness.

Functionality

The package provides two complementary approaches to predictor pruning: model-agnostic methods that operate on predictors alone, and model-based methods that require fitting a model.

Model-Agnostic Pruning

Model-agnostic pruning removes redundant predictors based on pairwise correlation or association measures, without requiring a response variable. This unsupervised approach is useful for pre-modeling dimensionality reduction or when the outcome is not yet defined. The package provides both a simple pruning interface and exhaustive enumeration functions:

- **`corrPrune()`**: Returns a pruned data frame with pairwise associations below a user-specified threshold. Supports exact mode (exhaustive search, recommended for $p \leq 100$) and greedy mode (fast polynomial-time algorithm for large p). Automatic measure selection handles numeric, factor, and ordered variables. The `force_in` parameter protects key predictors from removal.
- **`corrSelect()`**, **`assocSelect()`**, **`MatSelect()`**: Exhaustive enumeration functions that return all maximal admissible subsets rather than a single solution. `corrSelect()` handles numeric data with correlations in $[-1, 1]$; `assocSelect()` handles mixed-type data using normalized association measures in $[0, 1]$, including Pearson, Spearman, and Kendall correlations, biweight midcorrelation (Langfelder & Horvath, 2008), distance correlation (Székely et al., 2007; Székely & Rizzo, 2009), the maximal information coefficient (Reshef et al., 2011), ANOVA η^2, and Cramér's V; `MatSelect()` operates directly on a symmetric association matrix.

All enumeration functions return a `CorrCombo` object containing maximal subsets, summary statistics, and standard methods (`print`, `summary`, `as.data.frame`). The helper function `corrSubset()` extracts filtered data frames from results.

For example, on the `mtcars` dataset:

```r
library(corrselect)
result <- corrSelect(mtcars, threshold = 0.7)
result
#> CorrCombo object with 12 maximal subsets
#>   Threshold: 0.7 ¦ Correlation method: pearson
#>   Sizes: 6, 5, 5, 5, ... ¦ Avg ¦r¦: 0.30, 0.27, 0.30, 0.31, ...

as.data.frame(result)[1:3, c("subset", "size", "avg_corr")]
#>                           subset size avg_corr
#> 1 mpg, cyl, drat, qsec, vs, am    6     0.30
#> 2     mpg, drat, qsec, vs, gear    5     0.27
#> 3      mpg, hp, drat, qsec, am    5     0.30
```

Unlike `caret::findCorrelation()` which returns a single variable set, `corrSelect()` reveals all 12 equally valid solutions, enabling informed selection based on domain knowledge.

Algorithms

The admissible set problem is mathematically equivalent to finding all maximal cliques in the "compatibility graph" where edges connect variable pairs with $|r_{ij}| \leq t$, or equivalently, all maximal independent sets in the "conflict graph" where edges connect pairs exceeding the threshold.

For exact enumeration, the package implements two algorithms natively in C++ (not as wrappers around external libraries such as igraph (Csardi & Nepusz, 2006)):

- **Bron-Kerbosch**: The classical maximal clique enumeration algorithm (Bron & Kerbosch, 1973), used by default for unrestricted enumeration.
- **Eppstein-Löffler-Strash (ELS)**: A near-optimal algorithm for sparse graphs (Eppstein et al., 2010), used when `force_in` seeds are specified.

Both exact methods ensure non-redundant and complete enumeration of admissible subsets. However, maximal clique enumeration is NP-hard in the general case, and exact mode may become impractically slow on large or densely connected problems. On a modern desktop (Intel i9-14900K) with correlations clustered near the threshold, $p = 100$ completed in under 1 second, $p = 150$ in ~19 seconds, $p = 175$ in ~3 minutes, and $p = 200$ in ~17 minutes. Exact mode is therefore recommended only for moderate-sized problems ($p \leq 100$).

For larger predictor sets or low thresholds where exact enumeration becomes infeasible, `corrPrune(mode = "greedy")` provides a fast polynomial-time alternative. Unlike `caret::findCorrelation()` which removes the variable with the highest mean correlation, this custom greedy algorithm iteratively removes the variable involved in the most threshold violations, with ties broken by maximum and then average association. This runs in $O(p^2 \times k)$ time where k is the number of variables removed.

Model-Based Pruning

Unlike the model-agnostic functions, `modelPrune()` addresses multicollinearity using variance inflation factors (VIF) computed within a modeling context. This supervised approach requires a response variable and considers joint relationships among predictors rather than pairwise associations alone.

- **`modelPrune()`**: Iteratively removes the predictor with the highest VIF until all remaining predictors fall below a user-specified limit. Supports multiple modeling engines (`lm`, `glm`, `lme4`, `glmmTMB`) and custom engine definitions for integration with any R modeling package (e.g., INLA, mgcv, brms). For mixed-effects models, only fixed effects are pruned while random effect structures are preserved.

Related Work

Heuristic correlation filters are widely used but are order-dependent and return only a single result. `corrselect` goes further by providing both fast deterministic pruning and exhaustive enumeration, support for mixed data types, VIF-based model pruning, and user control via `force_in`. The model-agnostic functions are interpretable and independent of any particular modeling framework, while the graph-theoretic foundation links admissible subsets to maximal cliques and independent sets.

Other feature selection methods include embedded approaches such as the elastic net (Zou & Hastie, 2005), recursive feature elimination (Witten et al., 2009), or permutation-based algorithms such as Boruta. These methods can be powerful but are tied to specific modeling frameworks and may be non-deterministic or hard to interpret when predictors are collinear. By contrast, the model-agnostic functions in `corrselect` are fast, deterministic, and formulate subset selection as a well-defined optimization problem.

Applications

The package supports feature screening in high-dimensional modeling and exploratory mapping of alternative, equally valid predictor sets. With support for correlation and association measures such as biweight midcorrelation (Langfelder & Horvath, 2008), distance correlation (Székely et al., 2007; Székely & Rizzo, 2009), and the maximal information coefficient (Reshef et al., 2011), `corrselect` is applicable across domains including genomics, network analysis, environmental modeling, and machine learning. The VIF-based `modelPrune()` function integrates directly with regression and mixed-effects modeling workflows, while the custom engine interface enables extension to specialized modeling packages.

References

Bron, C., & Kerbosch, J. (1973). Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM*, *16*(9), 575–577. https://doi.org/10.1145/362342.362367

Colling, G. (2025). *Corrselect: Correlation-based variable subset selection*. https://doi.org/10.32614/CRAN.package.corrselect

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, *1695*(5), 1–9. https://igraph.org

Dormann, C. F., Elith, J., Bacher, S., & al., et. (2013). Collinearity: A review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, *36*(1), 27–46. https://doi.org/10.1111/j.1600-0587.2012.07348.x

Eppstein, D., Löffler, M., & Strash, D. (2010). Listing all maximal cliques in sparse graphs in near-optimal time. *Algorithms and Computation (ISAAC 2010)*, *6506*, 403–414. https://doi.org/10.1007/978-3-642-17517-6_36

Langfelder, P., & Horvath, S. (2008). WGCNA: An R package for weighted correlation network analysis. *BMC Bioinformatics*, *9*(1), 559. https://doi.org/10.1186/1471-2105-9-559

Reshef, D. N., Reshef, Y. A., Finucane, H. K., & al., et. (2011). Detecting novel associations in large data sets. *Science*, *334*(6062), 1518–1524. https://doi.org/10.1126/science.1205438

Székely, G. J., & Rizzo, M. L. (2009). Brownian distance covariance. *The Annals of Applied Statistics*, *3*(4), 1236–1265. https://doi.org/10.1214/09-AOAS312

Székely, G. J., Rizzo, M. L., & Bakirov, N. K. (2007). Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, *35*(6), 2769–2794. https://doi.org/10.1214/009053607000000505

Witten, D. M., Tibshirani, R., & Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, *10*(3), 515–534. https://doi.org/10.1093/biostatistics/kxp008

Yu, L., & Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 856–863.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, *67*(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x