***Feature importance:*** Feature importance is needed before developing scaling relationships. This is because the number of extrinsic features is significant and uninformative features may not be predictive indicators in developing scaling laws. Hence, this feature database needs to be distilled to get informative extrinsic variables that influence species richness for all ten compounds. For instance, the table below describes the dataset type and the total number of extrinsic features within this database. We explore six different state-of-the-art approaches to identify and rank important features (see supplementary material and hydrosheds\_vs\_SR\_v7.pptx). This ranking using six different methods allows us to reduce the bias of discarding important features when performing feature selection. As the dataset is imbalanced, we used all 54 samples to perform feature importance. The inputs to the sensitivity analysis methods are the values of extrinsic variables. The corresponding output on which the extrinsic features are assessed is the species richness of all 10 compounds. The feature importance values are then aggregated and averaged to get a single value upon which the extrinsic variables are ranked.

Table-XXX: Summary of dataset type, the total number of extrinsic features, and a number of essential features synthesized using six different sensitivity analysis methods.

|  |  |  |
| --- | --- | --- |
| **Dataset name/type** | **Total number of extrinsic features** | **Total number of important features** |
| WHONDRS | 45 | 12 |
| StreamStats | 8 | 8 |
| HydroSheds | 294 | 8 |
| EPAWaters-Catchment | 137 | 12 |
| EPAWaters-Watersheds | 137 | 22 |

The six sensitivity analysis methods we used for feature importance include the Pearson correlation coefficient, Spearman correlation coefficient, F-test, Mutual information (MI), Random Forest (RF), and SHAPley values. Pearson correlation measures the linearity between extrinsic descriptors and species richness, while the Spearsman correlation measures this relationship's non-parametric measure of monotonicity. F-test is a univariate feature selection that allows us to select the best features based on univariate statistical tests. MI measures the strength of non-linear dependency between the variables and species richness. It is equal to zero if species richness is independent of extrinsic variables, and higher values mean higher dependence. RF model with 100 decision trees is trained, and essential extrinsic descriptors are evaluated using permutation-based feature importance. This technique allows us to measure the increase in the prediction error of the RF model after we permuted the extrinsic descriptor's values. This permutation method breaks the relationship between the feature and the actual outcome. SHAPley is a method based on cooperative game theory and is used to increase transparency and interpretability of model-agnostic feature selection models such as RF. It provides an average of all the marginal contributions of an extrinsic variable considering all possible combinations.

***Scaling laws –*** A power law function (y = bxz) describing the relationship between extrinsic descriptors and species richness for each compound is used to develop scaling laws. Here, ‘y’ is the species richness, ‘x; is the extrinsic descriptor, b is the power-law coefficient, and z is the scaling exponent. We use Scipy Python non-linear least squares to fit this power law scaling function to input-output data. Statistical tests are performed to compute the p-value and check if the estimated parameters of the power law (i.e., b and z) are significant.

***PCA analysis –*** We have also performed PCA analysis to understand the data variation across the 54 samples. A total of 11 features among the important extrinsic variables across all the datasets are used in the PCA analysis (slide-19 in hydrosheds\_vs\_SR\_v19.pptx). A total of two PCA components are extracted with scores and loadings matrix. Positive PCA loadings indicate that an extrinsic feature and a principal part are positively correlated. That is an increase in one value results in an increase in the PC. Negative loadings indicate a negative correlation between a feature and its PC. A significant loading value (either positive or negative) indicates that an extrinsic descriptor strongly affects that specific principal component.

we used statistical methods and principal component analysis…" please add details on statistical tests used, R packagaes or software used, data transformations, scaling, parameters included, etc.

Please include details of the method used for the PCA, including data scaling etc.

The important features in the Figure-4b are assembled into a dense data matrix of size 54 x 11. That is, 54 samples and 11 features. We standardize the features in the data matrix by removing the mean and scaling to unit variance. This is performed using StandardScaler Class object in Python’s Scikit-learn ML module’s data pre-processing pipeline [Pedregosa et al, 2011]. Centering and scaling happen independently on each data feature by computing the relevant statistics on the samples in the data matrix. Mean and standard deviation are then stored to be used on the data using transform method in the Scikit-learn’s StandardScaler. Once the data is processed PCA is performed. We use scikit-learn’s PCA decomposition method. Two principal components are extracted by computing the full Singular Value Decomposition (SVD) and optionally truncated afterwards. Scikit-learn uses the LAPACK implementation of the full SVD [Halko et. al., 2009] to extract the two principal components. The extracted principal components in feature space, representing the directions of maximum variance in the dense data matrix are loadings. That is, loadings represent the correlation between the original variables and the principal components. The amount of variance explained by each of the selected two components is called explained variance. The two principal components are sorted by decreasing explained variance ratio (i.e., percentage of variance explained by each of the selected components). The length of arrows in the Figure-4a indicate how much each original feature/variable contributes to a particular principal component, with higher loadings signifying a stronger relationship.

Improved text: The important features in the Figure-4b are assembled into a dense data matrix of size 54 x 11 with 54 samples characterized by 11 key features. Prior to dimensionality reduction using PCA, each feature is independently centered and scaled to unit variance using Scikit-learn’s StandardScaler class in Python [Pedregosa et al., 2011]. In this pre-processing step, the sample mean and standard deviation for each feature are computed and stored. These statistics are then applied via the StandardScaler’s transform method to ensure that every variable contributes equally, regardless of its original scale. Once the data are standardized, principal component analysis is carried out using Scikit-learn’s PCA implementation. The algorithm performs a full singular value decomposition (SVD) via LAPACK [Halko et al., 2009], yielding two principal components, which focus on the most informative directions in the feature space. These two principal components, often called loadings, define the directions of maximum variance within the standardized data matrix and quantify the correlation between the original features/variables and the new component axes. Each component’s importance is measured by its explained variance, with components ordered by descending explained variance ratio (i.e., the percentage of total variance captured). In Figure 4a, the length of each arrow corresponds to the magnitude of a feature’s loading on a particular component. Longer arrows indicate stronger relationships between the original variables and the principal components, thereby highlighting which features drive the major axes of variability in the dataset.

References:

* Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, pp.2825-2830.
* Halko, N., Martinsson, P. G., and Tropp, J. A. (2011). “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions”. SIAM review, 53(2), 217-288.

The statistical tests we performed for feature selection include F-test, mutual information, Pearson correlation, and Spearsman correlation. The methods based on F-test estimate the degree of linear dependency between two random variables. On the other hand, mutual information methods can capture any kind of statistical dependency, but being nonparametric, they require more samples for accurate estimation. For more specifics, see Section-2.3. This statistical testing is necessary to downselect the features to construct scaling laws and PCA analysis. It can be seen as a preprocessing step to an estimator such as Random Forest and also to estimate hierarchical feature importance (see Figure-2, Figure-3, and Figure-4b).

In our hierarchical feature selection pipeline, we applied multiple statistical tests such as F-test, mutual information, Pearson correlation, and Spearman correlation. This is performed to rigorously evaluate the relationships between feature/variables and target outcomes (e.g., species richness). The F-test-based approach assesses the strength of linear dependencies between pairs of random variables, providing a univariate parametric perspective on feature relevance. In contrast, mutual information techniques offer a nonparametric framework capable of detecting any form of statistical dependence, however, at the cost of requiring larger sample sizes to achieve reliable estimates. Pearson correlation further complements this analysis by quantifying the degree of linear association, while Spearman correlation captures monotonic relationships regardless of their functional form. Detailed implementation and specifics can be found in Section 2.3. This comprehensive statistical testing serves to downselect the most informative features before constructing scaling laws and performing subsequent PCA. By prefiltering the feature set, we streamline the subsequent PCA decomposition and ensure that only variables with demonstrable predictive power contribute to the derived principal components. Additionally, these tests function as a preprocessing step for ML estimators such as Random Forest, improving model efficiency and interpretability. Hierarchical feature importance metrics, illustrated in workflow Figure 2, results Figure 3, and downselected features Figure 4b, leverage this testing to rank variables according to their relative contribution. Overall, our multi-method approach enables robust feature selection, balancing parametric sensitivity and nonparametric flexibility to support both statistical analysis and downstream scaling law development.