**Short title:**

MVApp - Multivariate data analysis pipeline

**Corresponding author:**

Magdalena M. Julkowska

**Article Title:**

MVApp – Multivariate analysis application for streamlined data analysis and curation

**All authors and their affiliations:**

Magdalena M. Julkowska1, Stephanie Saade1, Gaurav Agarwal2, Ge Gao1, Yveline Pailles1, Mitchell Morton1, Mariam Awlia1, Mark Tester1

King Abdullah University of Science and Technology (KAUST), 1Biological and Environmental Sciences and Engineering Division (BESE), 2Computer, Electrical and Mathematical Sciences and Engineering Division (CEMSE)

**One sentence summary:**

MVApp offers a free and collaborative platform for streamlined curation and analysis of the plant phenotyping datasets.

**Authors contributions:**

MMJ designed the project and supervised the design of the MVApp tabs, wrote the first draft of the manuscript, and designed the data upload, curve fitting, outlier removal and hierarchical clustering tabs; SS and MM designed the data exploration tab; SS designed the heritability tab; GG designed the data correlation analysis tab; MA designed the reduction of dimensionality tab; YP designed the k-means clustering tab; and GA designed the quantile regression tab. All authors contributed by writing their designated sections in the MVApp README page (https://mmjulkowska.github.io/MVApp/), filming YouTube video tutorials on the MVApp channel, and editing the manuscript. MT provided the funds and resources for the project and edited the final version of the manuscript.

**Funding information:**

The research reported in this publication was supported by funding from KAUST, through both baseline support to MT and Office of Sponsored Research (OSR) Award No. 2302.

**Email address of corresponding author:**

Magalena.Julkowska@kaust.edu.sa

**Abstract**

Modern phenotyping techniques yield vast amounts of data that are challenging to manage and analyze. When thoroughly examined, this type of data can reveal genotype-to-phenotype relationships and meaningful connections among individual traits. However, efficient data mining is challenging for experimental biologists with limited training in curating, integrating and exploring complex datasets. Additionally, data transparency, accessibility and reproducibility are important considerations for scientific publication. The need for a streamlined, user-friendly pipeline for advanced phenotypic data analysis is pressing. In this manuscript we present an open-source, online platform for multivariate analysis (MVApp), which serves as an interactive pipeline for data curation, in-depth analysis and customized visualization. MVApp builds on the available R-packages and adds extra functionalities to enhance the interpretability of the results. The modular design of the MVApp allows for flexible analysis of various data structures and includes tools underexplored in phenotypic data analysis, such as clustering and quantile regression. MVApp aims to enhance FAIR data transparency, streamline data curation and analysis, and increase statistical literacy among the scientific community.

**Introduction:**

Advances in data acquisition methods have enabled the rapid collection of a vast range of multivariate biological datasets from plants, animals, single-cell systems and more (Marx, 2013). The development of next-generation high-throughput phenotyping platforms in particular has led to data-rich experimental outputs that capture many physiological processes simultaneously across large populations and under various conditions through time. And yet, the usefulness of these emerging technologies is limited by high operational costs of commercial phenotypic platforms and the ability of researchers to curate, integrate and explore these complex outputs (Howe et al., 2008). The appropriate interpretation of phenotypic data often requires expertise in programming (R, Python, MATLAB) and statistics or access to expensive statistical software, such as SPSS or JMP. While great efforts have been made to develop open-source data curation and analysis platforms for exploring RNAseq (McCarthy et al., 2017) and metabolomics data (Xia et al., 2015), the platforms for large phenomic datasets are not as advanced. Although the interactive tools used for Genome-Wide Association Studies (GWAS) (Seren et al., 2012) include a brief section on analyzing phenotypic data, the scope of such analysis is not comprehensive. While the software typically provided with the commercial phenotyping platform provides an insight on the observed trends during the experiment, the underlying code is not available, making it inflexible and oftentimes not suitable for various experimental designs. Additionally, most off-the-shelf tools lack modules crucial for multivariate analysis, such as Principal Component Analysis (PCA), Multidimensional Scaling (MDS) or regression models, and none of them provides a standardized pipeline for outlier identification. Yet, standardized methods of data curation, processing and analysis prior to publication are increasingly necessary as the scientific community strives for FAIR (findable, accessible, interoperable and reproducible) data (Wilkinson et al., 2016; Reiser et al., 2018). Such optimization of data processing tools and protocols will not only accelerate and standardize data exploration and visualization but will also promote better reproducibility and transparency of data curation and analysis. While tools for high-throughput phenotyping, like HTPmod (Chen et al., 2018), are gradually providing easier access to the tools for data analysis, we think that integrating the community input, keeping the code open-source, and keeping those tools up-to-date is crucial for sustainable and transparent data analysis.

To contribute towards this goal, we developed an open-source online platform called MVApp, an interactive and modular data curation and in-depth multivariate analysis pipeline. MVApp provides a comprehensive toolkit for the thorough exploration of diverse data structures, where the effect of multiple independent variables, including genotype, treatment and time, can be examined. MVApp aims to help scientists and research staff with limited knowledge of R programming and statistics to curate their data in a standardized manner and to perform comprehensive and robust statistical tests within minutes. We developed MVApp in R using the Shiny framework (Chang et al. 2017), exploiting R’s computational potential by integrating various packages into a graphical user-interface. Examples of previously developed Shiny applications are abundant, including box plot generators (Spitzer et al., 2014), RNAseq data analysis pipelines (Nelson et al., 2017; Li and Andrade, 2017), genomic prediction simulators (Morota, 2017), growth modeling (Chen et al., 2018) and tools for teaching plant breeding (Matias et al., 2018) or business analytics (Nijs, 2018). MVApp is not only combining the standard methods for multivariate analysis, but also helps with data interpretation. By following the comprehensive README (<https://mmjulkowska.github.io/MVApp/> ), first-time users can decide which modules are of the interest for their datasets. Further guidance is provided by in-app suggestions, which advise on decisions between specific methods (e.g. type of curve to fit, number of clusters to use) to apply for their datasets. The MVApp provides the output of the analytical tools and statistical tests along with guidelines for interpretation, e.g. what can be inferred from a statistical test based on the null hypothesis and resulting p-value. Additionally, the R-commands used to make individual parts of the analysis can be viewed for each section, ensuring the full transparency of the analysis. To make our platform accessible and safe to use, the data uploaded to MVApp ([http://MVApp.kaust.edu.sa/MVApp](http://mvapp.kaust.edu.sa/MVApp)) is not saved on the server; the input is deleted when the session is closed.

In this manuscript, we showcase the possibilities of MVApp using a dataset that includes the phenotypic data of nine Arabidopsis accessions grown under either control or salt-stress conditions over the course of eight days (Awlia et al., 2016). The dataset (**Table S1**) was collected using the high-throughput phenotyping PlantScreenTM conveyor system (PSI, Czech Republic), which captured seven plant morphology traits that describe the rosette size and architecture, as well as 43 chlorophyll fluorescence traits, reflecting plant photosynthetic activity (**Table S2**). In total, the dataset contains 50 phenotypes scored for 160 individual plants, across seven time points. This dataset exemplifies data structures containing up to three independent variables. A similar analysis could be employed to examine natural diversity panels consisting of hundreds of genotypes, to compare the phenotypes of mutant and wild-type lines, or to explore phenotype-to-phenotype relationships across different genetic backgrounds. The spatial variation in individual phenotypes can also be inspected, though MVApp does not yet provide a spatial correction feature. The modular design of the MVApp ensures that other modules can be added in the future, dependent on community involvement, contributions and feedback (<https://github.com/mmjulkowska/MVApp/blob/master/CONTRIBUTING.md>). We hope that the widespread use of MVApp will increase statistical literacy among the early career peers, accelerate scientific discovery and ensure transparency of data processing.

**Results:**

**Dynamic response characterization**

High-throughput phenotyping platforms allow for the non-destructive screening of e.g. plants and, therefore, the characterization of their dynamic temporal responses, such as growth or changes in photosynthetic efficiency. These dynamic responses can be summarized by fitting different types of curves (linear, quadratic, exponential and square root) and polynomial functions to changes in the measured trait values. Here, we use the increase in rosette area as an example (**Figure 1**). The simple functions (linear, quadratic, exponential and square root) are fitted using a linear model through the data points. For quadratic, exponential and square root functions, the phenotype (e.g. rosette area) is transformed by using squaring, natural logarithm or square root, respectively, before fitting the linear model. The linear fit for each sample can be examined using the fit-plot tab in MVApp (**Figure 1A**). Based on the mean regression coefficient (R2) of the curve, which is calculated for each individual sample, MVApp suggests the model with the best fit. Polynomial functions can also be fitted to the data points using cubic and smoothed splines. Because polynomial functions have high R2 values compared to the other models, they are not included in the best model-fit suggestions. In our example, we evaluate the increasing rosette area in the nine Arabidopsis accessions described above. We found that a quadratic function fit the dataset most accurately (R2 values of 0.9711 and 0.913 for the control and salt-treated plants, respectively), while the exponential model had the second-best fit (R2 values of 0.9668 and 0.907 for the control and salt-treated plants, respectively) (**Table S3**).

Curve fitting can also be used as a data curation method in which the user excludes samples whose R2 is below a chosen threshold (R2 < 0.7 is the default option). All of the fitted curves can be compared to the original data points and viewed on a fit plot to identify samples that do not follow the expected dynamics observed in the experiment (**Figure 1B**). For example, in our dataset, we identified seven samples with R2 values below 0.7 (**Table S4**). By examining pictures of those specific samples, we realized that those plants had died during the experiment. Therefore, those samples were removed in the curated dataset for subsequent analyses.

The MVApp offers the option to view summary statistics based on the fitted curves. The user can choose whether the summary statistics are calculated for all the data points or for the data curated based on the R2-value threshold. The summary of the growth dynamics can be viewed as box plots, violin plots, dot plots or bar plots, where the significant differences between user-defined groups (e.g. based on genotype and treatment) are tested using one-way ANOVA and Tukey’s HSD pairwise-comparison tests. For our dataset, we observed that the treatment, the genotype and their interaction had significant effects on the rosette growth rate (**Figure 1C**). Col-0 and Te were the fastest growing accessions under control conditions, while the highest growth rate was observed in Rsch and C24 under the salt-stress conditions. Cvi was the slowest growing accession under both the control and salt-stress conditions. Including the seven samples with R2 values below the threshold level of 0.7 did not significantly affect the observed trends (**Figure S1**).

**Outlier identification**

Data curation prior to any analysis is crucial but is oftentimes a lengthy, inconsistent and manual process with a distinct paucity of standardized guidelines. To ensure consistency between individual experiments performed across different labs, methods used for data curation require standardization. MVApp accelerates and standardizes the curation of data by highlighting possible outliers based on user-selected grouping variables, such as treatment, genotype and time point. These outliers are identified using various detection methods (CHALONER and BRANT, 1988; DiazGarcia:2004hq; Leys et al., 2013), which include the interquartile range (1.5xIQR), the Cook’s distance, the Bonferroni outlier test or the standard deviation from the median, based on one or multiple measured traits. If the input data contain a time element or a gradient series suitable for curve-fitting, then a low regression coefficient (indicating a poor fit of the trend line) can be used as an additional criterion for outlier selection. In our example, all traits were used for data curation. Samples were identified as outliers when the values for at least 12 measured traits from that specific sample were outside of the 1.5 interquartile range. In this manner, 28 outliers were identified and removed from the curated dataset (**Table S5**).

The graphical overview in the outlier removal tabs in MVApp allows the user to visualize the data both with and without the outliers in order to observe the effects of the selected outlier identification method. We use the trait rosette perimeter as an example (**Figure 2**). The effects of outlier removal were particularly visible in the Te accession, where both the variance among individual time points and the number of data points outside the normal distribution range were visibly reduced. For other traits, such as non-photochemical quenching or maximal quantum yield in the light-adapted state (Fv´/Fm´) (**Figure S2**), the effects of outlier removal were not as pronounced, with only very extreme data points being removed. As the removal of outliers can result in the loss of important information, we strongly encourage the user to inform their decisions concerning outlier removal and data curation by reviewing their experimental notes, raw data and images.

**Tests for significant differences**

MVApp facilitates standard statistical tests, including parametric and non-parametric tests. Initially, measured traits can be evaluated for normal distribution and equal variance, and be visually examined using histograms, quantile-quantile plots and box plots to validate whether the parametric test assumptions have been met. Responsive in-app message boxes provide assistance with the interpretation of the p-values and selection of parametric or non-parametric tests. In our dataset, the rosette area follows a normal distribution across all accessions according to the Shapiro-Wilk test (**Figure 3A**). As the parametric tests assume that the samples being compared have equal variance, MVApp integrates two tests for variance (Bartlett and Levene) and a graphical visualization of the observed variance using box plots for the observed data (y), the subtracted median [y – med(y)] and the absolute deviation from the median [abs(y-med(y))]. We observed an equal variance among the accessions for all the salt-stressed plants seven days after salt treatment (p-values > 0.05), but not for plants grown under control conditions (p-values of 1.65x10-6 and 0.026 for the Bartlett and Levene tests, respectively) (**Figure 3B**). This suggests that the differences among the measured accessions are more pronounced under control conditions than under salt stress. Furthermore, the traits related to quantum yield and non-photochemical quenching [ϕ(P) and ϕ(NPQ)] showed a normal distribution and equal variance under both control and salt-stress conditions (**Figure S3 A-B**), indicating that genetic variation and the environment affect these traits less significantly than the rosette area.

Basic parametric tests such as ANOVA assume a normal distribution and equal variance. When these assumptions are not met, users should consider transforming the data or using a non-parametric test, such as Kruskal-Wallis. Both parametric and non-parametric tests are included in MVApp, together with Tukey’s test and the Mann-Whitney/Wilcoxon tests for pairwise comparison. As our data on rosette area did not meet the assumptions for a parametric test, we examined the differences among the accessions using the Kruskal-Wallis test. Significant differences were observed between accessions grouping for the final day of measurement under control and salt-stress conditions (**Figure 3C**). On the final day of measurements, the Col-0, Te and C24 accessions had developed the largest rosettes under both conditions studied, while Nd, Can and Cvi had the smallest. Rsch and Co exhibited the highest and lowest quantum yields (ϕ(P)), respectively, measured across the nine accessions under both the control and salt-stress conditions (**Figure S3 C**). The exact opposite trend was observed for non-photochemical quenching [ϕ(NPQ)] (**Figure S3 C**). This suggests that the traits reflecting chlorophyll fluorescence might be complementary to the traits commonly used to describe rosette size, providing new insight into plant performance under various conditions.

The parametric tests implemented in MVApp include the one- or two-sample t-test and one- or two-way ANOVA using Tukey’s HSD pairwise test. The non-parametric tests available in MVApp include the Kolmogorov-Smirnov test for examining the differences between two samples, as well as the Kruskal-Wallis and Wilcoxon tests, which are commonly used to compare multiple samples. All of these methods of analyses are accompanied by graphical visualizations in the form of box plots, violin plots, bar plots, and scatter plots. Additionally, MVApp facilitates analysis of the interaction between two predefined factors, such as treatment and genotype, using two-way ANOVA. Depending on the data structure, the interaction between two user-defined variables, such as genotype and treatment, can be examined in separate data subsets*,* *e.g.* across individual time points. In our datasets, we examined the interaction between genotype and treatment in the data subsets for individual time points. In the case of rosette area, we found a significant interaction between genotype and treatment starting from four days after the salt treatment and continuing through to the end of the experiment (**Figure 3D**). Interestingly, the interaction between genotype and treatment was already apparent in quantum yield and non-photochemical quenching just one day after treatment (**Figure S3 D**). This indicates that the chlorophyll fluorescence traits are more responsive to GxE interactions than plant size.

**Correlation analysis**

Measuring multiple traits facilitates the study of phenotype-to-genotype relationships (*e.g.* comparing mutants to wild-type varieties), but also allows a better understanding of the phenotype-to-phenotype interactions. For instance, correlation analyses can be used to identify phenotypic traits that are highly correlated. Changes in correlation caused by different environments or genetic contexts, for example, can also be detected. These subtleties often remain unexplored. Interactive scatterplots can be used to further inspect the correlation between two chosen traits. Various data structures are accommodated by allowing the user to subset the data prior to the correlation analysis. As such, the MVApp allows the user to examine the strength, variability and significance of correlations across individual subsets.

We examined the correlations for similarities and differences between the control and salt-stressed plants in our dataset (**Figure 4, Table S6**), which allowed us to identify major clusters of traits. Traits related to photosynthetic quantum yield (Fv’/Fm’ and ϕ(P) were strongly correlated with each other, while traits reflecting photochemical quenching and electron transport beyond PSII photochemistry (qP, qL, and ETR) were less correlated with the single time-point measurements of chlorophyll fluorescence traits reflecting the maximum, minimum and instantaneous fluorescence (Fm, Fo and Ft, respectively). All of these photosynthetic traits were found to be negatively correlated with traits reflecting non-photochemical quenching parameters [ϕ(NPQ), qN], but the strongest negative correlations were observed between the maximal and actual quantum yields in the light-adapted state [Fv’/Fm’ and ϕ(P)]. Additionally, various photosynthetic traits measured at different photon irradiances (Lss1 to Lss4) showed highly significant correlations. The rosette area was positively correlated with the quantum-yield cluster of traits, and was negatively correlated with the non-photochemical quenching traits. However, relationship between plant size and the non-photochemical quenching traits was less pronounced under salt-stress conditions; many of the correlations were not significant (p-value > 0.05) (**Figure 4**).

**Reduction of dimensionality**

When multiple traits are measured from the same individual, a multidimensional phenotypic space that defines each sample can be obtained. However, the measured traits are likely to be highly correlated. Therefore, the dimensionality of the data can be reduced, thus simplifying the data while maintaining important trends and patterns.

The contribution of each measured trait to the overall observed variance can be examined using principal component analysis (PCA). PCA allows users to determine the minimum number of dimensions required to adequately summarize the phenotypic variance. Observations of potentially correlated measured traits are transformed into a set of linearly uncorrelated variables, called principal components (PCs) (Lever et al., 2017). The first PC accounts for the largest observed variance, providing the best possible summary of the variance observed across the experiment. As PCA is sensitive to relative scaling of the observed variables, we included the option to scale the data prior to performing PCA in MVApp. Additionally, depending on the data structure the user can perform the PCA on the specific data subsets, grouped by treatment or time point, for instance. In our example of a scaled dataset, ten PCs were needed to describe at least 98% of the variance observed in the plants grown under both the control and salt-stress conditions (**Figure 5 A, Table S7**). The highest contribution to the first PC came from traits related to non-photochemical quenching and quantum yield under the light-adapted state (**Figure 5 B, Table S8**), with no differences in contribution observed between the two conditions. When the data was reduced to the first two PCs, differences between individual accessions were found in both the salt-stress and control conditions (**Figure 5 C**). While PCA is not uncommon for multivariate datasets, its implementation and interpretation can be challenging. Furthermore, the contributions of individual traits to PCs are seldom described. Those issues are addressed in MVApp, making it intuitive for the user to understand and explore the PCA, providing additional insight into the data.

Another method of reducing the dimensionality of a dataset containing multiple phenotypes is multidimensional scaling (MDS). With MDS, the levels of similarity among the objects in a dataset are visualized based on their distances from each other. MDS can be done for individual samples in the dataset to see the similarity among individual replicates, or for all measured traits to determine their relationships and changes therein under different conditions. As MDS, like PCA, is also sensitive to the scaling of the variables, the scaling is optional. MVApp also allows the user to perform MDS on all the measured phenotypes, and then cluster them using k-means clustering. In our dataset, we observed four clusters, which were related to rosette area, quantum yield, non-photochemical quenching and single-time point chlorophyll fluorescence parameters (**Figure 6**). Interestingly, the traits were grouped into the same clusters and the positions of the traits respective to each other were almost mirror images for both the control and salt-stress conditions. These results suggest that the relationships among the traits do not change in response to salt stress.

**Cluster analysis**

Once the most interesting traits have been identified, they can be used to group samples with similar behavior into clusters. Clustering can be based on any number of measured traits, and is used to reveal subgroups of samples within an experiment that exhibit similar response patterns. In MVApp, users can perform either k-means or hierarchical clustering analyses. K-means clustering assigns the individual samples to a user-defined number of centroids. In order to aid the user, MVApp provides a number of visual and cumulative methods for defining the optimal number of K-mean clusters. The clusters can be visualized using bar plots or scatter plots according to their measured traits in order to evaluate the contribution of the user-defined trait to the overall variation in the data. Hierarchical clustering, in contrast, does not require the user to predefine the number of clusters. Based on selected traits, the samples are sorted into a dendrogram. This is matched to a heatmap illustrating the value of each trait selected for clustering (**Figure 7A**).

Using hierarchical clustering, we grouped the average values of the accessions scored on the final day of measurement based on traits reflecting rosette architecture, quantum yield and non-photochemical quenching using Ward’s method (**Figure 7A**). From this, we found that accessions with a large rosette area showed high quantum yield and low non-photochemical quenching, while accessions with a small rosette area showed low quantum yield and high quenching. Using the dendrogram, which represented the relationships between the individual samples, we selected a specific distance at which the samples were separated into three clusters (**Figure 7B, Table S9**). MVApp allows the examination of the differences between clusters using box plots, ANOVA or Tukey’s test for significance (**Figure 7C**), and automatically lists the traits that are significantly different between clusters. Cluster 1 included accessions grown mainly under salt-stress conditions, with a medium-sized rosette area and high non-photochemical quenching. Cluster 2 contained the accessions with the largest rosette area, measured exclusively under control conditions. In Cluster 3, accessions grown under both salt-stress and control conditions were found, and these accessions had the smallest rosette area but a high quantum yield. No differences among the clusters were observed in the traits reflecting rosette architecture (**Figure S4**). The high photosynthetic performances of Clusters 2 and 3 were also reflected in an Electron Transport Rate beyond Photosystem II (ETR) (**Figure S4**). These results confirm the previous observation that plant size is defined by photosynthetic efficiency (**Figure 4**), which is represented by both quantum yield and non-photochemical quenching. Additionally, the hierarchical clustering shown here indicates that a group of accessions can develop small rosettes despite high photosynthetic performance. As no differences in the single time-point measurements of chlorophyll fluorescence were found between Clusters 2 and 3, mechanisms other than photosynthetic efficiency may be limiting the growth of accessions belonging to Cluster 3. This type of cluster analysis for multivariate phenotypic data remains uncommon in the field, despite the intriguing insight it can provide into plant responses. Through its inclusion in MVApp, we aim to encourage the use of this type of analysis.

**Quantile regression**

Advances in phenotyping methods have enabled scientists to simultaneously examine multiple traits that could be contributing to a trait of major interest, such as plant size or yield. However, correlation and PC analyses limit this examination to the general trends observed in an experiment. Furthermore, the Ordinary Least Squares regression model that are typically used to evaluate trait contributions to yield or survival are based on the trait average values observed across a population. However, the contribution of individual traits biomass or yield can differ between the smallest and largest plants within an experiment. Therefore, we integrated quantile regression, which identifies those traits that might contribute significantly to the trait of major interest in each quantile, into MVApp. As quantile regression is relatively new, quite uncommon, and challenging to set up – MVApp brings it within the reach of any scientist. The various data structures can be accommodated by selecting multiple grouping variables and performing the quantile regression on the subsets of the data. Additionally MVApp allows for results of quantile regression to be compared with the results obtained from traditional ordinary least squares regression model.

For our dataset, we grouped the traits per treatment and performed the quantile regression for the subsets of individual time points. The quantum yield in the light-adapted state (ϕP) and ETR contributed to the size of the rosette area in plants grown under both the control and salt-stress conditions through time, though the contribution was higher for the plants grown under control conditions (**Figure 8 A, Table S10**). Interestingly, neither quantum yield in the dark-adapted state nor non-photochemical quenching contributed to the rosette area under either condition (**Figure 8 A, Table S10**). When non-photochemical quenching [ϕ(NPQ)] was chosen as the trait of major interest, we observed significant contributions of quantum yield in the light-adapted state (Fv’/Fm’) in most quantiles; this trait’s contribution was mainly observed under the control condition (**Figure 8 B, Table S10**). When either quantum yield or ETR was chosen as the trait of major interest, these two traits contributed only to each other, and not to other chlorophyll fluorescence parameters such as non-photochemical quenching or quantum yield in the light-adapted state (**Figure 8 C, Table S10**). By performing quantile regression on different traits of interest, we developed an understanding of how the traits related to each other (**Figure 8 D**). We observed that ETR and quantum yield under light-adapted state were tightly linked to each other in all quantiles, affecting the majority of the measured traits and mutually affecting each other, but were not explained by any of the other measured traits. Although the trait contribution did not differ substantially among the individual quantiles in our example dataset, we think that it will be a useful feature for more complex datasets covering higher ranges of variation.

**Estimation of heritability**

Many high-throughput phenotyping experiments use data from forward genetics studies. For in-depth analyses in these types of studies, estimating heritability based on a limited number of genotypes is of utmost importance. MVApp allows the estimation of broad-sense heritability from the measured traits, thereby enabling an informed decision on whether a specific phenotypic trait has sufficient genetic variance to be used for forward genetic studies. In MVApp, heritability is calculated as the ratio of the total genetic variance to the total phenotypic variance. Depending on the data structure, heritability can be estimated for individual data subsets, such as treatment and time point. For our datasets, we calculated heritability of individual traits per treatment and time point. The heritability of individual traits varied between 0 and 0.9862 (**Table S11**). Traits measured within the first two days exhibited low heritability (< 0.6), which increased above 80 after four days of measurement for all traits except the single time-point measurements of photosynthetic activity in light- and dark-adapted conditions (Fm, Fo, Ft in Lss1 to Lss4). Traits associated with quantum yield, non-photochemical quenching and rosette area showed the highest heritability, while traits associated with rosette architecture and fluorescence in the dark-adapted state (Fm, Fo, and Ft) had the lowest heritability. The estimated heritability was slightly higher in plants grown under the control condition (0.9081) than in plants grown under salt stress (0.875) (**Table S11**). As the median-estimated heritability for all the measured traits was above 0.89, most of the measured traits seemed to have a genetic basis, which could be dissected using forward genetic screens.

**Discussion:**

We live in the exciting time of high-throughput phenotyping, which can generate an avalanche of data reflecting the complexity of different biological systems (Fahlgren et al., 2015). However, analysis of this data takes a significant amount of time and effort (Howe et al., 2008), and comparing data outputs requires standardization of data curation and analysis. The MVApp provides a flexible analytical pipeline, which can deal with multiple data structures and accommodates multiple independent variables, such as treatment, genotype and time points, which can be used to subset or group the data for individual analyses. The existing tools, such as HTPmod (Chen et al., 2018), recognize the need for visualization and modeling of the high-throughput phenotyping data, but do not offer opportunities for future extensions. MVApp uniquely strives to make a first step towards a future framework for standardizing data curation, analysis processing, and visualization of diverse datasets, with community input in this process being clearly invaluable. Therefore, we encourage our peers to submit their comments and suggestions for new and improved feature of the MVApp as indicated in the contribution guidelines (<https://github.com/mmjulkowska/MVApp/blob/master/CONTRIBUTING.md>). By including in-app messages, helping to interpret the results of various tests, MVApp provides a comprehensive guideline, particularly valuable to first-time users and early career researchers, for data curation, interpretation and analysis. Streamlining and standardizing data analysis protocols will contribute to the FAIR data principles and improve the review process for publications and other scientific outputs. Our interactive tool for data curation and analysis, MVApp, strives to enhance the transparency of data curation and analysis by improving processing times and by reducing the need for expensive software and extensive knowledge of R or statistics. Despite the wide availability of data analysis tools, none of them are designed for time-efficient analyses of the big datasets generated by medium- and high-throughput phenotyping platforms. co MVApp exploits the flexibility of R-based statistical analysis and combines it with a graphical user interface. The graphs and tables produced in MVApp can be downloaded in a publication-ready format with default figure legends, which contain information about the analysis performed and the preprocessing steps, including data curation, allowing the graphs to be easily reproduced. We think that applications such as MVApp will not only contribute to the availability of FAIR data, but also will encourage the scientific community to both share their raw data and standardize data curation so that the figures and analyses reported in scientific publications can be reproduced and better understood by the wider audience.

Outlier curation is an integral part of data analysis, but its importance is often overlooked. As none of the existing data-analysis pipelines includes outlier detection, the MVApp is a pioneer for streamlined data curation, providing a significant contribution to FAIR data processing. In the past, outlier curation methods were developed using coarse-to-fine models for the identification of abnormalities in phenotypes related to plant photosynthesis (Xu et al., 2015). The outlier selection methods in MVApp provide opportunities to identify the possible outliers independent of the nature of the phenotypic trait. In our first release of MVApp, we have included four different methods for outlier identification. Provided that the data contains a time or gradient component, curation can also be performed based on fitting a curve to the growth of individual plants. By visualizing the data before and after outlier identification, the user can make an informed decision about which samples to retain for further analysis. However, the user must be careful when removing data points, as the outliers themselves might contain important information (Altman and Krzywinski, 2016). Therefore, the original dataset containing all the samples, including outliers, remains available in the MVApp dropdown menu for subsequent analysis. We are not aware of any other application, which integrates outlier curation that is as transparent and simple to use as the one integrated into MVApp.

For the large datasets that are often produced by high-throughput phenotyping platforms, MVApp offers different methods of dimensionality reduction. Dimension reduction by PCA or MDS can simplify the data by summarizing it in a limited number of dimensions. Both PCA and MDS are often used in metabolomic (Zhang et al., 2016), transcriptomic (Yano et al., 2018) and genomic studies (Miłobędzka and Muszyński, 2017) to help identify interesting patterns among the samples. However, PCA and MDS are underexplored for use in large phenotypic studies. We propose that these methods can clarify trait contributions to observed variances, phenotype-to-phenotype relationships and trends that change in response to components of treatment, genotype or time. Including PCA and MDS in MVApp facilitates the broader use of these methods by the scientific community.

While most studies still use linear regression models, focusing only on the contribution of traits for an average plant (Sellam & Poovammal, 2016; Sitienei et al., 2017), MVApp integrated quantile regression, which models the contribution of traits across the entire distribution of plants. Quantile regression can be used as a hypothesis-generating tool, identifying novel plant phenotypes with significant contributions to yield or stress tolerance, and the trait contribution can be estimated for individual quantiles. The use of quantile regression in the field of plant phenotyping is a novel concept. Its integration into the MVApp will help to develop a better understanding of the phenotype-to-phenotype interactions and the contribution of individual traits to the trait of major interest, e.g. survival, yield or metabolite production. The quantile-regression approach can be applied to the field of plant breeding, where understanding the traits contributing to productivity is key to the development of superior plant varieties with increased yields.

MVApp combines several existing statistical R libraries into a pipeline, presented in **Figure 9**, guiding the user through the interactive process of data curation, exploration and analysis. The graphic interface of the MVApp and the messages aim to provide better understanding and interpretation of the statistical test outputs, as well as empower the users without skills in command-line software to explore the full potential of multivariate analysis. In this manuscript, we have presented the different functionalities of MVApp, which we aim to expand in the future. We encourage the submission of new modules, suggestions and contributions from the scientific community to new releases of MVApp. Please see <https://github.com/mmjulkowska/MVApp/blob/master/CONTRIBUTING.md> for more information about how to contribute. Our goal for MVApp is that it will facilitate data analysis and statistical literacy across the scientific community by compiling different methods that are already in use across various disciplines. We hope that MVApp can unlock the potential of those methods and enhance the experience of data curation and exploration for researchers, especially those investigating phenotype-to-genotype and phenotype-to-phenotype relationships in phenotypic datasets.

**Material & Methods:**

**MVApp setup**

MVApp was written in R (Team, 2015) and its interactive user interface was made from the shiny library (Chang et al.). The interactive plots are produced from the plotly (Sievert et al., 2017), ggplot2 (Wickham, 2009) and gplots (Warnes et al., 2016) libraries. The color scales for plots are enabled using the RColorBrewer (Neuwirth et al., 2014) and colorRamps (Keitt, 2012) libraries. The data table display is based on DT library (Xie et al., 2018). Reshaping on the data tables into various formats was accomplished using reshape and reshape2 (Wickham, 2007) libraries. Users can download all the graphs in the “.pdf” format, and all the tables in the “.csv” format. The data is scaled using the scale() function. The values represented in the tables are rounded to four decimal numbers using the round() function, but are left unaltered in the table available for download. All functions are part of the stats library in R unless indicated otherwise.

**Curve fitting**

The individual samples are separated based on Sample ID, Genotype and selected independent variables. Depending on the fitted function, the trait data are not transformed (linear) or are transformed using square root, quadratic or natural logarithm functions (for quadratic, square root and exponential functions, respectively). After the transformation, the linear model is fitted using the lm() function, and the r-squared value is calculated using the summary(lm())$r.squared function. The cubic splines are calculated using lm(trait ~ bs()), while the smoothed splines are calculated using smooth.spline(). Spline functions were developed in splines library (Team, 2015). The ANOVA analysis is performed using the aov() function, while the Tukey-HSD pairwise comparison test is calculated using the HSD.test() function from the agricolae library (de Mendiburu, 2017).

**Data curation**

For outlier detection, the data are first grouped based on the selected independent variables (e.g. genotype, treatment and time). The selected independent variables are fused into one ID (e.g. genotype\_treatment\_time) and used as grouping variables to identify the outliers.

In our example, we used the boxplot()$out function for the outliers identified with 1.5\*IQR. For the Bonferroni test, we fitted a linear model with lm(), followed by car::outlierTest() from the car library (Fox and Weisberg, 2011). To identify outliers using the Cook’s distance, we fitted the linear model with lm(), followed by the cooks.distance() function from the base library (Team, 2015).. The samples having a Cook’s distance larger than four times the mean Cook’s distance were classified as outliers. To identify outliers by their +/- SD from the median, we calculated the median and standard deviation using summaryBy(), median() and sd() functions, respectively, from the doBy library (Hojsgaard and Halekoh, 2018) per pre-defined grouping variables (e.g. genotype, treatment, time), then subsetted the dataset for pre-defined grouping variables and identify the individual values outside of median +/- SD range for that specific subset.

**Summary statistics**

The summary statistics are calculated using the summaryBy() function from the doBy library (Hojsgaard and Halekoh, 2018). The mean, standard deviation, standard error, min, max, sum and number of samples are calculated using the mean(), median(), sd(), std.error(), min(), max(), sum() and length() functions, respectively, from the doBy (Hojsgaard and Halekoh, 2018) and plotrix libraries (Lemon, 2006).

**Hypothesis testing using parametric and non-parametric tests**

The normal distribution of data is tested using the Shapiro-Wilk test using the shapiro.test() function and the QQ-plots are produced using the qqnorm() and qqline() functions. The equal variance is tested using Bartlett and Levene tests by applying the bartlett.test() and leveneTest() functions from the car library (Fox and Weisberg, 2011). The equal variation is represented visually using the hovPlot.bf() function from the HH library (Heiberger, 2018). One- and two-sample t-tests are performed using the t.test() function, while the Kolmogorov-Smirnov test is performed using ks.test(). The ANOVA analysis is performed using the aov() function, while the Tukey-HSD pairwise comparison test is calculated using the HSD.test() function. The non-parametric Kruskal-Wallis test is performed using kruskal.test(), while the pairwise Wilcoxon/Mann-Whitney test is executed using pairwise.wilcox.test(). The two-way ANOVA analysis is performed using the linear model lm(), followed by the anova() function. The two-way ANOVA interaction plot is produced by interaction.plot(), while the residual plot is produced by plotting lm()$fitted versus lm()$residual.

**Correlation analysis**

For our data, we made the correlation plot using the corrplot() function from the corrplot library (Wei and Simko, 2017). The correlation coefficient was calculated by the selected input method (Pearson or Spearman), and the r-squared and p-values were extracted from the rcorr() functions. The correlation significance test was performed by cor.mtest().

**Reduction of dimensionality**

We performed our principal component analysis (PCA) using the PCA() function from the factoextra (Kassambara and Mundt, 2017) and FactoMineR (Le et al., 2008) libraries. The eigenvectors were extracted by using PCA$eig, and the PCA contribution plot was made using the fviz\_pca\_var() function. The contribution of individual traits for each selected PC was calculated using PCA$var$contrib, while the plot was made using the fviz\_contrib() function. Multidimensional scaling (MDS) was performed using the dist(), cmdscale() and as\_tibble() functions, and the k-means clustering of the MDS results was performed using the kmeans()$cluster function. MDS of the measured traits was performed by transposing the dataset using the t() function and completing the MDS analysis as described above. Thus, we were able to examine the dimensional relationships among the traits rather than among the samples.

**Cluster analysis**

For hierarchical clustering in MVApp, the correlations among the samples in terms of the chosen traits are calculated using the cor() function. Then, the dist() function is used to determine the distances between the samples. The hierarchical analysis is performed on the transposed dataset using the t() function in order to determine the relationship among the selected traits with functions from pvclust (Suzuki and Shimodaira, 2015) and NbClust (Charrad et al., 2014) libraries. The hierarchical clustering is performed using the hclust() function. The heatmap is produced using the heatmap.2() function from the gplots library (Warnes et al., 2016), scaled per row. The dendrogram is produced from the output of the hclust() function using the plot(as.dendrogram()) function. The number of clusters is determined by cutting the dendrogram at a user-identified distance using the cutree() function. The significant differences among individual clusters are identified by the ANOVA and Tukey-pairwise comparison tests using the aov() and glht()functions from the multcomp library (Hothorn et al., 2008), followed by the cld() function, which allows letters indicating significant differences among the predefined groups to be integrated into a box plot.

For k-means clustering in MVApp, the optimal cluster number is calculated using different methods. The elbow plot is produced using the fviz\_nbclust(method = “ws”) function from the factoextra library (Kassambara and Mundt, 2017). The silhouette plot is made using the fviz\_nbclust(method = "silhouette") function. In order to ensure that the suggested number of clusters is neither too ambitious nor too conservative, the results of thirty other indices are compared (Charrad et al., 2014), and the best number of clusters is suggested according to “majority rule” using the NbClust(distance = "euclidean", min.nc = 2, max.nc = 10, method = "kmeans") function, followed by fviz\_nbclust(). Based on these results, the user select number of the clusters and the individual samples are subsequently assigned to a k-means cluster using kmeans().

**Estimation of heritability**

Broad-sense heritability is calculated as $h^{2}=\frac{V\_{G}}{V\_{p}},$ where *VG* is the genetic variance and *VP* is the total phenotypic variance. The phenotypic variance can be explicitly expressed as

$V\_{G}+ \frac{V\_{GL}}{l}+ \frac{V\_{GY}}{y}+ \frac{V\_{GLY}}{ly}+ \frac{V\_{R}}{rly}$, where *VGL*, *VGY*, *VGLY* and *VR*represent the genotype-by-location, genotype-by-year, genotype-by-location-by-year and residual variances, respectively. The number of locations, years and replicates (within location and year) are represented by *l*, *y* and *r*, respectively, and these values are input by the user. The variance components are estimated by using the VarCorr function on the fitted linear mixed model lmer(Trait ~ 1 + (1 | Genotype) + (1 | Year) + (1 | Location) + (1 | Genotype:Location) + (1| Genotype:Year) + (1 | Genotype:Year:Location) from lme4 library (Bates et al., 2015). Heritability values are rounded to two digits.

**Quantile regression**

The quantile regression models for the traits of major interest are fitted using the rq() function from the quantreg library (Koenker et al., 2017) for different quantile levels. The estimated values of the regression coefficients and the p-values are extracted from the summary of the fitted model using summary(rq())$coefficients[,"Value"] and summary(rq())$coefficients[,"Pr(>|t|)"], respectively. The quantile plots (Agarwal et al., in review) are produced by plotting the estimated regression coefficients against the quantile level using the plot() and lines() functions, and the legends are produced using the legend() function.

**Author contributions:**

MMJ designed the project, supervised the design of the MVApp tabs, wrote the first draft of the manuscript, and designed the data upload, curve fitting, outlier removal and hierarchical clustering tabs; SS and MM designed the data exploration tab; SS designed the heritability tab; GG designed the data correlation analysis tab; MA designed the reduction of dimensionality tab; YP designed the k-means clustering tab; and GA designed the quantile regression tab. All authors contributed by writing sections of the MVApp README page (https://mmjulkowska.github.io/MVApp/), filming YouTube video tutorials on the MVApp channel (<https://www.youtube.com/channel/UCeTCqj3dHWbjIbt9cXVjHMQ>) and editing the manuscript. MT provided the funds and resources for the project and edited the final version of the manuscript.

**Acknowledgements:**

The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST), through both baseline support to MT and under Office of Sponsored Research (OSR) Award No. 2302. Figure 9 was produced by Ivan Gromicho, scientific illustrator at KAUST. We would like to thank Antonio Arena from Research Computing at King Abdullah University of Science and Technology (KAUST) for his help with putting MVApp on the server and making it accessible online; KAUST IT Linux Systems Team who provided the infrastructure for the online hosting of MVApp; and Veronica Tremblay, scientific editor at KAUST, for editing the manuscript. Additionally, we would like to thank Dr. Guillaume Lobet (Louvain / Jurlich University), Dr. Sandra Schm[ö](https://en.wiktionary.org/wiki/%C3%B6)ckel and Dr. Boubacar Kountche (KAUST), Prof. Julia Bailey-Serrez (UC Riverside) and Dr. Nazgol Emrani (Kiel University) for their helpful comments on the MVApp design and functionality.

**Competing financial interests:**

The authors declare no competing financial interests. MVApp is free and publicly available.

**References:**

**Agarwal, G., Saade, S., Shahid, M., Tester, M., and Sun, Y.** (2019). Quantile function modeling applied to analysis of salinity tolerance of plants. *In review.*

**Altman, N. and Krzywinski, M.** (2016). Analyzing outliers: influential or nuisance? Nat Meth **13**: 281–282.

**Awlia, M., Nigro, A., Fajkus, J., Schmoeckel, S.M., Negrão, S., Santelia, D., Trtílek, M., Tester, M., Julkowska, M.M., and Panzarová, K.** (2016). High-Throughput Non-destructive Phenotyping of Traits that Contribute to Salinity Tolerance inArabidopsis thaliana. Front Plant Sci **7**: 1414.

**Bates, D., Maechler, M., Bolker, B., Walker, S.** (2015). Fitting Linear Mixed-Effects Models Using lme4. Journal of Statistical Software, 67(1), 1-48. doi:10.18637/jss.v067.i01.

**Chaloner , K. and Brant, R.** (1988). A Bayesian approach to outlier detection and residual analysis. Biometrika **75**: 651–659.

**Chang, W., Cheng, J., Allaire, J.J., Xie, Y., and McPherson, J.** (2017) Shiny: Web Application Framework for R. R package version 1.0.4. https://CRAN.R-project.org/package=shiny

**Charrad, M., Ghazzali, N., Boiteau, V., Niknafs, A.** (2014). NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. Journal of Statistical Software, 61(6), 1-36. URL http://www.jstatsoft.org/v61/i06/.

**Chen, D., Fu, L.-Y., Hu, D., Klukas, C., Chen, M., and Kaufmann, K.** (2018). The HTPmod Shiny application enables modeling and visualization of large-scale biological data. Commun Biol **1**: 89.

**Dowle, M., Srinivasan, A.** (2017). data.table: Extension of `data.frame`. R package version 1.10.4-3. https://CRAN.R-project.org/package=data.table

**Fahlgren, N., Gehan, M.A., and Baxter, I.** (2015). Lights, camera, action: high-throughput plant phenotyping is ready for a close-up. Current Opinion in Plant Biology **24**: 93–99.

**Fox, J. Weisberg, S.** (2011). An {R} Companion to Applied Regression, Second Edition. Thousand Oaks CA: Sage. URL: http://socserv.socsci.mcmaster.ca/jfox/Books/Companion

**Heiberger, R.M. (2018).** HH: Statistical Analysis and Data Display: Heiberger and Holland. R package version 3.1-35. URL https://CRAN.R-project.org/package=HH

**Hojsgaard, S., Halekoh, U. (2018).** doBy: Groupwise Statistics, LSmeans, Linear Contrasts, Utilities. R package version 4.6-2. https://CRAN.R-project.org/package=doBy

**Hothorn, T., Bretz, F., Westfall, P.** (2008). Simultaneous Inference in General Parametric Models. Biometrical Journal 50(3), 346--363.

**Howe, D. et al.** (2008). Big data: The future of biocuration. Nature **455**: 47–50.

**Kassambara, A., Mundt, F.** (2017). factoextra: Extract and Visualize the Results of Multivariate Data Analyses. R package version 1.0.5. https://CRAN.R-project.org/package=factoextra

**Keitt, T.** (2012). colorRamps: Builds color tables. R package version 2.3. https://CRAN.R-project.org/package=colorRamps

**Koenker, R. (2017).** quantreg: Quantile Regression. R package version 5.34. https://CRAN.R-project.org/package=quantreg

**Le, S., Josse, J., Husson, F.** (2008). FactoMineR: An R Package for Multivariate Analysis. Journal of Statistical Software, 25(1), 1-18. 10.18637/jss.v025.i01

**Lemon, J.** (2006) Plotrix: a package in the red light district of R. R-News, 6(4): 8-12.

**Lever, J., Krzywinski, M., and Altman, N.** (2017). Points of Significance: Principal component analysis. Nat Meth **14**: 641–642.

**Leys, C., Ley, C., Klein, O., Bernard, P., and Licata, L.** (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. Journal of Experimental Social Psychology **49**: 764–766.

**Li, Y. and Andrade, J.** (2017). DEApp: an interactive web interface for differential expression analysis of next generation sequence data. Source Code Biol Med **12**: 2.

**Marx, V.** (2013). Biology: The big challenges of big data. Nature **498**: 255–260.

**Matias, F.I., Granato, I., and Fritsche-Neto, R.** (2018). Be-Breeder: an R/Shiny application for phenotypic data analyses in plant breeding. Crop Breeding and Applied Biotechnology **18**: 241–243.

**McCarthy, D.J., Campbell, K.R., Lun, A.T.L., and Wills, Q.F.** (2017). Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. Bioinformatics **33**: 1179–1186.

**de Mendiburu, F.** (2017). agricolae: Statistical Procedures for Agricultural Research. R package version 1.2-8. https://CRAN.R-project.org/package=agricolae

**Miłobędzka, A. and Muszyński, A.** (2017). Can DNA sequencing show differences between microbial communities in Polish and Danish wastewater treatment plants? Water Sci. Technol. **75**: 1447–1454.

**Morota, G.** (2017). ShinyGPAS: Interactive genomic prediction accuracy simulator based on deterministic formulas. Genet. Sel. Evol. **49**: 91.

**Nelson, J.W., Sklenar, J., Barnes, A.P., and Minnier, J.** (2017). The START App: a web-based RNAseq analysis and visualization resource. Bioinformatics **33**: 447–449.

**Neuwirth, E.** (2014). RColorBrewer: ColorBrewer Palettes. R package version 1.1-2. https://CRAN.R-project.org/package=RColorBrewer

**Reiser, L., Harper, L., Freeling, M., Han, B., and Luan, S.** (2018). FAIR: A Call to Make Published Data More Findable, Accessible, Interoperable, and Reusable. Molecular Plant.

**Seren, Ü., Vilhjálmsson, B.J., Horton, M.W., Meng, D., Forai, P., Huang, Y.S., Long, Q., Segura, V., and Nordborg, M.** (2012). GWAPP: a web application for genome-wide association mapping in Arabidopsis. The Plant Cell **24**: 4793–4805.

**Sellam, V., Poovammal, E.** (2016). Prediction of Crop Yield using Regression Analysis. Ind. J. of Sci. and Tech. 9(38), 10.17485/ijst/2016/v9i38/91714

**Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., Ram, K., Corvellec, M., Despouy, P.** (2017). plotly: Create Interactive Web Graphics via 'plotly.js'. R package version 4.7.1. https://CRAN.R-project.org/package=plotly

**Sitienei, B., Juma, S., and Opere, E.** (2017). On the Use of Regression Models to Predict Tea Crop Yield Responses to Climate Change: A Case of Nandi East, Sub-County of Nandi County, Kenya. Climate 2017, Vol. 5, Page 54 **5**: 54.

**Spitzer, M., Wildenhain, J., Rappsilber, J., and Tyers, M.** (2014). BoxPlotR: a web tool for generation of box plots. Nat Meth **11**: 121–122.

**Suzuki, R., Shimodaira, H.** (2015). pvclust: Hierarchical Clustering with P-Values via Multiscale Bootstrap Resampling. R package version 2.0-0. https://CRAN.R-project.org/package=pvclust

**Team, R.C.** (2015). R: A language and environment for statistical computing.

**Warnes, G.R., Bolker, B., Bonebakker, L., Gentleman, R., Liaw, W.H.A., Lumley, T., Maechler, M., Magnusson, A., Moeller, S., Schwartz, M., Venables, B.** (2016). gplots: Various R Programming Tools for Plotting Data. R package version 3.0.1. https://CRAN.R-project.org/package=gplots

**Wei, T., Simko, V.** (2017). R package "corrplot": Visualization of a Correlation Matrix (Version 0.84). Available from https://github.com/taiyun/corrplot

**Wickham, H.** (2007). Reshaping Data with the reshape Package. Journal of Statistical Software, 21(12), 1-20. URL http://www.jstatsoft.org/v21/i12/.

**Wickham, H.** (2009) ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York.

**Wilkinson, M.D. et al.** (2016). Comment: The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data 2016 3 **3**: 160018.

**Xia, J., Sinelnikov, I.V., Han, B., and Wishart, D.S.** (2015). MetaboAnalyst 3.0-making metabolomics more meaningful. Nucleic Acids Research **43**: W251–W257.

**Xie, Y., Cheng, J., Tan, X.** (2018). DT: A Wrapper of the JavaScript Library 'DataTables'. R package version 0.5. https://CRAN.R-project.org/package=DT

**Xu, L., Cruz, J.A., Savage, L.J., Kramer, D.M., and Chen, J.** (2015). Plant photosynthesis phenomics data quality control. Bioinformatics **31**: 1796–1804.

**Yano, R., Nonaka, S., and Ezura, H.** (2018). Melonet-DB, a Grand RNA-Seq Gene Expression Atlas in Melon (Cucumis melo L.). Plant and Cell Physiology **59**: e4–e4.

**Zhang, J., Luo, W., Zhao, Y., XU, Y., Song, S., and CHONG, K.** (2016). Comparative metabolomic analysis reveals a reactive oxygen species-dominated dynamic model underlying chilling environment adaptation and tolerance in rice. New Phytologist **211**: 1295–1310.