## Experimental section

### MCnebula algorithm

**Overview**. The analysis of untargeted LC-MS/MS datasets typically starts with feature detection. Annotating these features is time-consuming and extracting useful information for further biological research can also be challenging. To address these issues, we present MCnebula, with an abundance-based classes (ABC) selection algorithm, to boost mass spectrometry data analysis by focusing on critical chemical classes and visualization in multiple dimensions.

**Molecular formula and chemical structure prediction**. The analysis of MS/MS spectra involve a process of inference and prediction. We deduce the molecular formula based on the molecular weight of elements composition and possible fragmentation pattern of the MS/MS spectrum by SIRIUS; We then search for the exact chemical structure from compound structure databases. However, this process is often uncertain as several factors can affect the reliability of MS/MS data and the correctness of inference. SIRIUS generates a list of candidates for the potential molecular formula, chemical structure, and chemical classification behind each MS/MS spectral feature. In such cases, MCnebula extract these candidates and determine the unique molecular formula and chemical structure for each MS/MS spectrum based on the highest score of prediction.

**Top candidate selection by multiple score systems**. After the process of compound prediction, we get candidates for molecular formula, structure, and chemical classes. Some candidates are correct while others are not. While the correct predictions for molecular formula and chemical structure are unique, the chemical classification may have multiple correct predictions belonging to different classes of hierarchy. The scoring method depends on personalized research purpose. We can use scores based on various criteria such as isotopes, mass error, structural similarity, or chemical classes to rank and filter the candidates. With numerous score systems available, we filter out low-scoring candidates and focus on those with higher scores, which are more likely to be the correct compound. However, in most cases, the top candidates from the three scoring systems are not always consistent. So, we choose a ‘specific candidate’ as a reference in the user-selected scoring system and retrieve its chemical information from the other systems for data integration. We obtain unique molecular formulae and chemical structure for the reference by scoring and ranking, but for chemical classes, more work is needed.

**Chemical classification**. Compounds with MS/MS spectrum can be classified based on their overall structure or local structure, which we refer to as the dominant structure and substructure, respectively. Therefore, in the chemical classification system, we can classify compounds not only based on their dominant structure, but also based on their substructure. When the dominant structure is unavailable, or the MS/MS spectral fragment is insufficient, we classify the compounds using substructure information to gain knowledge of the compound. Note: Classifying compounds based on their dominant structure is straightforward. For example, we classify Taxifolin as a Flavone, not a phenol, even though its local structure contains a phenol substructure. We prefer to classify compounds based on their dominant structure because it is more concise and provides more information. However, during the MS/MS spectral analysis, we sometimes can only classify compounds based on their substructure.

**ABC selection**. The ABC selection algorithm evaluates all features collectively in an untargeted LC-MS/MS dataset, by examining the number and abundance of features for each chemical classification at different levels with both substructure and dominant structure. Based on this analysis, representative classes are selected for subsequent analysis (as shown in Fig. 6).

* Create Stardust Classes (Inner filter). By using the posterior probability of classification prediction (PPCP) data for each feature, simple threshold or absolute conditions are set to filter the chemical classes, and refer to as ‘inner’ filtering.
* Cross filter Stardust Classes. This involves combining the data of the chemical classes and their classified features (i.e. Stardust Classes data), grouping them based on the chemical classes, and then performing statistics on the features within each group. Statistics may also be performed on these data in conjunction with features annotation data, or to compare groups with each other. This method involves crossover of attributes for filtering, hence it is referred to as ‘cross’ filtering. (See details in the next subsection about Cross filter Stardust Classes.)

The resulting dataset is called Nebula-Index, which records multiple chemical classes and their associated features. Each chemical class is considered as a ‘Nebula’ and its classified ‘features’ are the components of the Nebula. These Nebulae are visualized as networks, with Parent-Nebula representing all features combined and Child-Nebulae representing individual classes with their ‘features’.

**Details of Cross filter Stardust Classes.** This method involves integrating three modules, as shown in Fig. 6:

*Cross-filtering by ‘quantity’* (abundance selection): The ‘features’ quantity limitation is set for each group (a chemical class with its classified ‘features’). The minimum quantity of ‘features’ within the group and the maximum proportion of ‘features’ quantity within the group versus all ‘features’ (unique) quantity of all groups are used as thresholds. Chemical classes with too many or too few ‘features’ are filtered out.

*Cross-filtering by ‘score’* (Goodness assessment): This step associates the Stardust Classes data with ‘features’ annotation data. For each group, the Goodness assessment is performed for each target attribute (continuous attribute, generally a scoring attribute of compound identification, such as ‘Tanimoto similarity’). If the group satisfies all the expected Goodness, the chemical class is retained. The Goodness () is calculated as follows: , where is the quantity of ‘features’ of which the target attributes satisfy the cut-off, and is the quantity of all ‘features’.

The assessment of Goodness is related to the parameters of ‘tolerance’ and ‘cutoff’: the expected Goodness value of ‘tolerance’ and the actual Goodness, which is related to the parameter ‘cutoff’.

Goodness assessment can be given to multiple target attributes. Note that the chemical class is retained only if it passes the Goodness assessment of all target attributes. The main purpose of this step is to filter out chemical classes with too many ‘features’ of low structural identification.

*Cross-filtering by ‘identical’* (identicality assessment): This step involves a similarity assessment of chemical classes. A hierarchical range is set for chemical classification, and groups within this range are compared for similarity. If the classified ‘features’ of two groups are almost identical to each other, the chemical class represented by one of the groups is excluded. The degree of identicality between two groups (A and B) is assessed, as well as the value of the parameter ‘identical\_factor’ () :

: ratio of the classified ‘features’ of A belonging to B

: ratio of the classified ‘features’ of B belonging to A

: value of parameter ‘identical\_factor’

If and , the two groups are considered identical, and the group with fewer ‘features’ is discarded. The purpose of this step is to filter out classes that may incorporate each other and are similar in scope. The *in silico* prediction approach may not be able to distinguish which class the potential compound belongs to from the LC-MS/MS spectra.

**Network graph presentation**. The features and their annotations are integrated as Nebulae based on the Nebula-Index. These Nebulae are represented as network-type graph data. The feature annotation data includes top candidates for chemical formula and structure. The MS/MS spectral similarity of the features is calculated and used to generate the edge data for the network graph.

**Visualization system**. MCnebula integrates various R packages to format data, including the ‘ggplot2’ package. To make visualization easier for users, we developed the ‘ggset’ data class, which stores pre-defined ggplot2 plotting functions and parameters for visualizing Nebulae. Users can customize the visualization according to their specific needs or the requirements of the publisher.

**Statistical analysis**. MCnebula integrates the functions of the ‘limma’ package for differential expression analysis of RNA-sequence and microarray data[40], and package them for differential analysis of metabolomics data. The gene expression matrix and feature quantification matrix of LC-MS are similar, both have phenotypic variables (sample information) and dependent variables (gene expression or feature quantification values). Our method can be appropriate for statistical analysis of feature quantification of experimental designs in which explanatory variables are factorial variables and the design matrix is without an intercept [41].

**Data structure**. MCnebula was primarily developed using the R S4 system of object-oriented programming. All data including ‘features’ annotation data and visualization data is stored in a single object (class ‘mcnebula’), which simplifies the application, makes data management and analysis easier to perform and repeat.

**Reporting system**. MCnebula includes a reporting system that enables the analysis process to be output as a PDF document or in other formats. The reporting system is based on the ‘report’ data class, which stores each step of the analysis as a section and can be easily modified according to the user’s requirements. Furthermore, the ‘rmarkdown’ R package [43] is incorporated in the reporting system to generate reports.

**Code Compatibility**. MCnebula performs downstream analysis by extracting data from the pre-computed SIRIUS project, which is the primary data source for MCnebula 2. The SIRIUS software is continually updated and enhanced. From SIRIUS version 4 to version 5 (https://github.com/boecker-lab/sirius), the data structure and attribute names in the project directory have been modified. To ensure that MCnebula is not affected by version problems, we have designed its application programming interface (API) for various SIRIUS versions.

### MCnebula evaluation

**Spectra dataset for evaluation**. To evaluate the performance of MCnebula, the spectra from the GNPS MS/MS library was used (http://prime.psc.riken.jp/compms/msdial/main.html#MSP). To prevent overfitting during library match evaluation, ‘noise’ was added to the MS/MS spectra[44]. Two models of noise were simulated: medium noise and high noise. The simulation involved a global mass shift, individual mass deviations, intensity variations, and additional ‘noise peaks.’ Isotope patterns were also simulated using the ‘get.isotopes.pattern’ function within the ‘rcdk’ R package[45]. The mass and abundance of isotopes were considered for the adduct type to increase or decrease exact mass. The ‘isotope peaks’ were merged into the MS1 list of compounds, and all spectra collections were formatted into mgf or csv file for the continuing MCnebula and benchmark analysis.

**Evaluation method.** MCnebula and benchmark workflow was conducted for all the three simulated datasets. SIRIUS 4 command-line interface (CLI) (version 4.9.12) was applied for computation, and filtered out MS/MS spectra with empty fragmentation peaks. In total 7524 out of 8782 compounds were left for evaluation. ClassyFire was used to assess the classification accuracy[24]. After *in-silico* annotation, we obtained structure annotation, International Chemical Identifier Key (InChIKey), and other metadata of these compounds. Considering that ClassyFire only supports chemical identities those structures have been classified in its server previously, we used the first hash block of InChIKeys (InChIKey planar, which represents the molecular skeleton) to query the PubChem application programming interface (API) (https://pubchemdocs.ncbi.nlm.nih.gov/pug-rest) [46]. This provided us with all the possible InChIKeys of isomerism (stereo, isotopic substitution) [47]. Classification of small molecules depends on their molecular skeleton, so chemical identities that share the same InChIKey planar are identical in classification. The InChIKey list was imported into ClassyFire to obtain chemical classification. In our R script, once any InChIKey of isomerism matched the classified data in the database, we turned off the acquisition status for this molecular skeleton. In the end, we collated, integrated, and assigned all these chemical annotations as standard reference.

Due to differences in algorithms and classified results, we evaluated the MCnebula and benchmark methods separately. Since sub-structural classification was not available for the benchmark method, we excluded these classes during the evaluation analysis. Nevertheless, some compounds within the remaining classes may still be classified into sub-structural classes. We assigned three levels for evaluation: ‘True,’ ‘Latent,’ and ‘False.’ ‘True’ indicated that the classified classes were consistent with those of ClassyFire. ‘Latent’ indicated that the classified classes were not consistent with ClassyFire, but their parent classes at the ‘class’ level were consistent. ‘False’ indicated that the classified classes were completely inconsistent with ClassyFire. Then, the true positive (TP) was assigned as: ‘True’ + ‘Latent’; the false positive (FP) was assigned as ‘False’.

To evaluate the identification of classes or structures, we merged the results with a standard reference by InChIKey planar. For the evaluation of chemical structure identification, we considered a structure as ‘True’ if it matched the chemical structure identified by InChIKey planar. However, this evaluation neglected stereochemistry due to the limit of LC-MS/MS detection.

### Other information

More methodological details regarding the MCnebula assessment, handling of samples in the study (serum and phytochemicals), data processing, and acquisition of available data and codes are documented in the supplemental file.