FEBS openbio

FEBS PRESS science publishing by scientists

METHOD

MVP – an open-source preprocessor for cleaning duplicate records and missing values in mass spectrometry data

Geunho Lee[1], Hyun Beom Lee[2], Byung Hwa Jung[2] and Hojung Nam[1]

1 School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST), Korea
2 Molecular Recognition Research Center, Korea Institute of Science and Technology (KIST), Seoul, Korea

Keywords
dirty data; duplicate record; mass spectrometry; missing value; MS data preprocessor; R package

Correspondence
B. H. Jung, Molecular Recognition Research Center, Korea Institute of Science and Technology (KIST), 5, Hwarang-ro 14-gil, Seongbuk-gu, Seoul 02792, Korea
Tel: +82 2 958 5062
E-mail: jbhluck@kist.re.kr
and
H. Nam, School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST), 123, Cheomdangwagi-ro, Buk-gu, Gwangju 61005, Korea
Tel: +82 62 715 2641
E-mail: hjnam@gist.ac.kr

(Received 7 February 2017, revised 27 April 2017, accepted 15 May 2017)

doi:10.1002/2211-5463.12247

Mass spectrometry (MS) data are used to analyze biological phenomena based on chemical species. However, these data often contain unexpected duplicate records and missing values due to technical or biological factors. These ‘dirty data’ problems increase the difficulty of performing MS analyses because they lead to performance degradation when statistical or machine-learning tests are applied to the data. Thus, we have developed missing values preprocessor (MVP), an open-source software for preprocessing data that might include duplicate records and missing values. MVP uses the property of MS data in which identical chemical species present the same or similar values for key identifiers, such as the mass-to-charge ratio and intensity signal, and forms cliques via graph theory to process dirty data. We evaluated the validity of the MVP process via quantitative and qualitative analyses and compared the results from a statistical test that analyzed the original and MVP-applied data. This analysis showed that using MVP reduces problems associated with duplicate records and missing values. We also examined the effects of using unprocessed data in statistical tests and examined the improved statistical test results obtained with data preprocessed using MVP.

Mass spectrometry (MS) data are widely used to analyze various biological phenomena by producing mass spectra patterns for the associated chemical or biological species, such as compounds, metabolites, peptides, and proteins. Analyses of MS data consist of ionization analyses, mass analyses, and chemical species detection [1]. MS measures the mass-to-charge ratio (m/z) of an ionized chemical species, and when MS is coupled to a second MS, known as tandem MS or MS/MS, it allows the detection of the fragment ions of a selected ion that is an identifier for a molecule. Interpreting MS results enables researchers to biologically interpret many domains. MS data are widely used in proteomics, metabolomics, and drug development [2–4]. For example, researchers can utilize MS to analyze complex protein mixtures, identify metabolites, or discover target drugs. Over time, the performance of MS systems has continually improved, which has allowed for the analysis of as-yet unanalyzed chemical species. Many types of MS systems have been developed, including time-of-flight (ToF) and Orbitrap spectrometers [5,6]. Each mass analyzer has different

Abbreviations
AMI, acute myocardial infarction; *m/z*, mass-to-charge ratio; MI, multiple imputation; MS, mass spectrometry; MVP, missing values preprocessor; ToF, time-of-flight.

operating principles and a different mass resolution. In this study, we focused on data generated from an LC-ToF/MS machine and used these data to verify the performance of our open-source software.

Unintended duplicate records and missing values are representative dirty data problems associated with MS data and should be mitigated to improve the data analysis. The process of handling dirty data is also called data cleansing, and it has been performed in statistics and data science, when dealing with large numbers of records, tabular data, or databases. We often encounter duplicate records that originate from identical molecules and missing values in the measured intensity signals in MS data sets. These problems frequently occur for technical or biological reasons [7]. These dirty data problems can affect the power of statistical and machine-learning tests [8]. Therefore, methods have been developed to alleviate these problems. Duplicate detection is generally identified faster if the data are sorted by a key. We used this property to detect candidate duplicate records in our research. Compared with duplicate detection, many approaches have been developed to handle missing values. Simple methods of handling missing values include filling them in with '0's or imputing the missing data point with the mean value. However, these simple methods can produce biased values and results [9,10]. The k-nearest neighbor (k-NN) approach is a popular imputation method with good performance [7,8]. However, k-NN has a disadvantage in that its performance depends on the number of complete records. Another common approach for efficiently imputing missing values is the multiple imputation (MI) method. MI is an efficient method, and its performance has been verified in several studies [11,12]. Recently, machine-learning methods, including naïve Bayesian-, neural network-, and decision tree-based imputation methods, have also been widely used in various domains [13–15]. However, these methods focus only on imputing missing values. With tabular data, which is the most common structured data form for data analysis, dirty data problems might include both unintended duplicate records and missing values. The methods described above are not dedicated methods for MS data and thus cannot effectively handle unintended duplicate records. Additionally, the previously discussed missing value imputation methods rely on statistics or mathematics without considering the basic domain properties of the MS data.

To properly handle problems of dirty data in MS data, methods are required that can preprocess duplicate records and missing values while also considering the MS data characteristics. We considered the key identifier in MS data as the implementation core for the missing values preprocessor (MVP) open-source platform. Key identifiers of chemical species are observed in MS data, such as the *m/z*, retention time, and intensity. MVP uses the property of MS data wherein identical chemical species have the same or similar values for key identifiers. In other words, similar *m/z* values will be accompanied by similar intensities, and this property can be used to determine whether or not chemical species are identical [16]. Based on this idea, MVP can detect and process duplicate records by examining the similar values of key identifiers for individual chemical species. MVP merges duplicate records into one record because it considers each duplicate record as an identical chemical species. Thus, in the merging process, certain intensity columns that have missing values can be imputed if the complete values of other records are available. Figure 1 shows the overall MVP process and illustrates the previously explained chemical species properties. The processing flow of MVP includes the following four separate steps: (a) extracting duplicate candidate records by using *m/z* and the retention time, (b) calculating the pairwise record similarity of the intensities in the candidate group, (c) converting record information to a graph structure, and (d) merging similar records into one record. Detailed descriptions of each step are provided in the Materials and methods section.

Materials and methods

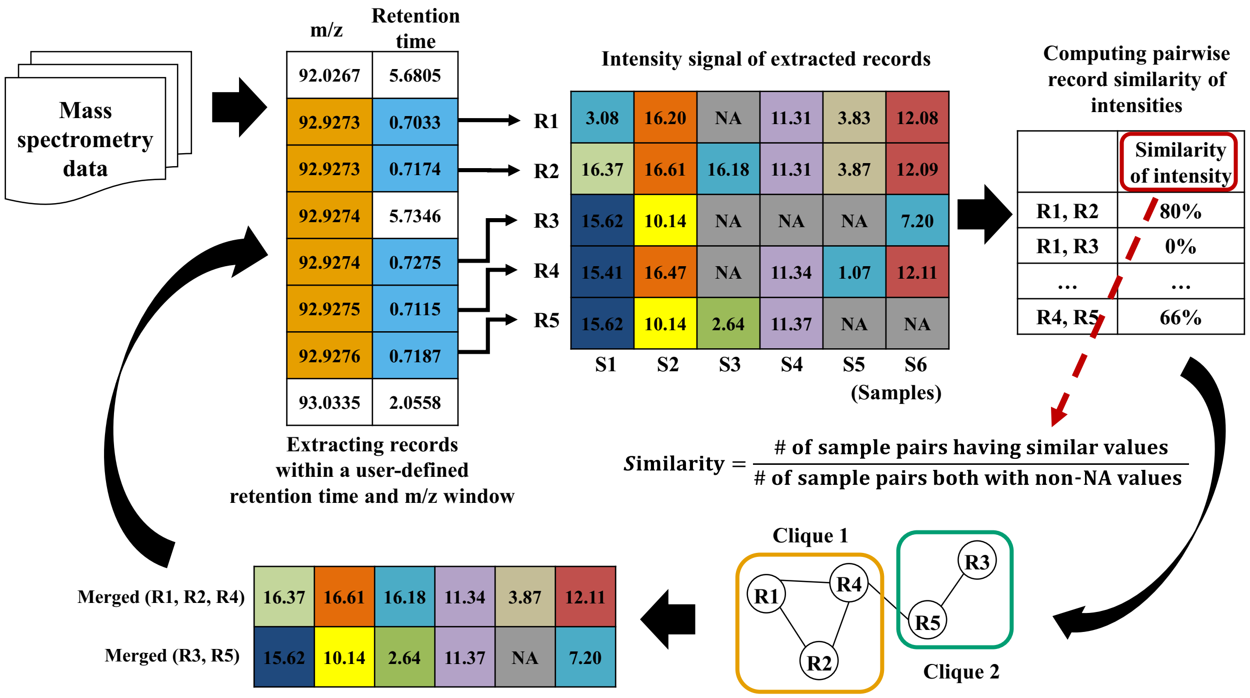
Materials

Sample collection

The study subjects were sorted into the following four groups: (a) control (n = 40), (b) acute myocardial infarction (AMI, n = 42), (c) angina pectoris (n = 72), and (d) unstable angina (n = 23). Serum samples were obtained by placing blood samples collected from the antecubital vein of the forearm into serum-separating tubes and centrifuging for 10 min at 956 ***g***.

Sample preparation

For protein precipitation, an ice-cold methanol/water (3 : 1) solution was added to the serum and centrifuged at 20 817 ***g*** for 15 min at 4 °C. The supernatant was diluted with distilled water to half the volume of the supernatant, and it was then injected in the ultra-performance liquid chromatography (UPLC) (ACQUITY®; Waters, Milford, MA, USA) coupled with a QToF-MS (SYNAPT™ G2; Waters).


Mass spectrometry data
Extracting records within a user-defined retention time and m/z window
Intensity signal of extracted records
(Samples)
Computing pairwise record similarity of intensities
Clique 1
Clique 2
R1
R4
R2
R3
R5

m/z	Retention time	
92.0267	5.6805	
92.9273	0.7033	R1
92.9273	0.7174	R2
92.9274	5.7346	
92.9274	0.7275	R3
92.9275	0.7115	R4
92.9276	0.7187	R5
93.0335	2.0558	

	S1	S2	S3	S4	S5	S6
R1	3.08	16.20	NA	11.31	3.83	12.08
R2	16.37	16.61	16.18	11.31	3.87	12.09
R3	15.62	10.14	NA	NA	NA	7.20
R4	15.41	16.47	NA	11.34	1.07	12.11
R5	15.62	10.14	2.64	11.37	NA	NA

	Similarity of intensity
R1, R2	80%
R1, R3	0%
…	…
R4, R5	66%

$$\mathit{Similarity} = \frac{\text{\# of sample pairs having similar values}}{\text{\# of sample pairs both with non-NA values}}$$

Merged (R1, R2, R4)	16.37	16.61	16.18	11.34	3.87	12.11
Merged (R3, R5)	15.62	10.14	2.64	11.37	NA	7.20

Fig. 1. Overview of MVP processing. The MVP procedures include extraction and merging steps. Using MS data identifiers, MVP extracts identical record candidates. After calculating the pairwise similarity of all candidates, MVP constructs a graph and identifies cliques. The clique record components are merged by MVP using user-specified parameters.

Data collection

All samples were analyzed in a randomized order in both positive and negative ionization modes. The raw MS data were obtained by MASSLYNX software version 4.1 and MARKERLYNX software (Waters) after deconvolution, which included peak detection, alignment, and normalization.

Development process

Description of overall process and user-specified parameters of MVP

MVP provides various user-specified variables. First, MVP supports different window sizes for m/z and the retention time. In Fig. 1, because the window size of the retention time is 0.02, only the five records colored in blue that satisfy this condition were extracted. Using the data in Fig. 1, MVP first extracts the largest clique composed of R1, R2, and R4. After extracting the candidate records using m/z and the retention time, MVP calculates the possible pairwise similarity of the record intensity. The pairwise similarity of the record intensity is calculated as the number of samples having similar intensity values divided by the number of samples having both non-NA values. In MVP, the definition of similarity of intensity is when the difference is within ±5%. For example, if the intensity value of record 1 is 100, and record 2 has an intensity between 95 and 105, they are judged to have similar values. In Fig. 1, R1 and R2 have four similar intensity values (S2, S4, S5, and S6) and five complete intensity values (S1, S2, S4, S5, and S6). Thus, the similarity of R1 and R2 is 4/5. The similarity of R4 and R5 can be computed in the same way, with two similar intensity values (S1 and S4) and three complete intensity values (S1, S2, and S4), yielding 2/3. The merging process continues for the extracted cliques. MVP uses one method, such as the maximum, median, mean, or minimum, when merging the component records in the clique to one record, and this method can be specified by the user. For example, when the maximum method is applied, the value of S1 in clique 1 is 15.41 because 15.41 is the maximum value among 3.08, 10.53, and 15.41. Similarly, S3 in clique 1 is 16.18 because 16.18 is the maximum value excluding NA. From the calculation of S3 in clique 1, we can identify how MVP interpolates the data for the missing value problem.

Duplicate candidate detection by identifiers (m/z, retention time)

In the first step, to extract duplicate candidate records, MVP rearranges the input tabular data using a key identifier of the chemical species, such as the m/z or retention time. Table 1 shows an example of an input format of the MS tabular data. Users can set indices of key identifiers before executing MVP. For example, users can set column indices for m/z and retention time or set one column index based on the sorting key. Users can also assign criteria for similar key identifiers. In this step, MVP filters out a candidate group of duplicate records. The top left of Fig. 1 shows five duplicate candidate records that were extracted when the similarity criteria for m/z and retention time were 0.002 and 0.3, respectively.

Computation of pairwise record similarity of intensity

After extracting the duplicate candidate records, MVP calculates the pairwise record similarity of the intensity in the candidate group. When constructing the graph structure, these calculated pairwise record similarities and the record similarity threshold, which can be specified by the user, are used to connect the edges. The record similarity value is defined as the number of samples with similar values divided by the number of samples with no unavailable (NA) values. For example, R1 and R2 in Fig. 1 have 80% similarity because there are five samples that have no NA values (S1, S2, S4, S5, and S6) and four samples that have similar values (S2, S4, S5, and S6).

Constructing the graph form and finding possible cliques

MVP builds the graph structure based on the results from step 2. MVP considers each record in the duplicate candidate group as a vertex and makes edges that have larger pairwise similarities than the record similarity threshold specified by the user. MVP identifies all possible cliques in the graph after constructing the base graph. The right bottom of Fig. 1 shows the base graph structure when the record similarity threshold is 50%.

Merging duplicate records and imputing missing values

Finally, MVP conducts a merging procedure with regard to the constructed graph structure. MVP extracts the clique in an order from large to small. When extracting the cliques, MVP checks whether the current clique is independent, which indicates whether the component of cliques overlaps with the previously selected cliques. A detailed example is shown in the Appendix S1.

Table 1. General tabular form of MS data (m/z, mass-to-charge ratio; RT, retention time).

Compound identifier	m/z	RT	Intensity of sample_1	...	Intensity of sample_n
Compound_1	92.9273	0.7033	11.31	...	12.08
Compound_2	92.9274	0.7174	11.34	...	NA
...	...	...	...	...	...

Simulation data construction

To validate the performance of MVP in terms of qualitative analyses, we generated simulation data based on heart disease LC/ToF-MS data as explained in the material section. We generated simulation data that are similar to actual MS data, which contain duplicate records and missing values. The simulation data were generated by randomly increasing the number of records, inserting random noise into the intensity values, and randomly incorporating missing values. Performance measurements were used to compare the answer data and the results after applying MVP to the simulation data. We measured how well MVP restored the simulation data. We performed 30 repetitions and calculated the standard error to determine the reliability of the experiment.

Comparison of the statistical test performance before and after applying MVP

Additional experiments were conducted to ascertain whether the duplicate records and missing values affected the statistical testing. We assumed that preprocessing these dirty data problems would improve the performance of the statistical or machine-learning tests. For example, we evaluated whether the accuracy of classification could be improved or the number of significant metabolites can be increased when applying Student's t-test. These evaluations used LC/ToF-MS heart disease data, which represented positive and negative ion mode data with preprocessing.

Results

Quantitative analysis by interpreting the changes in duplicate records and missing value proportions

We assessed the performance of the MVP software (Computational Systems Biology Lab., School of Electrical Engineering and Computer Science (EECS), Gwangju Institute of Science and Technology (GIST), Gwangju, Korea) via quantitative and qualitative analyses. We obtained preprocessed data that contained missing values and duplicate records via MVP for the

Table 2. Record reduction rates before and after applying MVP with different record similarity variable thresholds.

	Original data	MVP-processed data (50% similarity)	MVP-processed data (70% similarity)
Positive ion mode	3908	3360 (−14.0%)	3567 (−8.7%)
Negative ion mode	18 253	13 874 (−24.0%)	15 648 (−14.3%)

quantitative analysis. The number of duplicate records was reduced after applying MVP (Table 2). Table 2 shows the number of original data records and the MVP-applied data specified by the different record similarity parameter thresholds. If the record similarity parameter threshold was assigned a small value, too many records could be eliminated, suggesting that the output may contain mistakenly merged records that are not duplicate records. Conversely, if this parameter was set to a large value, users can obtain more robust results but a relatively small record reduction advantage.

Managing missing values is another benefit of MVP as shown in Fig. 2, which is formed by four histograms from the same data already described in Table 2. The *x*-axis reflects the missing value proportion in a record, and the *y*-axis represents the number of total records corresponding to the missing value

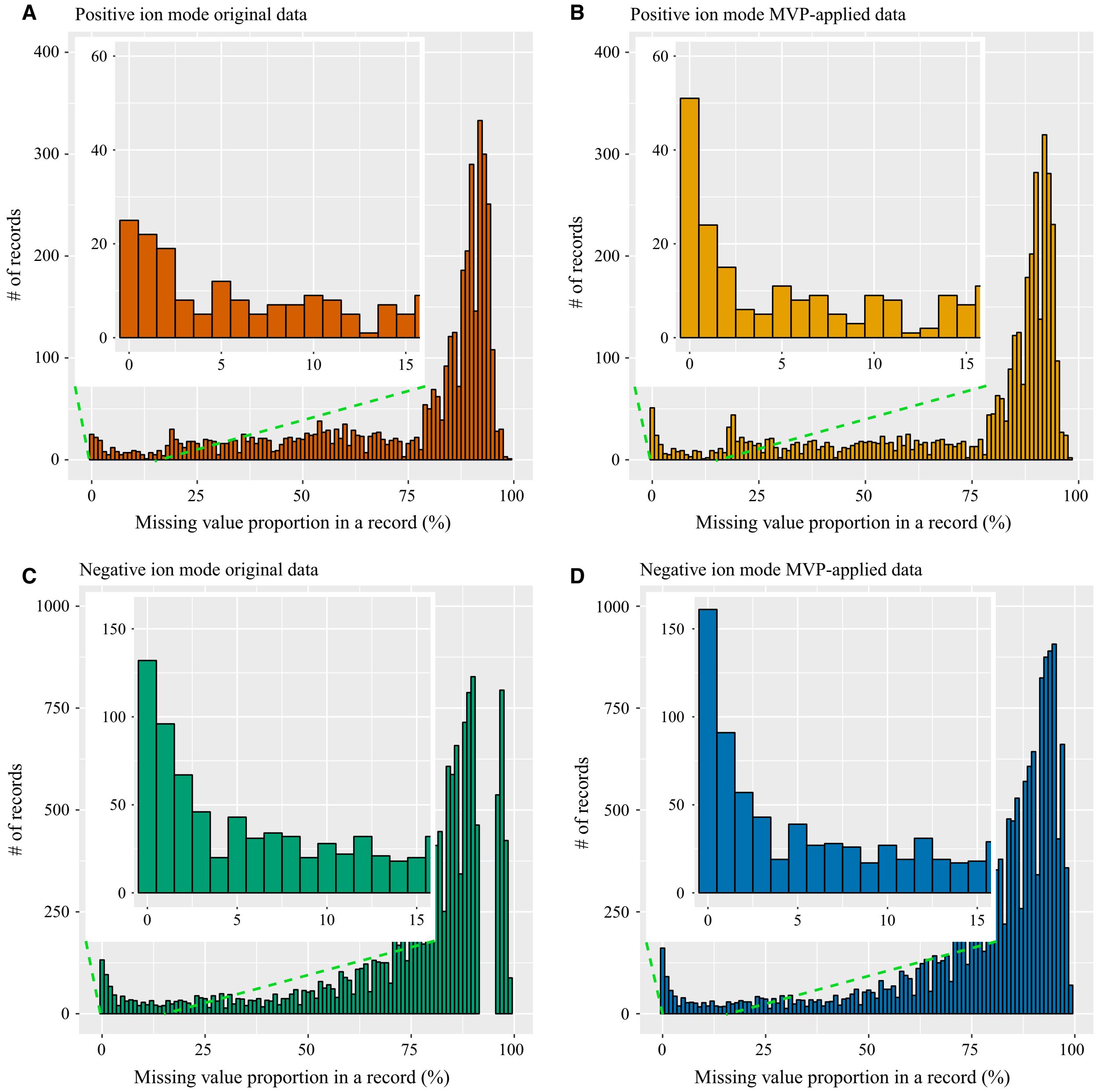


Fig. 2. Histogram for missing value problems before and after applying MVP. (A), (B) and (C), (D) Heart disease positive and negative ion mode results, respectively. (A) and (C) Original results. (B) and (D) MVP-processed results obtained using the following default parameter values: (a) *m/z* similarity threshold of 0.001, (b) retention time similarity threshold of 0.3, and (c) record similarity threshold of 70%.

proportion. Figure 2A and C shows that the original data have a large number of incomplete records, with the proportion on the *x*-axis greater than approximately 90%, and few complete records, with the proportion on the *x*-axis at 0% for the two different data sets. Figure 2B,D represents the missing value proportion after applying MVP, and the number of complete records increases from Fig. 2B,D to Fig. 2A,C. This result indicates that MVP properly manages missing values.

Accuracy assessment by qualitative test

In addition to the quantitative analysis, we also conducted a qualitative analysis via simulation testing. To obtain the simulation data, we removed all records with NA values. From the NA-removed data, we created simulation data similar to the actual MS data, which contain duplicate records and missing values. The simulation data were generated by randomly increasing the number of records, inserting random noise to the intensity values, and randomly incorporating missing values. When inserting the random noise to the intensity values, we add the value of the normal distribution with four noise intensities, which were assigned based on the standard deviation value of the normal distribution. Figure 3 suggests how well MVP recovers simulation data that have deliberately produced noise. We applied a different record similarity threshold for the two different data sets. The *x*-axis of Fig. 3 represents the parameter value record similarity threshold, and the *y*-axis represents the restoration accuracy calculated by comparing the original data, which are the source of the simulation data, and the MVP-applied data. Each experiment was conducted 30 times, and the accuracy of restoration and standard error of the mean were recorded. We found that higher noise intensity corresponded to a lower accuracy of the restoration. Moreover, the strict record similarity threshold of 70% allowed the MVP process to recover less data compared with when applying a threshold of 50%.

Statistical test results before and after applying MVP

Lastly, the numbers of significant features after applying MVP were analyzed. We examined the data sets by selecting records that had proportions of complete values greater than 95% and 90%. Finally, the following

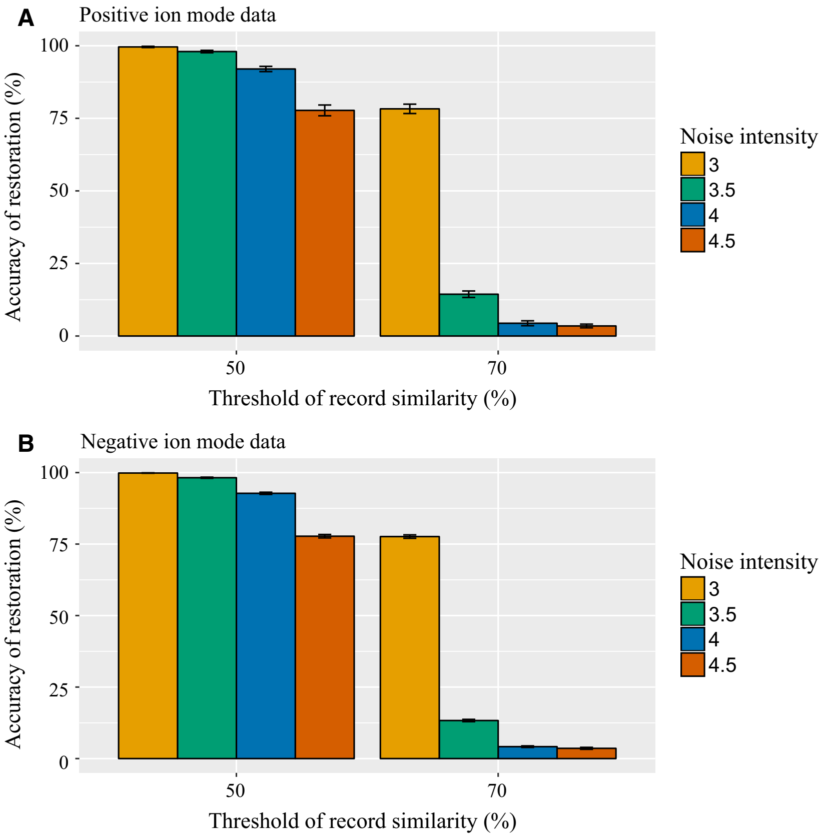


Fig. 3. MVP performance accuracy for simulation data record restoration. (A) and (B) Results based on positive and negative ion mode data for heart disease, respectively. Each simulation was tested by assigning a different record similarity and noise intensity threshold. The error bars illustrate the mean ± SEs.

four data sets were obtained: (a) positive ion mode with a 95% threshold, (b) positive ion mode with a 90% threshold, (c) negative ion mode with a 95% threshold, and (d) negative ion mode with a 90% threshold. From each data set, we generated original data and MVP-applied data. The original data were generated by applying a k-NN imputation to the previously generated data. MVP-applied data were generated by applying MVP first and then a k-NN imputation to the previously generated data. We then applied Student's *t*-test to the original and MVP-applied data and compared the results for the control and AMI groups. The *P*-value of each metabolite was calculated after applying Student's *t*-test. The FDR test (Benjamini–Hochberg procedure) was also applied to compensate for the *P*-values [17] and determine the number of significant (FDR < 0.05) metabolites. Furthermore, we also calculated the number of significant metabolites that were not identified in the original data because of duplicate records and missing values but were newly identified after preprocessing with MVP. Figure 4A–D illustrates the results of four independent experiments and indicates that significant metabolites that were not previously discovered were identified after applying MVP preprocessing.

Discussion

MS data present dirty data problems that include duplicate records and missing values, and these problems may cause the degradation of statistical tests or machine-learning algorithms. Thus, we implemented an R language-based open-source software named MVP to preprocess duplicate records and missing values in

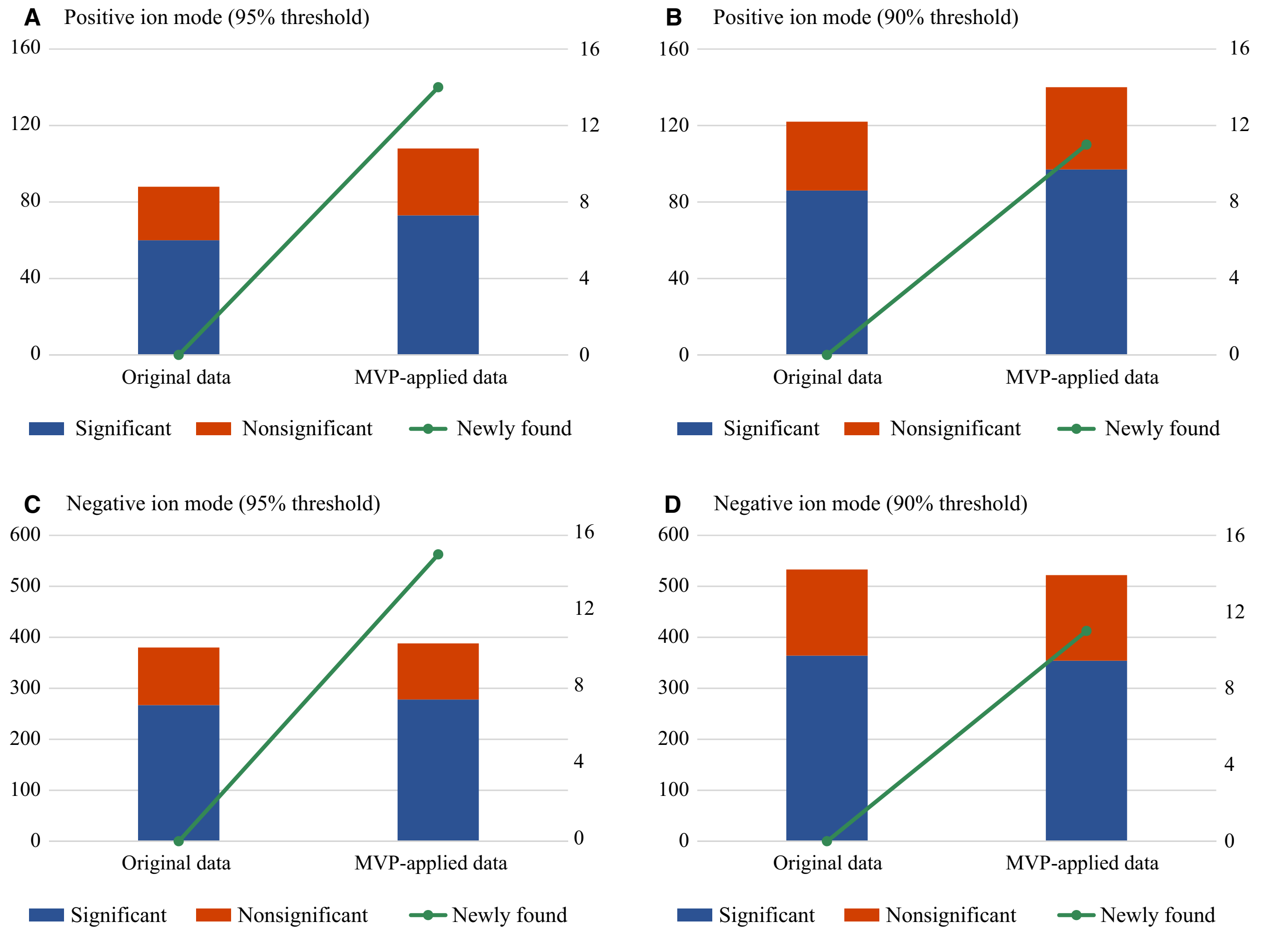


Fig. 4. Results from FDR test (Benjamini–Hochberg procedure) after Student's *t*-test before and after the application of MVP. (A), (B) and (C), (D) Results based on positive and negative ion mode data for heart disease, respectively. The height of the red box illustrates the proportion of chemical compounds with FDR < 0.05, and the sum of the heights of the blue and red boxes illustrates the total number of chemical compounds. The green line with the secondary *y*-axis shows the number of significant chemical species that were newly identified after applying MVP.

MS data. MVP uses the property of MS data wherein identical chemical species also present the same values for key identifiers, such as the *m*/*z*, intensity, and retention time.

We verified the performance of MVP via quantitative and qualitative approaches. For MS data, duplicate records and missing values were managed by MVP preprocessing. Furthermore, a comparison of the original and MVP-applied data indicated that dirty data problems could actually lead to decreased statistical test performance. These results were consistent with our assumption regarding the relationship between dirty data problems and statistical test performance degradation. In this regard, researchers who work with MS data can use MVP with various user-specified parameters to preprocess MS data before applying these data to statistical tests or machine learning.

Certain minor limitations were observed in the application of MVP because of the data types and algorithms used in this study. First, many types of MS equipment are available, including ToF/MS and Orbitrap spectrometers. Only the results for the LC/ToF-MS spectrometer are described in the main text, although we also experimented with Orbitrap data and found that MVP produced good results. An examination of dirty data in the Orbitrap data yielded different result trends compared with the ToF-MS data because the two machines have different characteristics. Another problem involves the identification of cliques in a dense graph. Finding cliques is a well-known NP-complete problem [18,19]. Thus, if the graph is too dense, then finding the cliques will take a long time. To handle this problem, MVP automatically changes the algorithm based on finding cliques. For a dense graph, MVP finds a pair of records with the best similarity and merges them, and changing the algorithm can reduce the computation time.

The main objective of MVP is merging multiple records that potentially originate from a single substance. MVP was designed to set a threshold of record similarity to fit the researcher's objective. For example, if a researcher wants to minimize false positives, the threshold of record similarity can be set to a high value such as 0.9 or 0.95, while if the researcher needs to obtain more candidates even at the risk of collecting more false positives, it is possible to set the threshold of record similarity to a low value such as 0.3 or 0.5.

When using MVP to analyze MS data, researchers need to take a careful approach depending on the resource type such as a metabolite or protein. Unlike metabolites, in the case of proteins, different peptides could have similar identifiers (*m*/*z*, retention time). Therefore, the researchers should consider the characteristics of the resource types when setting the user-specified parameters provided by MVP.

Our findings indicate that the open-source software MVP can facilitate the preprocessing of MS data with respect to duplicate records and missing values. The various validation procedures showed that MVP properly manages duplicate records and missing values. Moreover, the performance of statistical tests was improved by the application of MVP because new significant metabolites were identified after preprocessing. MVP is an open-source software that will be deposited at GitHub and CRAN to allow all users access to our software.

Acknowledgements

This work was supported by the Bio-Synergy Research Project (NRF-2014M3A9C4066449, 2013M3A9C4078145) of the Ministry of Science, ICT, and Future Planning through the National Research Foundation, supported by the National Research Foundation of Korea Grant funded by the Korea Government (MSIP) (NRF-2015R1C1A1A01051578), and supported by the GIST Research Institute (GRI) in 2017.

Author contributions

GL and HN conceived and designed the project. BHJ and HBL acquired the data used for evaluating the performance of our software. GL and HN analyzed and interpreted the results from our software. GL, HBL, BHJ, and HN wrote the manuscript. GL implemented the software.

Data accessibility

MVP is available at GitHub (https://github.com/GIST-CSBL/MVP).

References

1 Glish GL and Vachet RW (2003) The basics of mass spectrometry in the twenty-first century. *Nat Rev Drug Discov* **2**, 140–150.

2 Wysocki VH, Resing KA, Zhang Q and Cheng G (2005) Mass spectrometry of peptides and proteins. *Methods* **35**, 211–222.

3 Stobiecki M, Skirycz A, Kerhoas L, Kachlicki P, Muth D, Einhorn J and Mueller-Roeber B (2006) Profiling of phenolic glycosidic conjugates in leaves of *Arabidopsis thaliana* using LC/MS. *Metabolomics* **2**, 197–219.

4 Lee MS and Kerns EH (1999) LC/MS applications in drug development. *Mass Spectrom Rev* **18**, 187–279.

5 Boesl U (2016) Time-of-flight mass spectrometry: introduction to the basics. *Mass Spectrom Rev* **36**, 86–109.

6 Perry RH, Cooks RG and Noll RJ (2008) Orbitrap mass spectrometry: instrumentation, ion motion and applications. *Mass Spectrom Rev* **27**, 661–699.

7 Hrydziuszko O and Viant MR (2012) Missing values in mass spectrometry based metabolomics: an undervalued step in the data processing pipeline. *Metabolomics* **8**, 161–174.

8 Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D and Altman RB (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics* **17**, 520–525.

9 Donders ART, van der Heijden GJMG, Stijnen T and Moons KGM (2006) Review: a gentle introduction to imputation of missing values. *J Clin Epidemiol* **59**, 1087–1091.

10 Sterne JAC, White IR, Carlin JB, Spratt M, Royston P, Kenward MG, Wood AM and Carpenter JR (2009) Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *BMJ* **338**, b2393. https://doi.org/10.1136/bmj.b2393

11 He Y, Zaslavsky AM, Landrum MB, Harrington DP and Catalano P (2010) Multiple imputation in a large-scale complex survey: a practical guide. *Stat Methods Med Res* **19**, 653–670.

12 Parker JD and Schenker N (2007) Multiple imputation for national public-use datasets and its possible application for gestational age in United States Natality files. *Paediatr Perinat Epidemiol* **21**, 97–105.

13 Salari N, Shohaimi S, Najafi F, Nallappan M and Karishnarajah I (2014) A novel hybrid classification model of genetic algorithms, modified k-nearest neighbor and developed backpropagation neural network. *PLoS One* **9**, 1–50.

14 Oba S, Sato M, Takemasa I, Monden M, Matsubara K and Ishii S (2003) A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics* **19**, 2088–2096.

15 Nanni L, Lumini A and Brahnam S (2012) A classifier ensemble approach for the missing feature problem. *Artif Intell Med* **55**, 37–50.

16 Sparkman OD (2006) *Mass Spectrometry Desk Reference*. Global View Pub, Pittsburgh, PA.

17 Benjamini Y and Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc Ser B* **57**, 289–300.

18 Karp RM (2010) Reducibility among combinatorial problems. In *50 Years Integer Program. 1958-2008 From Early Years to State-of-the-Art* (Jünger M, Liebling TM, Naddef D, Nemhauser GL, Pulleyblank WR, Reinelt G, Rinaldi G and Wolsey LA, eds), pp. 219–241. Springer Berlin Heidelberg, Berlin.

19 Cook SA (1971) The complexity of theorem-proving procedures. *Proceedings of the Third Annual ACM Symposium on Theory of Computing*. pp. 151–158.

Supporting information

Additional Supporting Information may be found online in the supporting information tab for this article:

Appendix S1. Basic tutorial.