A Framework for Big Metabolomic Data Management and Analysis

Xiangyu Li, Leiming Yu
and David Kaeli

Department of Electrical and
Computer Engineering
Northeastern University
Boston, MA, USA
Email:{xili,ylm,kaeli}@ece.neu.edu

Yuanyuan Yao, Poguang Wang
and Roger Giese

Department of Pharmaceutical Sciences and
Barnett Institute, Bouve College
Northeastern University
Boston, MA, USA
Email:yao.yu@husky.neu.edu, {p.wang,r.giese}@neu.edu

Vicent Yusa

Department of Analytical Chemistry
University of Valencia
Burjassot, Spain
Email:yusa.vic@gva.es

Akram Alshawabkeh

Department of Civil and Environmental Engineering
Northeastern University
Boston, MA, USA
Email:aalsha@neu.edu

***Abstract*—Preterm birth is one of the major contributing factors to infant death. In the Puerto Rico Testsite for Exploring Contamination Threats Center we explore a variety of risk factors for preterm birth in Puerto Rico, including environmental, genetic and demographic factors. Given the challenge of managing such a large amount data, we have constructed a customized database specifically designed for managing our data and for facilitating efficient analysis. In this paper, we present our database design and open source Mass Spectrometry Data Analysis Toolbox. Our design allows for the efficient handling of storage and computation during metabolomic analysis. The Toolbox enables supports and end-to-end analysis protocol, from data processing and feature selection, to machine learning and visualization.**

***Keywords*–preterm birth; database; MSDA Toolbox; machine learning.**

I. Introduction

Preterm birth [1] has been identified to be a major cause of birth defects and infant deaths [2]. When an infant is delivered earlier than 37 weeks of pregnancy, the birth is considered as preterm. Research has shown that since 1990, the rate of preterm birth has been increasing worldwide, ranging from 5% to 18%. In 2010, preterm-related deaths were reported to be responsible for close to 35% of all infant deaths. A series of research findings suggests that environmental factors have a strong influence on preterm birth [3]–[8]. [9].

In our research, we focus on an area in northern Puerto Rico where the preterm birth rate is 50% higher than that in the rest of the United States. In the *Puerto Rico Testsite for Exploring Contamination Threats* (PROTECT) Center, we collaborate with a cohort of over 2000 women in northern Puerto Rico (presently 800 of the 2000 participant mothers have been recruited). We are analyzing many potential contributing factors, including environmental and biological factors, which could be linked to premature birth.

Our study is highly data driven. We collect and analyze data across a wide spectrum of sources, including:

- Environmental samples and measurements - soil samples, well and tap water samples, historical Environmental Protection Agency (EPA) data, Superfund site data,
- Biological samples - blood, urine, hair and placenta samples, and
- Human subjects information - medical history, reproductive health records, product use data surveys, and birth outcomes.

We have developed a carefully designed relational database system to manage this project. Up until now, we have collected over 400 million data entries and manage over 2467 data entities in our database. Since urine data presently dominates the volume of data collected in our database, we focus on the urine analysis.

In this paper, we provide an overview on PROTECT and present the database workflow for big data management. Particularly, we present our open source Toolbox for *Mass Spectrometry Data Analysis*, targeted for efficient machine learning and visualization on big datasets. We also provide a detailed description on each step of the metabolomic data analysis. Given the amount of data we need to work with, we discuss how we reduce the processing time by leveraging multi-core packages.

The rest of this paper is organized as follows. Section II presents background and related work. Section III provides an overview of PROTECT database, its current status and detailed workflow. Section IV describes the Mass Spectrometry Data Analysis Toolbox (MSDA), including discussions on performance tuning and lessons learned from our analysis. Section V concludes the paper and discusses areas for future work.

II. Background

Preterm birth is a worldwide issue and its leading causes are still under investigation. P. Meis et al. identified a set of risk factors that could contribute to preterm birth [10]. These

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

factors are categorized as: i) demographic factors, ii) medical history, iii) previous obstetric history and iv) current pregnancy. J. Meeker et al. correlated phthalate exposure with preterm birth by targeting specific phthalate metabolites, including MBP, MBzp and di(2-ethyl-hexyl) [11]. The contamination of groundwater has also been studied by T. Torres et al. [12]. They suggested a group of Chlorinated Volatile Organic Compounds (CVOCs), including trichloroethylene (TCE), tetrachloroethylene (PCE) and chloroform (TCM), could have a strong influence on preterm birth. Roca et al. proposed a strategy that combines a targeted approach for pesticide metabolites with a post-targeted screening for contaminant exposure, to determine the biomarkers in urine [13]. Their approach facilitates identifying biomarkers of exposure due to environmental pollutants.

In order to support a wide range of multidisciplinary studies, many Electronic Data Capture (EDC) systems have been developed to provide an automated workflow for data collection, reporting and exploration. EDCs are mainly designed to reduce the data retrieval cycle and to avoid errors during the data collection process. The StudyTRAX system can integrate data management with the process of generating academic outcomes (e.g., manuscripts, presentations, book chapters), which dramatically increases user productivity [14]. LimeSurvey is an open source package, providing a free and secured web-based interface to leverage the capability of customizable data collection [15]. Tools, such as Electronic Laboratory Notebooks (ELN), are designed to facilitate the documentation of experiments and procedures performed in a laboratory environment [16]. Among various web-based electronic data capture systems, REDCap is one of the most user-friendly tools that can stream captured data directly into the database [17]. The Environmental Quality Information System (EQuIS) from EarthSoft can integrate data collection with data management, provide automated web-based dashboards for the distributed environment, and support real-time data capturing and reporting schemes [18]–[20].

Based on the high quality data collected with these systems, previous studies have found that metabolites in urine could provides some clues, such as the residue of environmental pollutants in the human body that can trigger different clinical symptoms. W. Arlt et al. applied Generalized Matrix Relevance Learning Vector Quantization (GMLVQ) to discriminate adrenocortical adenoma (ACA) and malignant adrenocortical carcinoma(ACC), using urine steroid metabolomics as the biomarker [21]. Y. Kim et al. proposed using multivariate methods, decision trees and random forests, to diagnose breast cancer using urine metabolome profiles [22]. An efficient protocol for radiation metabolomics using urine samples was proposed by C. Lanz et al., which applies random forest techniques to gas chromatography, combined with mass spectrometry [23]. S. Reichenbach et al. proposed a new method to extract non-targeted chromatographic features from 2D chromatograms and showed that a Support Vector Machine (SVM) outperforms a k-Nearest Neighbor (kNN) clustering in their case studies [24]. In this study, we have developed a noncommercial *Mass Spectrometry Data Analysis Toolbox* and support a variety of machine learning techniques, including Principle Component Analysis (PCA) and hierarchical clustering to facilitate large-scale metabolomic analysis.

III. Urine Sample Database

The PROTECT database is built to handle terabytes of project data for our preterm birth study in Puerto Rico. We have designed an efficient framework for data import and cleaning, enabling the generation of detailed reports on specific queries to facilitate research activities. In terms of urine analysis, we store decoded raw urine data in the database and provide users with open source tools to extend their research ideas. Next, we will describe the goals of the PROTECT project, present current status of our data repository, demonstrate the workflow using proprietary software, and discuss details of our challenges with working with urine sample data in our study.

A. The PROTECT Center

The NIEHS Puerto Rico Testsite for Exploring Contamination Threats (PROTECT) Center studies the causal effects between exposure to environmental contamination and the high preterm birth rates recorded in Puerto Rico. We collect a wide range of data, including: blood, urine, ground water, tap water, placenta and medical records. Based on this rich range of data, we attempt to identify contributing factors associated with preterm birth. Domain specific analyses are applied, which include non-targeted chemical analysis, mechanistic toxicology, and targeted epidemiology. The organization of PROTECT is shown in Figure 1. An additional goal is to develop green remediation strategies to alleviate exposure and to reduce future preterm birth rates. For the PROTECT database, we support multiple research communities by facilitating data cleaning, data storage, data security and data reporting. We utilize software developed and marketed by EarthSoft called EQuIS. We are presently using EQuIS Professional and EQuIS Enterprise. Our backend database is Microsoft's SQLserver. We have developed a number of tools for data management and modeling to advance our preterm birth study.

B. Data Storage

In the current database we capture human subject data, environmental data and biological data. We currently have more than 400 million data points in our system. The structure of these data points is provided in Table I. In the near future, we expect to host more than 100 billion data entries in our system.

TABLE I. PROTECT database repository.

	Data Points (In millions)
Environmental	1.3
Human Subjects	1.5
Biological	0.2
Non-targeted	400

Since each data entry can be an indicator tied to an adverse reproductive outcome, we need the ability to carefully evaluate relationships between data entries across the millions of data points. Due to the sheer data volume, we leverage specialized software to facilitate the data management process. We also have the challenge that we are working is a geographically distributed team of researchers in PROTECT. Our researchers that need access to PROTECT data will have web-based dashboards to help them manage their data and perform customized

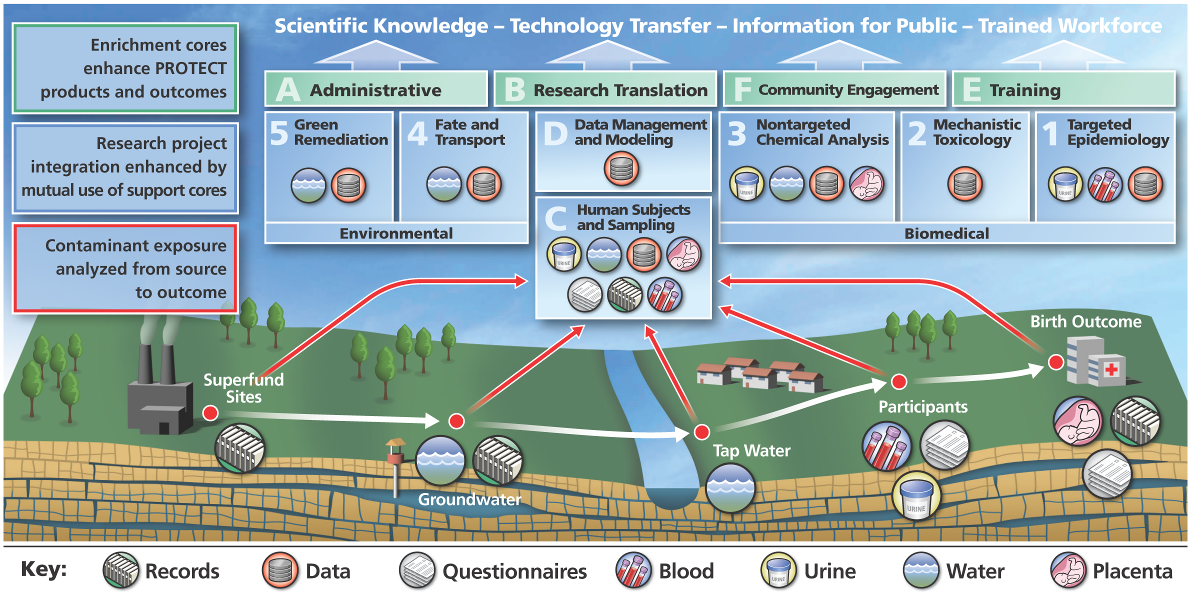

Scientific Knowledge – Technology Transfer – Information for Public – Trained Workforce
Enrichment cores enhance PROTECT products and outcomes
Research project integration enhanced by mutual use of support cores
Contaminant exposure analyzed from source to outcome
A Administrative
B Research Translation
F Community Engagement
E Training
5 Green Remediation
4 Fate and Transport
D Data Management and Modeling
3 Nontargeted Chemical Analysis
2 Mechanistic Toxicology
1 Targeted Epidemiology
Environmental
Biomedical
C Human Subjects and Sampling
Superfund Sites
Groundwater
Tap Water
Participants
Birth Outcome
URINE
Key:
Records
Data
Questionnaires
Blood
Urine
Water
Placenta

Figure 1. PROTECT collects source samples (white arrows, bottom) to analyze factors that contribute to preterm birth. Core C collects environmental and biological information. Core D handles data management and supports data analytics. Projects 1-5 utilize the collected data for their scientific studies.

queries. Our system helps to identify the linkages between pollutants and birth outcomes through the use of advanced machine learning algorithms.

C. Data Cleaning

After capturing the data, the integrity of each entity is inspected. This process is called *data cleaning*, and is fairly standard in any data collection campaign. The goal is to reduce errors in the data. The checking process needs to verify that each data field conforms to the data type and is within range for each field. Data dependencies between fields are also checked. This process is performed before incorporating any data into the database. To facilitate this checking, we have a complete data dictionary available for every entity stored in the database.

The data dictionary is developed by each domain expert. We abide by the rules present in the data dictionary when developing our schema, which in turn, helps to maintain a high level of data quality. Our cleaning tools can quickly highlight any detected anomalies in the data. Our comprehensive cleaning procedure can pinpoint corrupted data, and help to prevent errors from entering the database.

D. Software Stack

For the front-end of the PROTECT database, we use Microsoft Visual Basic to configure the schema for data cleaning. These scripts are used by EarthSoft's EQuIS Electronic Data Processor (EDP) to clean the input data according to the defined constraints. After data screening, EQuIS sends the cleaned data to Microsoft SQLserver. We leverage EQuIS Professional [25] and Enterprise [19] to support both standalone and distributed development environments, respectively. We use EQUIS's Electronic Data Processor (EDP) to import data into the database.

Users can customize data formats, also known as Electronic Data Deliverables (EDDs), for their individual study. EDDs can be stored in a number of popular documentation formats including Excel spreadsheets and Comma Separated values (CSVs). Typically, four files are needed to handle data cleaning: 1) format definition file, 2) a custom handler file, 3) an enumeration file, and 4) a reference value file. The *format definition file* follows the rules defined in the data dictionary. The *custom handler* applies the data checking scheme and generates discrepancy reports if mismatches are detected. Whenever a set of values need to be indexed based on their definition in the data dictionary, the *enumeration file* is used for this purpose, and so is an optional file. Users would use the reference value file to allow them check reference values remotely [18]. Errors are highlighted with detailed warnings to facilitate the debugging process. Only after all errors are resolved, then the input data values can be committed to the database.

This automated data cleaning process is handled through the EDP module in both EQuIS Professional and Enterprise. Distinct from the standalone development of EQuIS Professional, EQuIS Enterprise provides web-based dashboards to support distributed users [19]. Each dashboard is customized to include a set of widgets specific to the needs to the data researcher. For instance, data uploads and checking can be performed using the *EDD Upload* widget. The cleaning status can be configured to automatically inform a group of users through the *Notices* widget. The Environmental Information Agents (*EIA*) widget pushes reports to users on scheduled events or dates. Online data access is shared across the PROTECT center, providing access to researchers in Puerto Rico, Massachusetts, Michigan and West Virginia. The workflow from data collection to data reporting is shown in Figure 2.

E. Urine Samples

In our previous work, we have reported on a study of non-targeted analysis on 6 urine samples from Puerto Rico [1].

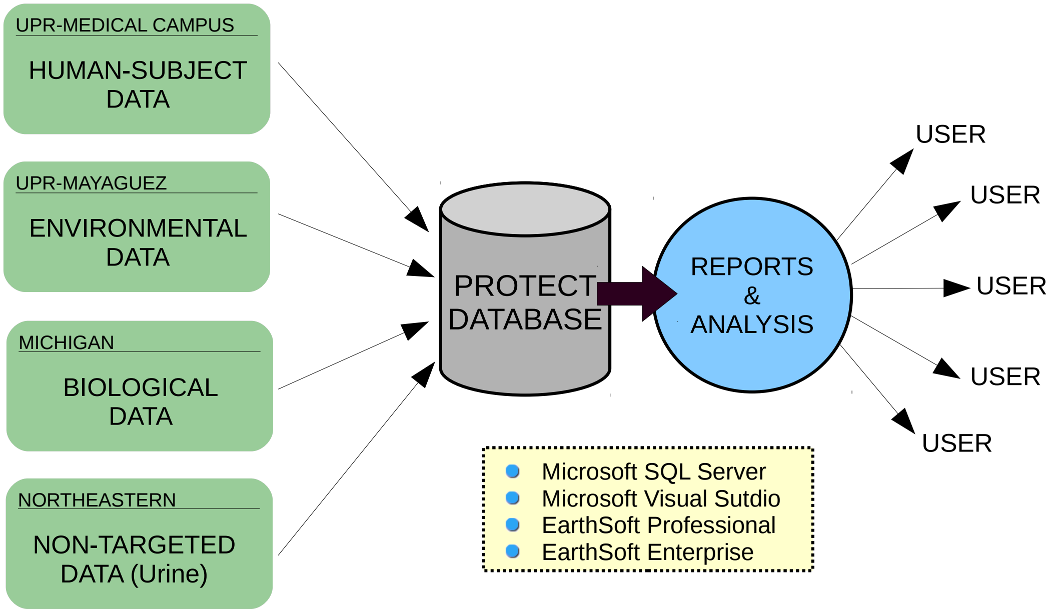


Figure 2. The PROTECT database collects data from different sources and includes data on: human subjects, environmental parameters, biological analysis and non-targeted chemical/biological data. Data is cleaned before it is imported into the database. The system supports both standalone analysis and distributed reporting capabilities through EQuIS Professional and Enterprise. Equipped with a distributed solution, users can query customized reports through the web server. The PROTECT database also supports data mining and modeling capabilities.

TABLE II. Urine samples from Developmental Neurotoxicity Assessment of Mixtures in Children (DENAMIC) project.

Country (Region)	Type	Resolution (FWHM)	Raw Data Samples		
			ESI+	ESI-	ESI+/-
Spain (Valencia)	Pregnant mothers	50,000	306	306	-
		25,000	-	-	143
	Children (4 years)	50,000	216	216	-
Spain (Sabadell)	Pregnant mothers	25,000	-	-	160
Slovakia	Pregnant mothers	25,000	-	-	52
	Children (4 years)	25,000	-	-	49

We discovered a range of clustering patterns present in these samples. Due to the limited sample data, the contributing chemicals remain to be identified.

To be prepared to compare results from our cohort to other cohorts of expectant mothers throughout the world, we have acquired an existing set of urine samples from the *Developmental Neurotoxicity Assessment of Mixtures in Children* (DENAMIC) Project [26] being carried out in Spain. The details for the DENAMIC urine samples are presented in Table II.

There are 661 mothers and 265 children across three different regions in the Spain study. The mass spectrum data is acquired using full scan mode (50-800 m/z), with a resolving power of 50,000 FWHM (full width at half maximum) (scan speed, 2 Hz), in both + and - modes of ESI (electrospray ionization), and with and without HCD (higher-energy c-trap dissociation fragmentation). Mass spectrum analysis is performed on the Orbitrap ExactiveTM mass spectrometer (Thermo Scientific, Bremen, Germany). Data acquisition is accomplished using Thermo Scientific's Trace Finder 3.1 software. The raw mzXML files for these samples are rather large (136 GB). Next, we will discuss how we work with this data in order to identify patterns in this data set. In the following section, we introduce our machine learning framework to deal with the data processing challenges with this large data set.

IV. MSDA

Our urine analysis procedure relies heavily on the accuracy of the mass spectrometer, and the supporting software. We could choose to use applications such as MarkerView [27], SIMCA-P+ [28] and SAS [29]. These packages provide black-box style analysis with limited flexibility and processing power. For example, MarkerView cannot handle large datasets efficiently; it takes more than 20 minutes to perform PCA on 6 urine samples, and is unable to process larger datasets due to memory storage issues. Meanwhile, there is free quantitative metabolomics software available, such as MetaboAnalyst [30] and MeltDB [31], which provides a comprehensive suite of analysis recipes. However, these packages, while free, come with a number of challenges, including limiting the maximum file (limited to 6 MB in MetaboAnalyst), and very poor

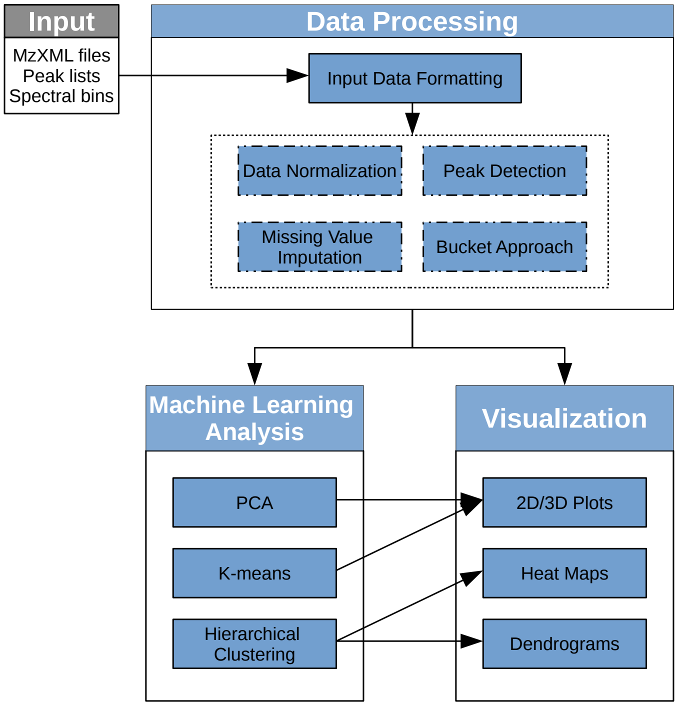


Figure 3. The Mass Spectrometry Data Analysis Toolbox.

runtime performance. In order to address these issues, we have developed our own open-source *Mass Spectrometry Data Analysis Toolbox* (MSDA), designed to efficiently carry out a number of different mass spectrometry (MS) data analyses tasks. Our MSDA Toolbox is able to analyze large datasets.

As shown in Figure 3, the Toolbox consists of three main components: 1) data processing, 2) machine learning analysis, and 3) visualization. The data processing component translates the input data to the required input format for the data analysis component. MSDA supports a wide variety of input data types, including raw mzXML files, peak lists and spectral bins. A set of data processing methods, such as peak detection/alignment, bucketing, missing value imputation and data normalization are supported. The machine learning analysis component provides a wide range of machine learning techniques, such as feature selection, clustering and PCA to provide insight into the data. The visualization component generates 2D/3D plots for PCA results, as well as heat maps and dendrograms for viewing hierarchical clustering results. The Toolbox is an open source software written in Python, which utilizes state of the art statistical and machine learning libraries written in Python, R and C. By integrating data analytics capabilities, the Toolbox can significantly reduce data processing time, providing researchers with a fast research toolset. We have modularized our design to facilitate future contributions from the open source community.

A. Data Processing

The data processing component filters the input data formats before passing the data on to other components for the further analysis. The first step is to check the input data formatting, which transforms input data files into a data matrix. Next, the user can choose to normalize the data, insert missing values, or perform peak detection and bucketing. In this paper we utilize the urine sample data from the DENAMIC project to demonstrate how our toolbox facilitates data discovery.

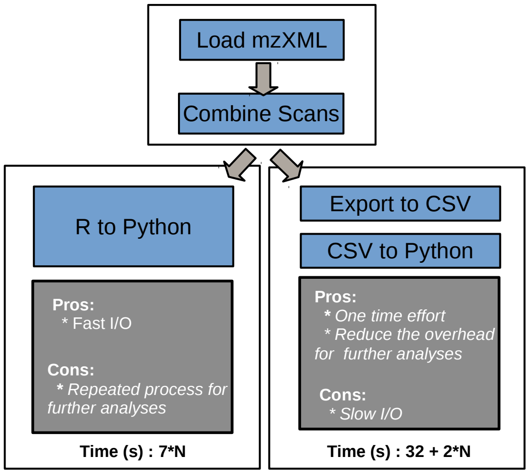


Figure 4. Compare memory exporting (bottom left) and disk (bottom right) exporting approach for formatting one mzXML file. N stands for the number of rounds of analysis. Reading data directly from R consumes 7 seconds / round, whereas reading from disk consumes 2 seconds / round, plus one time overhead of 32 seconds.

1) Input data formatting: This step converts input data into a data matrix, with samples in rows and features in columns. Three different data sources are supported: MS spectra raw files, peak list files and spectra bin files. The MS raw file should be in the *mzXML* format, while the rest are stored as CSVs. A peak list file should have either 2 columns (mass and intensities) or 3 columns (retention time, mass and intensities), with the first row reserved for column labels. A spectra bin file can have any number of columns, with the first row filled with labels for m/z buckets. The Toolbox also supports batch processing.

We utilize the *MALDIquant* [32] package in R and the *Panda* [33] library available in Python to transform mzXML and CSV files into data matrices, respectively. To seamlessly read the output of the *MALDIquant* package using Python, *rpy2* [34] is used to convert R objects into an accepted Python format, which avoids performing slow disk I/O operations. Since the mzXML files are frequently used, we convert them to CSV files and they are saved to disk for future analysis.

This data formatting is I/O bound, and so its performance depends on the size of the input file and output locations. To translate one urine sample file from the DENAMIC project, loading the mzXML file (83 MB) using *MALDIquant* takes 6 seconds and combining all of scans into a data matrix using *data.table* consumes 1 second. In MSDA, we provide two locations for data exporting, namely memory and disk. To export the data to memory, the *rpy2* package is used to facilitate the process, reducing runtime overhead to approximately 20 μs when offloading data (350 MB) in R to Python. On the other hand, we can export the data to disk first, which takes 25 seconds in R, then read it in as a Python object, which takes 2 seconds using *Pandas*. A comparison of both approaches is presented in Figure 4. Given the common case that we need to run many rounds (N being a big number) of analyses, the disk approach would be preferred by most users.

2) Data normalization and missing value imputation: Once a data matrix is generated, users can select whether to perform data normalization and missing value *imputation*. Many statistical and machine learning algorithms, such as PCA, do not work properly if features have a wide range of values or missing entries. Data normalization is used to modify the range of independent features so that they are normally distributed. Several data normalization methods are provided in MSDA, such as centering by mean or median values, scaling by the standard deviation, maximization, root square or logarithm. A variety of methods to treat missing values are also implemented in the Toolbox, including replacement by zero, mean, median and discarding the whole feature in the sample if the number of missing values is over a user-defined threshold. The *numpy* [35] library is used for data normalization and missing value imputation in our system. The resulting execution time is less than 50 ms for each urine sample.

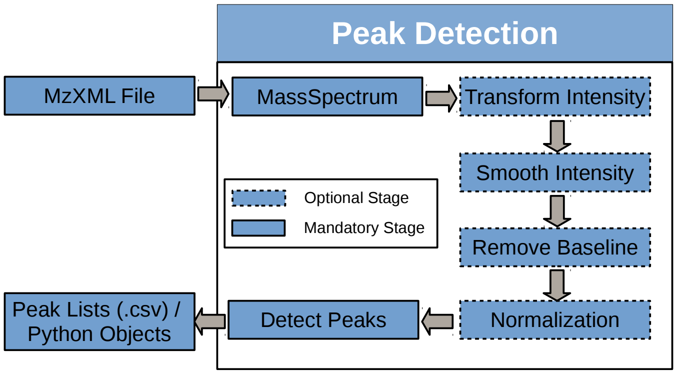
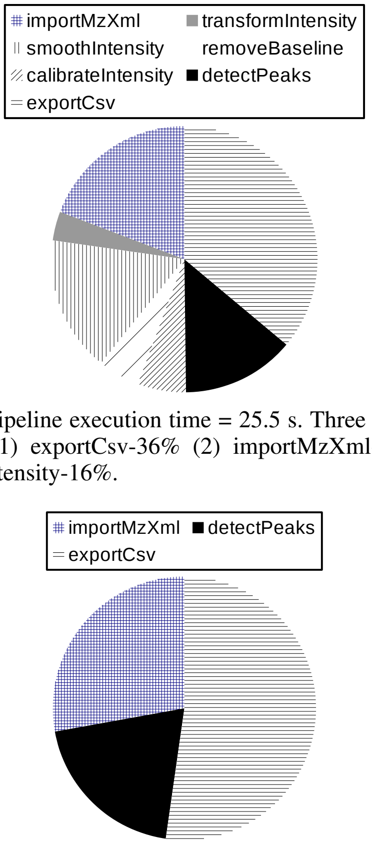


Figure 5. Peak detection pipeline. Dashed boxes represent optional steps and the solid ones stand for mandatory steps. A full peak detection pipeline includes all the stages, whereas a short one only includes the mandatory stages.

3) Peak Detection: Users can choose whether to select peaks for a given mzXML file or just leave the MS data unchanged in CSV format. *MALDIquant* is used to carry out the peak detection task. It provides a peak detection pipeline, as shown in Figure 5. The 4 stages of the pipeline include: i.) removing noise from the spectra, ii.) transforming intensities, iii.) correcting the baseline, and iv.) aligning the spectra.

At the beginning of the peak detection process, we read the mzXML data by calling the *importMzXml* function. Figure 6 (a) shows one scan of an input urine sample. The intensity values are then centered and scaled, as shown in Figure 6 (b). Several scaling methods are supported, including square root and logarithm. The smoothing stage can be implemented using either the *SavitzkyGolay* or *MovingAverage* method, and configurable to a half window size [36]. The *SavitzkyGolay* method with a half window size of 10 is applied in Figure 6 (c). Then, the spectra baseline can be estimated and removed by adopting *SNIP* [37], *TopHat* [38], *ConvexHull* [39] or *median* methods. These methods estimate the background signal, iteratively. Users can adjust the *iteration* parameter to achieve the best result. Figure 6 (d) shows the baseline estimation using the *SNIP* method with 10 iterations. Figure 6 (e) shows the spectra with the baseline removed. The resulting spectra can be normalized using either *Total-Ion-Current-Calibration* (TIC) or *Probabilistic Quotient Normalization* (PQN) [40]. Figure 6 (f) shows the normalized spectra using the TIC method. The last and most critical stage is the peak detection step. This step estimates noise using either the *media-absolute-deviation* or *Friedmans Super Smoother* method. Users can adjust the half window size and signal-to-noise ratio (SNR) to identify the local maximum intensities. The SNR can also be estimated automatically using the *estimateNoise* function. The baselines for SNR 1 and 2 are presented in Figure 6 (g). The results of using the full peak detection pipeline are plotted in Figure 6 (h), where an SNR 2 is applied. In addition, the short pipeline's results are shown in Figure 6 (i). This approach skips all the optional stages that were present in Figure 5. The output of the pipeline can be either directly fed to Python through *rpy2*, or saved to disk as a peak list file in CSV format for faster accesses in the future.

We show that different peak lists are generated by applying either the full or short peak detection pipeline in Figures 6 (h) and (i). Users can choose which pipeline to use, depending on the trade-off between the execution time and the peak resolution.

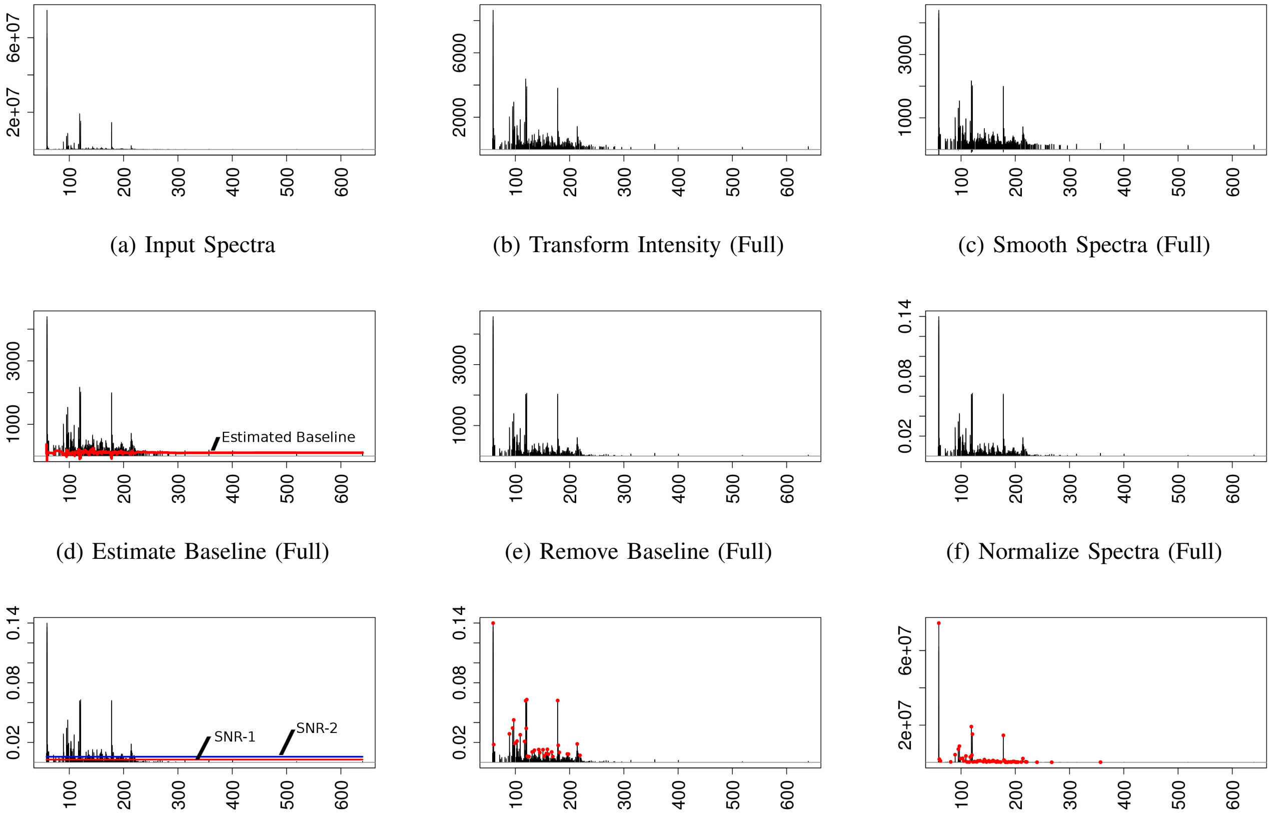


(a) Full pipeline execution time = 25.5 s. Three dominant factors: (1) exportCsv-36% (2) importMzXml-19% (3) smoothIntensity-16%.

(b) Short pipeline execution time = 17.6 s. Three dominant factors: (1) exportCsv-52% (2) importMzXml-28% (3) detectPeaks-20%.

Figure 7. Performance breakdown for both the full and short pipelines. The 3 most timing consuming steps are presented in descending order.

4) Peak Detection Performance: We use an Intel i7-4790K (4 cores / 8 threads, using hyperthreading) to evaluate peak de-



(a) Input Spectra

(b) Transform Intensity (Full)

(c) Smooth Spectra (Full)

(d) Estimate Baseline (Full)

(e) Remove Baseline (Full)

(f) Normalize Spectra (Full)

(g) Estimate SNR (Full)

(h) Detect Peaks (Full)

(i) Detect Peaks (Short)

Figure 6. A pipelined Peak Detection example. The spectral results after each stage in a fully pipelined peak detection are shown in Figure (b) to (h), whereas Figure (i) shows the detected peaks using a short pipeline. The x-axis is mass, the y-axis is intensity.

tection performance. In R, the average processing time for one urine sample is 17 and 25 seconds for the short and full peak detection pipelines, respectively. The detailed performance breakdown for each case is illustrated in Figure 7. Since there are 1448 urine samples in the *DENAMIC* datasets, it requires 7 hours for the short pipeline and 10 hours for the full pipeline to detect intensity peaks. To reduce the processing overhead, we identify the performance bottlenecks and explore multi-threading techniques to accelerate the process. We consider the aforementioned single-threaded performance as our baseline for comparison.

As shown in Figure 7, the dominant performance bottleneck lies in the *exportCsv* step, which merges a list of mass spectrometry data into a single matrix, and then exports the data to a CSV file. Due to the overhead of combining rows (*rbind*) and columns (*cbind*) in R, the merging step takes 9.1 seconds, as compared to 0.1 seconds to export the CSV file. In order to accelerate the merge step, we leverage the optimized *rbindlist* function in the *data.table* package [41]. First, for each peak scan, as a data frame object is appended to the pre-allocated list, the list is reduced into one data frame using *rbindlist*. We observed the execution overhead of the merge step drops from 9.1 seconds to just over 1 second. This is because *rbindlist* is highly optimized in C, whereas *rbind* is coded in a high level scripting language (R). We can reduce the execution time of *exportCsv* from the 9.2 seconds (baseline) to 1.1 seconds.

The next performance hot-spot is the *importMzXml* process. We leverage the *mzR* package from Bioconductor (an open software for bioinformatics) [42][43]. In our baseline approach, *importMzXML* in *MALDIquantForeign* reads the mzXML file (internally using *readMzXmlFile* from the *readMzXmlData* package) and creates the MassSpectrum class accordingly [44][45]. To improve performance, we utilize the *openMSfile* function from the *mzR* package to read the input file more efficiently, and apply the *peaks* method inside *mzR* to acquire the m/z and intensity values. We achieve a 2.6x speedup over the baseline, reducing the elapsed time from 4.9 seconds to 1.9 seconds.

Besides single-threaded optimization, we also utilize *parallel* packages (e.g., a multi-threaded implementation in R) to obtain more speedup [46]. The execution time for the three steps (*import, peak, export*) by varying the number of threads is shown in Figure 8. Here, the *peak* stage includes the steps from *transformIntensity* to *detectPeaks* in Figure 5.

In our optimization scheme, we use the *mclapply* function to parallelize the list operation on the *MassSpectrum* data. The *peak* operation is applied to every *MassSpectrum* data list using *mclapply*. The same operation is applied to the parallel creation of *MassSpectrum* objects in the *import* stage and the *export*

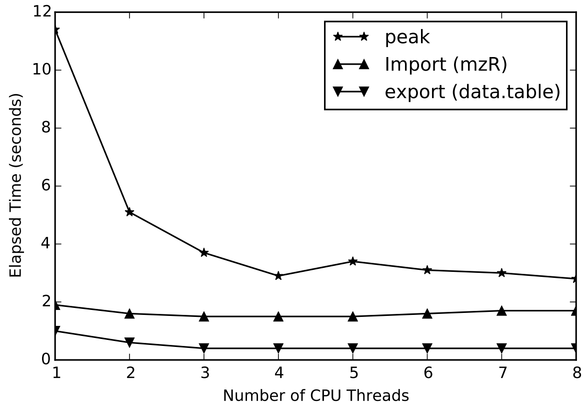


Figure 8. The performance for parallelizing three performance hotspots 1) import 2) peak 3) export for the full pipeline peak detection on an Intel i7-4790K.

stage. However, since the *parallel* package forks to create a new process by taking a complete copy of the master process, the overhead is very high for both *import* and *export* stages. Thus, the performance flattens out after two threads for *import* (red line) and *export* (yellow line), as shown in Figure 8. For the *peak* stage, only urine sample IDs are duplicated during the forking process. We achieved a 3.9x speedup using 4 threads. In summary, Figure 9 shows that by using *data.table* (rbindlist), we can achieve a 1.5x speedup on average. Adding *mzR* (openMSfile) to *data.table*, we can obtain on average 1.9x speedup. When using *parallel* (mclapply), and adding the benefits of the two previous optimization methods, we can achieve a 5.3x speedup by using 4 threads. Overall, we reduced the processing time for the full pipeline peak detection from 10 hours to 2 hours. Applying the same technique, we shortened the processing time from 7 hours to 1 hour for the short pipeline peak detection.

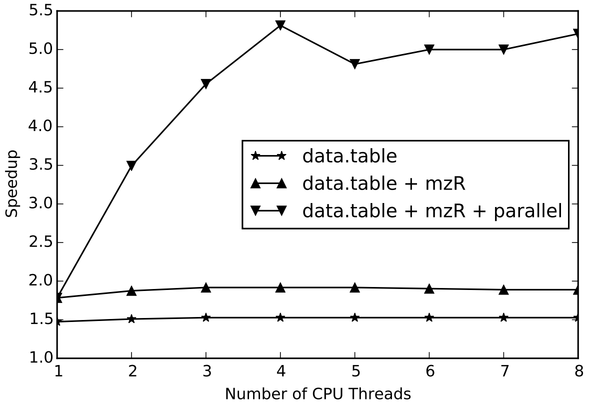
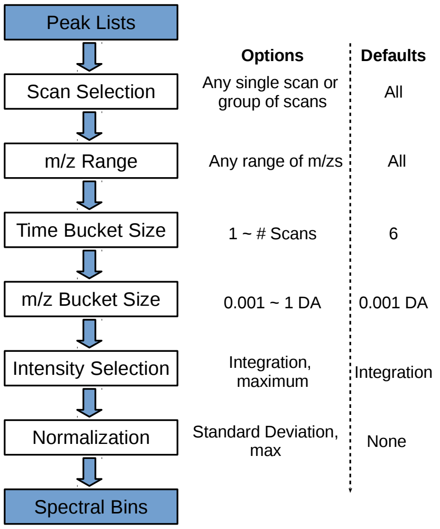


Figure 9. The total speedup achieved by parallelizing the full pipeline peak detection on Intel i7-4790K.

5) Bucketing Approach: The Bucketing approach is also implemented in MSDA. For each peak list, we group multiple scans together in a time-bucket, and multiple m/zs in a m/z-bucket. The time-bucket size and the m/z-bucket size are chosen based on the HPLC peak width, while considering the mass accuracy and the resolution of the mass spectrometer. The Bucketing approach essentially extracts the spectral features integrated over time in order to reduce the redundancy in the original MS data, and to improve our computational capabilities. As shown in Figure 10, the input is a set of peak lists in CSV format and the output is the spectral bins. The boxes between the input and the output represent the optional transformations that can be applied by the users. Users can choose from the listed options in Figure 10, or specify their own parameters. Otherwise, the default values are used. The first parameter allows the user to choose any combination of scans in a sample. A range of m/z sizes can be specified using the second parameter. Users can choose how many scans they want to put into one bucket by specifying the time-bucket size, and the size for m/z-buckets by the m/z bucket size. Users can also choose how to combine the bucketed scans, by either integrating all the intensity values, or by selecting the maximum intensity for each m/z bucket. In addition, users can also select the normalization method for the bucketed intensities.



	Options	Defaults
Scan Selection	Any single scan or group of scans	All
m/z Range	Any range of m/zs	All
Time Bucket Size	1 ~ # Scans	6
m/z Bucket Size	0.001 ~ 1 DA	0.001 DA
Intensity Selection	Integration, maximum	Integration
Normalization	Standard Deviation, max	None

Figure 10. Workflow of the Bucketing approach. Users can specify the parameters.

To process one urine sample from the DENAMIC project, whose scan speed is 2 Hz with normal MS scans and HCD fragmentation scans interleaved, we choose the even-numbered scans in the range of 40 to 1200 and use 6 as the time-bucket size and 0.001 DA as the m/z-bucket size.

The generated spectral bins for the 306 urine samples (spanish mothers with ESI+) consist of 97 time-buckets and 750k m/z-buckets in each sample, corresponding to a 29,682 x 750,204 sparse matrix of 1.5% density. The *Scipy* [47] library is used to store the sparse matrix in the compressed column storage (csc) format to minimize the memory cost. The spectral bins, represented as in a data matrix, can be directly fed into the machine learning analysis component or stored on disk as a CSV file.

Figure 11 plots the execution time of the bucketing approach for one urine sample consisting 600 scans and 942,397

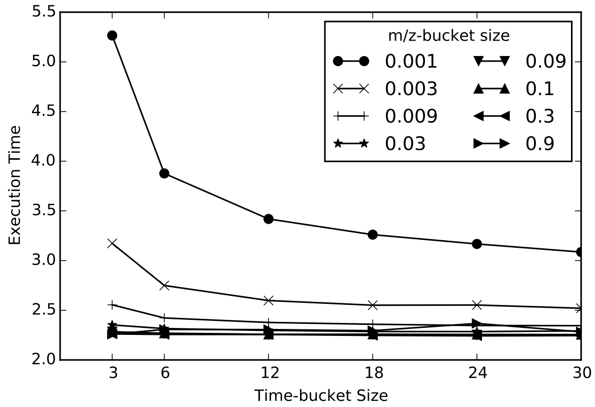


Figure 11. Bucketing performance while varying the bucket size.

(m/z, intensity) pairs. We vary the time and m/z-bucket sizes to compare their impact on the performance. It turns out that m/z-bucket size is the dominant factor as we can see two big jumps in the execution time when decreasing the m/z-bucket size from 0.009 to 0.003 and from 0.003 to 0.001. The time-bucket size, on the other hand, contributes little to the execution time, especially when the m/z-bucket size is large.

B. Machine Learning Analysis

A general set of machine learning techniques for mass spectrum analysis are implemented in MSDA. In this section, three different methods are discussed: i) Principal Component Analysis (PCA), ii) K-means clustering, and iii) Hierarchical clustering.

1) PCA: Principal Component Analysis (PCA) [48] is a commonly-used method to reduce a data matrix of n features to k ($k << n$) features, with much of the variability in the data preserved. The transformed k features are called *principal components*. The principal components are sorted by their variance, hence the first principal component has the largest variance and each subsequent component has the next largest variance. As noted by Worley and Powers, PCA is one of the most popular multivariate analysis methods used in metabolomics [49]. Because PCA can significantly reduce the dimensionality of a dataset, it is often used in compression algorithms as it provides an approximation of the original data using only k principal components. Another common use of PCA is visualization: datasets in high-dimensional space can be projected onto a 2D or 3D space, while most of the patterns in the dataset are preserved. Researchers can gain insight into complex data by just studying the 2D and 3D plots. In MS data analysis, the data matrix is usually a huge sparse matrix, especially when the m/z-bucket size is small. One of the data matrices from the DENAMIC project used in this work contains 306 urine samples, where each sample is represented by 97 time-buckets and 750,204 m/z-buckets. This results in a 29,682 x 750,204 sparse data matrix with ~ 350 million non-zero elements. Normal PCA methods that take a dense matrix as the input cannot be used in this case, hence MSDA uses a TruncatedSVD [50] from *scikit-learn* [51] for this task. The execution time of applying the TruncatedPCA on the aforementioned matrix is 34 seconds on average.

2) K-means Clustering: K-means [52] is one of the most popular clustering algorithms, especially given its simplicity and effectiveness. It is widely used in metabolomic studies due to its capability to perform rapid subset identification from the information-rich spectral datasets [53]–[55]. In MS analysis, K-means (and its variants) are heavily used to cluster urine samples to detect anomalies. It can either be applied on the original data matrix to calculate the pairwise distances in n dimensions, or directly on the dimensionality-reduced matrix generated by PCA.

3) Hierarchical Clustering: In K-means, the desired number of clusterers k must be specified by the user, and determining k is not an easy job. Hierarchical clustering is another widely used clustering algorithm that builds a hierarchy of clusters, so that the clustering results for all k (from 1 to n) are automatically generated [56][57]. MSDA uses an agglomerative clustering algorithm and displays the clustering results using a dendrogram.

C. Visualization

In this section, we showcase 2D/3D plots for the PCA results and the heat map / dendrogram for the hierarchical clustering results.

1) 2D/3D plots: Figures 12 and 13 show the 2D and 3D plots of the PCA results generated by MSDA using *matplotlib* [58] in Python. We are able to project 29,682 data points on a 2D and 3D space in 20 seconds and 25 seconds, respectively.

2) Heat map and Dendrogram: A heat map is a data visualization technique to reveal the relationships among data points, using a color scheme. A heat map applies a color-shared matrix display, and reorders the data matrix to disclose the underlying structure of the data [59]. It is widely applied in data visualization in the natural and biological sciences. This technique has been extensively used in previous urine studies [60]–[62].

A dendrogram is a tree-based diagram that illustrates how k clusters are grown out of n observations for any arbitrary k (from 1 to n). To view the clustering result for a specific k, a ”cut” can be taken horizontally on the y-axis where k intersections are created. Each vertical line at the intersection then leads to a cluster.

Heat maps and dendrograms are often combined to illustrate hierarchical clustering results. MSDA uses *matplotlib* to generate the combined plot. An example of 306 urine samples is shown in Figure 14.

V. Conclusion and Future Work

In this paper, we have presented an overview of *Puerto Rico Testsite for Exploring Contamination Threats* Center, and highlighted many of the challenges faced during data management and analysis. We have developed a highly efficient solution based on the EQuIS, providing for efficient data cleaning and reporting given the diversity of the data sources.

In order to begin to understand how environmental factors can impact preterm birth, we have developed a number of Toolboxes. We discuss the development of a Toolbox to streamline metabolomic analysis of expectant mother’s urine samples. The goal is to identify non-targeted compounds in the urine. Due to the computational challenges of this analysis, we have built an open source framework called the *Mass Spectrometry Data*

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

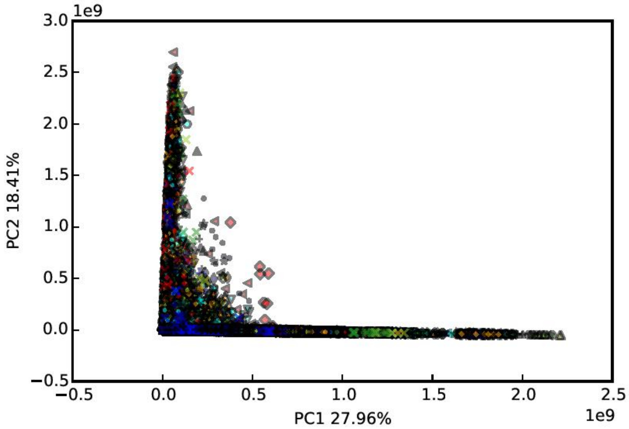


Figure 12. A 2-D PCA plot.

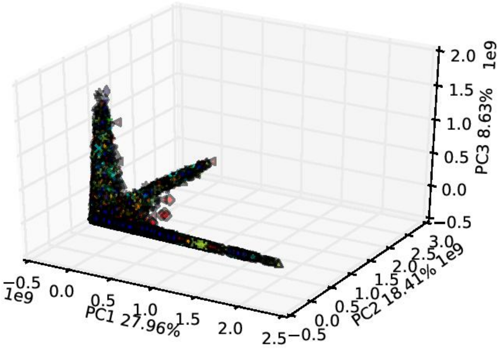


Figure 13. A 3-D PCA plot.

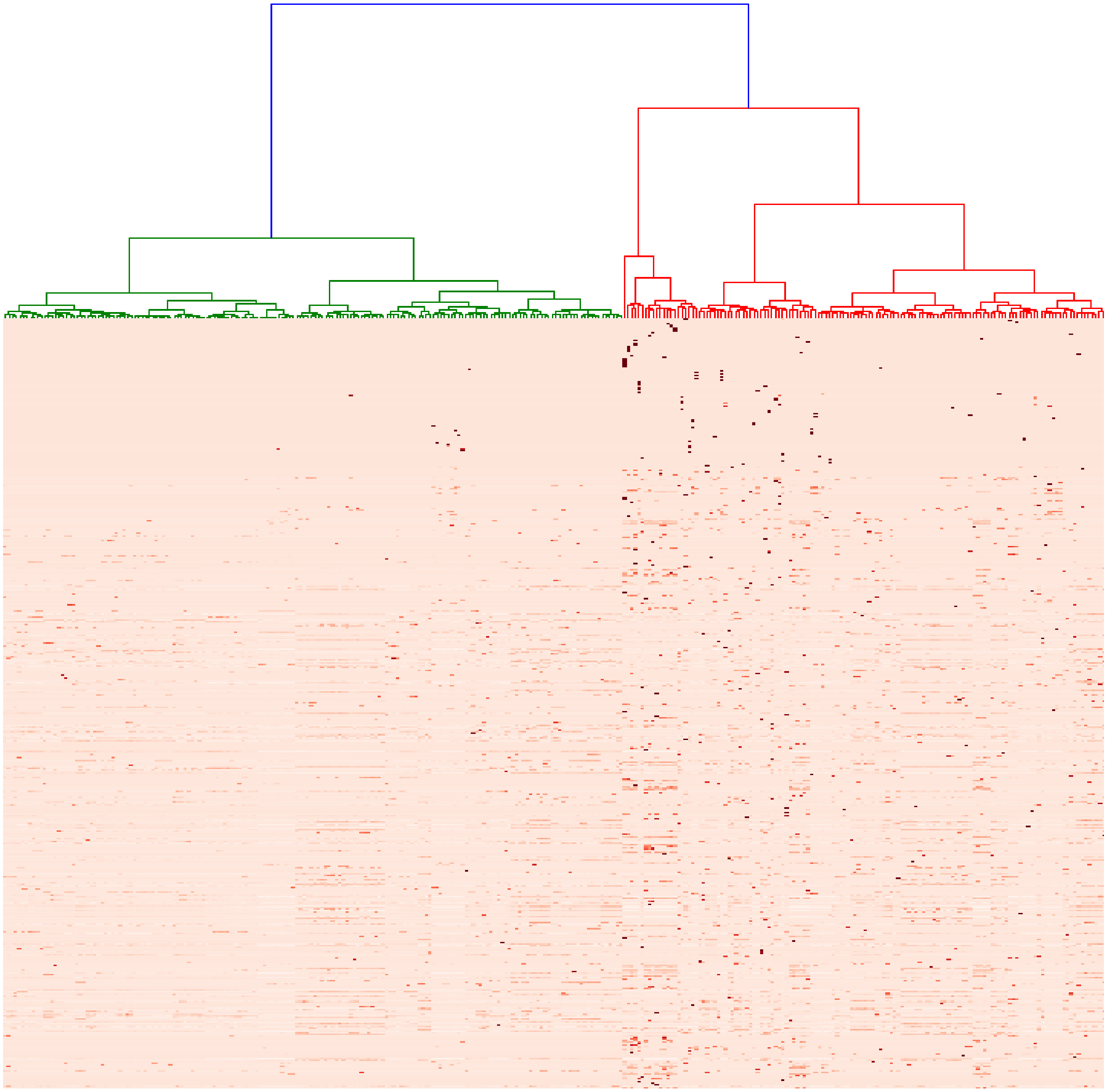


Figure 14. Example of a dendrogram and heatmap available in MSDA.

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Analysis Toolbox. The Toolbox can signficantly accelerate metabolomic analysis. MSDA can handle a complete analysis pipeline ranging from data processing to machine learning. The Toolbox also provides visualization capabilities to help the user understand sample characteristics present in a high-dimensional feature space.

We have been able to demonstrate the saving provided by MSDA, enabling much faster processing utilizing parallelization, but also integrating a number of tools together into a single framework. We believe MSDA will have a strong impact on discovering biological patterns in the future.

To further improve the capability of MSDA, we plan to implement additional machine learning and statistical techniques, including PLS-DA, t-Tests and SVM capabilities. We also plan to enhance the performance and scalability of the Toolbox further, leveraging GPUs and the Spark [63] distributed computation framework.

ACKNOWLEDGMENT

This work is supported in part by Award Number P42ES017198 from the National Institute of Environmental Health Sciences. Authors would like to thank Ana Miralles-Marco for her collaboration. This study had the support of the FP7-ENV-2011 DENAMIC project (cod 282957). In addition, we would like to thank EarthSoft for their generous support.

REFERENCES

[1] X. Li, L. Yu, Y. Yao, P. Wang, R. Giese, A. Alshawabkeh, and D. Kaeli, "Big Data Analysis on Puerto Rico Testsite for Exploring Contamination Threats," in *ALLDATA'2015: The First International Conference on Big Data, Small Data, Linked Data and Open Data*, pp. 29–34, 2015.

[2] H. Blencowe *et al.*, "National regional and worldwide estimates of preterm birth rates in the year 2010 with time trends since 1990 for selected countries: a systematic analysis and implications," *The Lancet*, vol. 379, pp. 2162–2171, 2012.

[3] J. D. Meeker *et al.*, "Urinary phthalate metabolites in relation to preterm birth in mexico city," *Environ. Health Perspect.*, vol. 117, pp. 1587–1592, 2009.

[4] D. Cantonwine *et al.*, "Bisphenol a exposure in Mexico City and Risk of prematurity: a pilot nested case control study," *Environ. Health*, vol. 9, pp. 62–68, 2010.

[5] A. P. Mucha *et al.*, "Abstract: Association between pbde exposure and preterm birth," in *10 Annual Workshop on Brominated Flame Retardants*, (Victoria, BC Canada), p. 42, 2008.

[6] K. Tsukimori *et al.*, "Long-term effects of polychlorinated biphenyls and dioxins on pregnancy outcomes in women affected by the yusho incident," *Eniron. Health. Perspect.*, vol. 116, pp. 626–630, 2008.

[7] P. Z. Ruckart, F. J. Bove, and M. Maslia, "Evaluation of contaminated drinking water and preterm birth, small for gestational age, and birth weight at Marine Corps Base Camp Lejeune, North Carolina: a cross-sectional study," *Environ. Health*, vol. 13, pp. 1–10, 2014.

[8] J. D. Meeker, "Exposure to environmental endocrine disruptors and child development," *Arch. Pediatr. Adolesc. Med.*, vol. 166, pp. 952–958, 2012.

[9] G. Guennebaud, B. Jacob, *et al.*, "Eigen v3." http://eigen.tuxfamily.org, 2010.

[10] P. J. Meis, R. L. Goldenberg, B. M. Mercer, J. D. Iams, A. H. Moawad, M. Miodovnik, M. K. Menard, S. N. Caritis, G. R. Thurnau, S. F. Bottoms, *et al.*, "The preterm prediction study: risk factors for indicated preterm births," *American journal of obstetrics and gynecology*, vol. 178, no. 3, pp. 562–567, 1998.

[11] J. D. Meeker, H. Hu, D. E. Cantonwine, H. Lamadrid-Figueroa, A. M. Calafat, R. Loch-Caruso, M. M. Téllez-Rojo, A. S. Ettinger, and M. Hernandez-Avila, "Urinary phthalate metabolites in relation to preterm birth in mexico city," 2009.

[12] N. Torres Torres, J. Howard, I. Padilla, P. Torres, I. Cotto, and C. Irizarry, "Effects of hydrogeologic conditions on groundwater contamination of cvocs in the north coast karst aquifer of puerto rico," in *AGU Fall Meeting Abstracts*, vol. 1, p. 1251, 2012.

[13] M. Roca, N. Leon, A. Pastor, and V. Yusà, "Comprehensive analytical strategy for biomonitoring of pesticides in urine by liquid chromatography–orbitrap high resolution mass spectrometry," *Journal of Chromatography A*, vol. 1374, pp. 66–76, 2014.

[14] Y. Chen, P. J. McGrath, and J. W. Stewart, "Web-Based Electronic Data Capture System in Psychiatry Clinical Trials: A StudyTRAX Review,"

[15] C. Schmitz *et al.*, "Limesurvey: an open source survey tool," *LimeSurvey Project Hamburg, Germany. URL http://www. limesurvey. org*, 2012.

[16] C. Voegele, B. Bouchereau, N. Robinot, J. McKay, P. Damiecki, and L. Alteyrac, "A universal open-source Electronic Laboratory Notebook," *Bioinformatics*, vol. 29, no. 13, pp. 1710–1712, 2013.

[17] P. A. Harris, R. Taylor, R. Thielke, J. Payne, N. Gonzalez, and J. G. Conde, "Research electronic data capture (REDCap)a metadata-driven methodology and workflow process for providing translational research informatics support," *Journal of biomedical informatics*, vol. 42, no. 2, pp. 377–381, 2009.

[18] EarthSoft, "EarthSoft: Standalone EQuIS Data Processor (EDP) User Guide." http://www.dec.ny.gov/docs/remediation_hudson_pdf/edpuserguide.pdf, 2008.

[19] EarthSoft, "EQuIS 6 Enterprise: Workflow Automation & Dashboards." http://www.earthsoft.com/products/enterprise/, 2015 (accessed September 1, 2015).

[20] EarthSoft, "EarthSoft Corporate Overview EQuIS^{TM}." http://www.earthsoft.com/wp-content/uploads/2014/11/2014-Corp-Overview-Nov.pdf, 2014.

[21] W. Arlt, M. Biehl, A. E. Taylor, S. Hahner, R. Libe, B. A. Hughes, P. Schneider, D. J. Smith, H. Stiekema, N. Krone, *et al.*, "Urine steroid metabolomics as a biomarker tool for detecting malignancy in adrenal tumors," *The Journal of Clinical Endocrinology & Metabolism*, vol. 96, no. 12, pp. 3775–3784, 2011.

[22] Y. Kim, I. Koo, B. H. Jung, B. C. Chung, and D. Lee, "Multivariate classification of urine metabolome profiles for breast cancer diagnosis," *BMC bioinformatics*, vol. 11, no. Suppl 2, p. S4, 2010.

[23] C. Lanz, A. D. Patterson, J. Slavík, K. W. Krausz, M. Ledermann, F. J. Gonzalez, and J. R. Idle, "Radiation metabolomics. 3. biomarker discovery in the urine of gamma-irradiated rats using a simplified metabolomics protocol of gas chromatography-mass spectrometry combined with random forests machine learning algorithm," *Radiation research*, vol. 172, no. 2, pp. 198–212, 2009.

[24] S. E. Reichenbach, X. Tian, Q. Tao, D. R. Stoll, and P. W. Carr, "Comprehensive feature analysis for sample classification with comprehensive two-dimensional lc," *Journal of separation science*, vol. 33, no. 10, pp. 1365–1374, 2010.

[25] EarthSoft, "EQuIS Professional." http://www.earthsoft.com/products/professional/, 2015 (accessed September 1, 2015).

[26] DENAMIC, "Developmental Neurotoxicity Assessment of Mixtures in Children." http://www.denamic-project.eu/, 2015 (accessed September 1, 2015).

[27] "Marker view software." http://sciex.com/products/software/markerview-software.

[28] "Simca-p." http://umetrics.com/products/simca.

[29] "Sas." https://www.sas.com/en_us/home.html.

[30] J. Xia, N. Psychogios, N. Young, and D. S. Wishart, "Metaboanalyst: a web server for metabolomic data analysis and interpretation," *Nucleic acids research*, vol. 37, no. suppl 2, pp. W652–W660, 2009.

[31] H. Neuweger, S. P. Albaum, M. Dondrup, M. Persicke, T. Watt, K. Niehaus, J. Stoye, and A. Goesmann, "Meltdb: a software platform for the analysis and integration of metabolomics experiment data," *Bioinformatics*, vol. 24, no. 23, pp. 2726–2732, 2008.

[32] S. Gibb and K. Strimmer, "Maldiquant: a versatile r package for the analysis of mass spectrometry data," *Bioinformatics*, vol. 28, no. 17, pp. 2270–2271, 2012.

[33] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference* (S. van der Walt and J. Millman, eds.), pp. 51 – 56, 2010.

[34] "rpy2 package." http://rpy2.bitbucket.org/.

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[35] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: A structure for efficient numerical computation," *Computing in Science and Engg.*, vol. 13, pp. 22–30, Mar. 2011.

[36] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures.," *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[37] C. Ryan, E. Clayton, W. Griffin, S. Sie, and D. Cousens, "Snip, a statistics-sensitive background treatment for the quantitative analysis of pixe spectra in geoscience applications," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 34, no. 3, pp. 396–402, 1988.

[38] J. Gil and M. Werman, "Computing 2-dimensional min, median and max filters," 1996.

[39] A. M. Andrew, "Another efficient algorithm for convex hulls in two dimensions," *Information Processing Letters*, vol. 9, no. 5, pp. 216–219, 1979.

[40] F. Dieterle, A. Ross, G. Schlotterbeck, and H. Senn, "Probabilistic quotient normalization as robust method to account for dilution of complex biological mixtures. application in 1h nmr metabonomics," *Analytical chemistry*, vol. 78, no. 13, pp. 4281–4290, 2006.

[41] M. Dowle, T. Short, S. Lianoglou, R. Saporta, A. Srinivasan, and E. Antonyan, "data. table: Extension of data. frame," 2015.

[42] M. C. Chambers, B. Maclean, R. Burke, D. Amodei, D. L. Ruderman, S. Neumann, L. Gatto, B. Fischer, B. Pratt, J. Egertson, *et al.*, "A cross-platform toolkit for mass spectrometry and proteomics," *Nature biotechnology*, vol. 30, no. 10, pp. 918–920, 2012.

[43] R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, *et al.*, "Bioconductor: open software development for computational biology and bioinformatics," *Genome biology*, vol. 5, no. 10, p. R80, 2004.

[44] S. Gibb, "readMzXmlData: Reads Mass Spectrometry Data in mzXML Format," 2014.

[45] S. Gibb, "MALDIquantForeign: Import/export routines for maldiquant," 2015.

[46] D. Eddelbuettel, "Cran task view: High-performance and parallel computing with r," 2014.

[47] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online; accessed 2016-02-28].

[48] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.

[49] B. Worley and R. Powers, "Multivariate analysis in metabolomics," *Current Metabolomics*, vol. 1, no. 1, p. 92, 2013.

[50] P.-G. M. Nathan Halko and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," 2014.

[51] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[52] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 881–892, July 2002.

[53] K. H. Liland, "Multivariate methods in metabolomics–from preprocessing to dimension reduction and statistical analysis," *TrAC Trends in Analytical Chemistry*, vol. 30, no. 6, pp. 827–841, 2011.

[54] D. J. Vis, J. A. Westerhuis, D. M. Jacobs, J. P. van Duynhoven, S. Wopereis, B. van Ommen, M. M. Hendriks, and A. K. Smilde, "Analyzing metabolomics-based challenge tests," *Metabolomics*, vol. 11, no. 1, pp. 50–63, 2015.

[55] J. Xia, R. Mandal, I. V. Sinelnikov, D. Broadhurst, and D. S. Wishart, "Metaboanalyst 2.0Ž014a comprehensive server for metabolomic data analysis," *Nucleic acids research*, vol. 40, no. W1, pp. W127–W133, 2012.

[56] X. Wang, A. Zhang, Y. Han, P. Wang, H. Sun, G. Song, T. Dong, Y. Yuan, X. Yuan, M. Zhang, *et al.*, "Urine metabolomics analysis for biomarker discovery and detection of jaundice syndrome in patients with liver disease," *Molecular & Cellular Proteomics*, vol. 11, no. 8, pp. 370–380, 2012.

[57] A. Miyagi, H. Takahashi, K. Takahara, T. Hirabayashi, Y. Nishimura, T. Tezuka, M. Kawai-Yamada, and H. Uchimiya, "Principal component and hierarchical clustering analysis of metabolites in destructive weeds; polygonaceous plants," *Metabolomics*, vol. 6, no. 1, pp. 146–155, 2010.

[58] J. Hunter, D. Dale, and E. Firing, "matplotlib: Python plotting," 2012.

[59] L. Wilkinson and M. Friendly, "The history of the cluster heat map," *The American Statistician*, vol. 63, no. 2, 2009.

[60] J.-Y. Moon, H.-J. Jung, M. H. Moon, B. C. Chung, and M. H. Choi, "Heat-map visualization of gas chromatography-mass spectrometry based quantitative signatures on steroid metabolism," *Journal of the American Society for Mass Spectrometry*, vol. 20, no. 9, pp. 1626–1637, 2009.

[61] X. Zhao, J. Fritsche, J. Wang, J. Chen, K. Rittig, P. Schmitt-Kopplin, A. Fritsche, H.-U. Häring, E. D. Schleicher, G. Xu, *et al.*, "Metabonomic fingerprints of fasting plasma and spot urine reveal human prediabetic metabolic traits," *Metabolomics*, vol. 6, no. 3, pp. 362–374, 2010.

[62] L. Mengual, M. Burset, M. J. Ribal, E. Ars, M. Marín-Aguilera, M. Fernández, M. Ingelmo-Torres, H. Villavicencio, and A. Alcaraz, "Gene expression signature in urine for diagnosing and assessing aggressiveness of bladder urothelial carcinoma," *Clinical Cancer Research*, vol. 16, no. 9, pp. 2624–2633, 2010.

[63] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, (Berkeley, CA, USA), pp. 10–10, USENIX Association, 2010.

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org