Setup and Annotation of Metabolomic Experiments by Integrating Biological and Mass Spectrometric Metadata

Oliver Fiehn, Gert Wohlgemuth, and Martin Scholz

University of California, Davis Genome Center, GBSF Building
Davis, CA, 95616, USA
ofiehn@ucdavis.edu
http://fiehnlab.ucdavis.edu/

Abstract. Unbiased metabolomic surveys are used for physiological, clinical and genomic studies to infer genotype-phenotype relationships. Long term re-usability of metabolomic data needs both correct metabolite annotations and consistent biological classifications. We have developed a system that combines mass spectrometric and biological metadata to achieve this goal. First, an XML-based LIMS system enables entering biological metadata for steering laboratory workflows by generating '*classes*' that reflect experimental designs. After data acquisition, a relational database system (BinBase) is employed for automated metabolite annotation. It consists of a manifold filtering algorithm for matching and generating database objects by utilizing mass spectral metadata such as 'retention index', 'purity', 'signal/noise', and the biological information *class*. Once annotations and quantitations are complete for a specific larger experiment, this information is fed back into the LIMS system to notify supervisors and users. Eventually, qualitative and quantitative results are released to the public for downloads or complex queries.

1 Introduction

Technology advances during the last decade have opened new ways to approach cellular phenotypes. These advances are summarized today as '-omics' platforms which generate quantitative and qualitative data on cellular components such as mRNA transcripts, proteins, or metabolite levels (metabolomics [1]). Metabolomics is a comparatively inexpensive though reliable and informative tool to monitor metabolic states in a variety of different genetic or environmental perturbations. Both for testing and for verifying biological hypotheses, a number of explanatory variables and background information is needed to assist the interpretation (or induction) process. Specifically, there is no way to use data from –omic databases without explaining which biological designs were underlying the experiments. *With other words, data without metadata are junk.* It is a general consensus that scientific experiments and conclusions must be at least explained in such a way that, in

B. Ludäscher and L. Raschid (Eds.): DILS 2005, LNBI 3615, pp. 224 – 239, 2005.

principle, the experiments could be repeated. However, labeling experiments with (biological) metadata is clearly lagging behind descriptions of processes in the data generating technical platforms. It is just now that the metabolomics community has started to develop standards tracking the way from sample to sample processing, data acquisition, data export and normalization to statistics. The ArMet group [2] proposed a generalized framework including various modules to describe a metabolomics experiment. This framework does not detail which (biological or instrumental) metadata are essential to re-use metabolomic experiments for other queries or under other perspectives, and which ontologies need to be used. A related opinion statement on the minimal requirements for a metabolomic experiment (MIAMet) emphasizes the importance for traceable metabolic annotations [3] but does not further embark on biological metadata. A similar trend is seen in the more mature fields of proteomics (the PEDRo standard [4, 5]) and transcriptomics (the MIAME standard [6). For gene expression experiments, a study-annotator has been developed for describing experimental designs [7]. However, users need to fill 25 forms which relate to 68 tables, and understand and follow pre-defined ontologies that are not authorized by a wide consensus in the biological community.

For metabolomics, an extensive discussion forum is formed by the international working group on *Standard Metabolic Reporting Structures* (SMRS) led by the Imperial College, London, UK [8]. It was summarized in the 2.2 version of the draft document that 'It should be clear from the previous discussion that the state of biological standardisation for metabonomics experiments is currently non-existent.' [9] The very reason for this inadequacy may be the sheer difficulty to design a comprehensive yet simple schema (and user front end!) to capture the ingenuity of experimental designs in biology. We here present pragmatic solution that helps biological researchers defining their experimental design in a coherent and logical metadata structure, with a focus on user friendliness. Together with instrument-related metadata, this design information is used to generate the sample sequence schedule, to define the validity of detected metabolic peaks and to form the basis for statistical treatments of result data. However, we do not envision a direct comparability of the actual data readouts between different experiments: there are no two biological experiments that are totally identical. In fact, it is even difficult to achieve identical results from independent replica setups of experimental designs within a given biological laboratory. The reason for this difficulty in comparability is that there are many fuzzy factors contributing to the actual (metabolic) phenotype of a given individual organism that are hardly controllable in tight manners. Nevertheless, quantitative data outputs will be comparable with respect to trends and magnitudes of control of metabolism, even between laboratories or technology platforms used. In this respect, any information on biological metadata descriptions will enable researcher to (a) carry out own data interpretations and calculations to generate novel hypotheses or (b) combine and compare experiments that share similarities on higher abstraction levels such as 'abiotic stress in plant' which would comprise cold, heat, light or nutritional stress.

2 Hierarchical Metadata Defining Biological Experimental *Classes*

We have adopted the general framework laid out by the ArMet group (Architecture for Metabolomics) which consists of nine generic modules [2]:

1. ***Admin*:** Informal experiment description and contact details.
2. ***BiologicalSource*:** Genotype and specification of biological source material (BS).
3. ***Growth*:** Environments in which the biological material developed.
4. ***Collection*:** Procedures followed for gathering samples BS material.
5. ***SampleHandling*:** Handling and storage procedures following collection.
6. ***SamplePreparation*:** Protocols sample preparation prior to data acquisition.
7. ***AnalysisSpecificSamplePreparation*:** Protocols specific to data acquisition.
8. ***InstrumentalAnalysis*:** Process description of data acquisition including quality control protocols.
9. ***MetabolomeEstimate*:** The output of processed data including data processing protocols.

In the implementation period of ArMet, it was found that the accurate description of the biological background of a given sample is the most difficult, but also most important part of the framework. Many steps of modules 4-9 can be easily standardized or described since these are technical procedures that are always performed in a defined manner, at least for a specific routine protocol in a given laboratory. However, the biology experimental part is highly flexible and depends solely on the hypothesis underlying the study. Therefore we decided to use a flexible XML data structure, in order to match a large variety of experimental designs. Given the flexibility and breadth of biological studies, capturing all biological descriptors is technically and intellectually demanding, if not impossible. It is equally difficult to prescribe which of the (potentially very complex) steps of the biological designs are required from the users, and which are just optional. Furthermore, a very in-depth and comprehensive database structure implies that users face highly complex entry forms (and

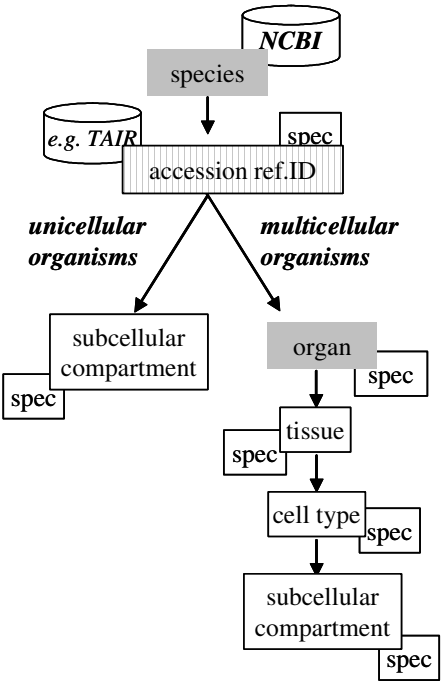


Fig. 1. Description hierarchy for *BioSource*. Use of controlled vocabularies is ensured for specific entries for which authoritative external databases have been assigned (such as NCBI). Others are cross-checked by dictionaries

underlying ontologies) which increases the risk of dummy entries, missing entries or to abstention from populating the database. We have therefore opted for a compromise: we request users to enter the minimal information that would also be required for publishing data in a peer-reviewed scientific journal. In addition we have implemented a structured way to capture this metadata reflecting the underlying biological design. For example, for some relationships and ontologies there are authoritative resources supporting the description of *BioSources* (BS). Besides species names and synonyms, the NCBI database [10](figure 1) supports taxonomic relationships, ultimately up to the top levels 'super kingdom' (arachae, bacteria, eukaryota, viroids and viruses). The underlying taxonomy can be used to distinguish unicellular microorganisms and multicellular (higher) organisms: the latter always consist of distinct 'organs' which may further be specified by tissue type, cell type or subcellular compartment that is under study. Microorganisms lack these and can only be further specified by potential subcellular compartments. For setup of an experiment, users can enter more than one species or more than one organ, each of which then may further get specified by additional information. Further authoritative databases are added that help specifying subgroups of species. For example, for the model plant *Arabidopsis thaliana* 831 ecotypes are notified in *t*he *A*rabidopsis *i*nformation *r*esource TAIR [11], and thousands of well-described Arabidopsis mutant lines, each with a specific ecotype genetic background. All these genetically different Arabidopsis lines are called 'accessions' and are assigned by a reference identifier in TAIR. As more and more biological communities establish such repositories, these are implemented in our experimental setup designs and made mandatory.

However, even on the level of 'organs' there are not many such compulsory lists. For plants, a comprehensive list of organs is given by plantontology.org [12], however, we have not yet identified an accepted standard for naming all mammalian organs, tissues, cell types or eukarytic subcellular compartments: in fact, this is a huge gap in ontology work [13] and frameworks describing relationships between hierarchical levels in biology. In such cases we gradually extend controlled vocabularies by (a) using publicly available lists such as tissue DB [14] that have not yet reached the level of a commonly accepted *de facto* standard and by (b) extending vocabularies used for experimental description in our own database after manual curation. All entries, include strings of flow text descriptions are automatically tested and corrected for spelling by dictionaries and synonym lists.

For a given experiment, all these entities together describe the number of different biological specimen to be tested. It is important to note that each experimental setup necessarily requires description of both *BioSource* and *Growth* condi- tions. It can be expected that metabolic responses on perturbation of growth conditions have at least the same magnitude as effects that are due to genetic changes. This observation is so general that it must be implemented as independent and equally important metadata into a design structure. The resulting biological setup will therefore always span a matrix $M = BioSource_{1\ldots n} * Growth_{1\ldots.n.}$

	growth history
BS_1	*N*
BS_2	*N*

Fig. 2. A simple experimental design *BioSource x Growth* for testing biological hypotheses. Each box in the matrices defines a *class* with $N \geq 6$ biological replicates as members

The simplest experimental design that can be devised may compare two different *BioSources* (figure 2), or, alternatively, the same type of *BioSources* under two different *Growth* conditions. There is no *BioSource* that has not been grown in a more or less defined manner. Therefore, the factor '*Growth*' is a general property for all biological specimens, however, for some organisms like human patients there is no detailed experimental design. In such cases, generic terms like 'western diet, age' may be used apart from potential treatments (see below) like therapies. This design is equivalent to the well known matrix 'Genotype *x* Environment' that is used in classical crop breeding. It is important to recognize that each of the different perturbations (*BioSource* or *Growth*) may result in different metabolic states which may be separated into groups or *classes* for statistical analysis of the metabolic levels. For any given experiment, parts of the growth conditions are identical to all *BioSources*. Otherwise, any comparison between the *classes* would be impossible and senseless! These past growth conditions may be described as a growth history G_1. For each species, a minimum set of growth metadata is required whereas other metadata are optional. In plant molecular biology, a single growth history may be defined for which details would be required on sowing and harvest date, harvest time, daylight period, light intensity, humidity, developmental stage, growth medium and type of growth location. Unfortunately, there is no consensus or ontology for this minimum set of 'background *Growth* metadata'. For a given biological field, experimental descriptors may have been passed on as 'necessary' by journal editors, reviewers and university courses. For example, it is most common to give details on light fluxes in plant biology when explaining the experimental setup in environmental growth chambers. However, it is far less common to say which actual light source was used and the emission spectrum of this, although it is known that plants do react very sensitively on higher or lower levels of red and blue parts of the light spectrum.

In the same way like molecular biologists will vary the genotype (or organ or cell type of a given genotype), physiologists and toxicologists will study variations of *Growth* conditions (including developmental stages) and external environmental impacts such as drugs ('treatments'). Each of these growth conditions may again split into different attributes and properties. An example would be 'variation of temperature' in a cold stress experiment in plant physiology, which might utilize high, low, and control temperatures, extending the matrix of BS1 and BS2 (each with three organs) 6 x 3 = 18 biological groups or *classes*. It is important to note here that the generation of these *classes* as derived conceptual information from the biology metadata is fed into various other locations within the mass spectral annotation system, most importantly into the data acquisition schedule, the metabolite verification algorithm (see section 3.1) and the statistics workflow. It can easily be imagined that this treatment might be followed in a time dependent manner, which would further increase the matrix (and the complexity of the experimental design). If four time points were included, the overall sample matrix would then be of a dimension of 6 x 3 4 = 72 different biological *classes*. In order to perform statistical tests on the resulting metabolomic data, it is wise to use more than six samples per biological *class*, say 10 independent plants. Consequently, 720 samples would be delivered for metabolite analysis: an undertaking that can indeed be carried out in a reasonable time frame and budget in metabolomics, but which would be less feasible for more costly and slower transcriptomic or proteomic experiments (i.e. in case

global gene or protein expression levels were to be analyzed). In this respect, metabolomics is different to other –omics techniques because very detailed and structured experimental designs are more likely to be performed with sufficient replicate numbers to carry out statistical tests on the resulting experimental data. In principle, a hierarchical tree of '*Growth*' may be drawn (figure 3).

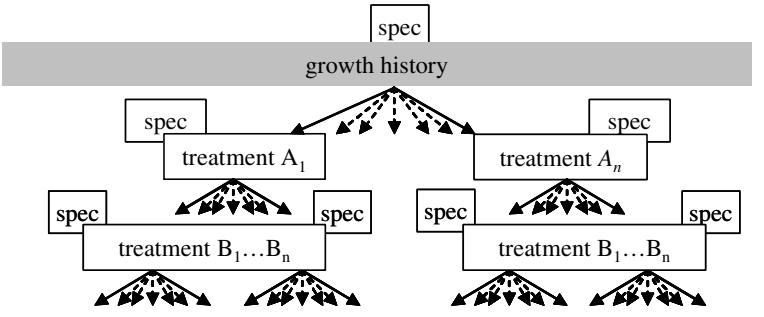


Fig. 3. Flowchart for the description of '*Growth*'. Very complicated experimental designs may be performed, based on the physiological tests that biologists devise. Further specifications (spec) may be entered but are not required

				BS_1 'control'				BS_2 'mutant'			
				blood	liver	heart	kidney	blood	liver	heart	kidney
growth history	$drug_1$	$dose_1$	$time_1$	*N*				*N*			
			$time_2$	*N*				*N*			
			$time_3$	*N*				*N*			
			$time_4$	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*
		$dose_2$	$time_1$	*N*				*N*			
			$time_2$	*N*				*N*			
			$time_3$	*N*				*N*			
			$time_4$	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*
	$drug_2$	$dose_1$	$time_1$	*N*				*N*			
			$time_2$	*N*				*N*			
			$time_3$	*N*				*N*			
			$time_4$	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*
		$dose_2$	$time_1$	*N*				*N*			
			$time_2$	*N*				*N*			
			$time_3$	*N*				*N*			
			$time_4$	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*

Fig. 4. Pharmacological comparison of two rat strains, four organs, and treatment with two drugs with two different doses which is followed at four time points

This *Growth* design hierarchy is obviously dependent on the underlying metadata from *BioSource*: it does not make sense to require 'light conditions' from a human blood plasma study, and it also is not reasonable to request 'gender' from a plant. However, for a given *BioSource* there is set of growth metadata that is always requested (such as age, sex and other parameters for human samples). The usability of

this flowchart for a variety of areas of biological research is exemplified by a pharmacological test setup. The flexible *BioSource x Growth* matrix allows an easy setup of this experiment which may consist of only two rat strains (control and mutant line), on which the effects of two drugs in two different doses is tested on four time points in the blood plasma, and (at end time point 4), also for liver, heart and kidney. Such a pharmacology design is depicted in figure 4. It is important to note that each individual biologist who defines an experiment also defines which metadata are mandatory: in this respect, this metadata layout does not prescribe the biologist what to do but helps scientists to describe the underlying idea behind the design. For both *BioSource* and *Growth*, users may want to add further specific attributes to tables. These cannot be restricted by ontology databases or dictionary comparisons. An example could be 'patient ID codes' for clinical samples.

2.1 Technical Implementation of SetupX

We call our system 'SetupX' which sets up experimental design *classes* and subsequently also manages laboratory workflows and user queries. Although developed for a certain purpose, SetupX' architecture allows the system to be used in other environments after small adaptations and configurations. A modular structure of this system guarantees that it is reusable, easy to maintain and expandable [15]. All separate functions are offered and used by SetupX in different smaller modules. Communication and interaction between these modules is interceded by the mediator layer. Therefore, different modules can be placed into other environments in short time without requiring major modifications.

Currently there are two access possibilities implemented that allow use of six core modules of SetupX. Any external access to the core modules is shielded by the mediator. One way of external access is the web service module which is based on SOAP (Simple Object Access Protocol) and which allows a platform independent administration and use *via* XML communication. The database is a native XML database that supports storage of metadata in true XML and that also supports the query language XQuery. Native XML databases support data that are not underlying a fixed schema, which is difficult or almost impossible using relational databases.

A swing-user-interface is connected to the web service module for system administration. The second type of external access is the JSP/Servlet module, which generates the standard user-interface for external collaborators and laboratory staff. Part of this user interface is the dynamically generated form for defining biological experiments and *classes*. The six core functions of SetupX include user communication and management, interaction with BinBase, generating and writing schedules for the mass spectrometer (based on *class* information), and eventually the definition of the laboratory workflow itself (figure 5).

1. User communication and management

Information stored in BinBase and SetupX must be regarded as confidential. This policy is enforced by defining user authorisations for the different roles. UC Davis users will use their account granted by the campus' Kerberos system. With this account, additional personalized information (e.g. affiliation, address, email, telephone etc) is referenced by SetupX through LDAP-directories (Lightweight

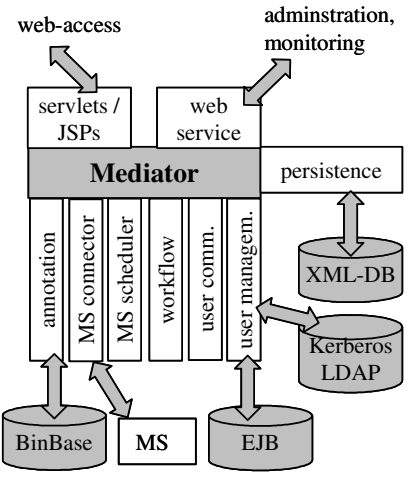


Fig. 5. Modular structure of SetupX and its connected components

Directory Access Protocol). For non-campus users, SetupX needs to generate an internal authentication. Users need here state once their personal information. Users, and particularly metabolomics staff members, can check the status of laboratory workflows directly by logging in, or are notified by email when predefined workflow parts have been finished or when problems occurred.

2. Interaction with BinBase
Users can request BinBase annotation result files through SetupX which activates the BinBase export function by EJB (Enterprise Java Beans) and JMS (Java Messaging Service, a Java interface to Message-Oriented Middleware). BinBase itself requests information about *class* labels of samples using EJBs.

3. Generating and writing schedules for the mass spectrometer
Through the user interfaces, *classes* and the number *N* of samples per class are entered. SetupX uses this information to generate a run sequence schedule for the mass spectrometer and to communicate this schedule to the instrument in an instrument-specific format. Once the sequence has been started by laboratory staff, an internal scanner is used to grab any information delivered by the mass spectrometer with regards to success or potential failure messages. This information is then fed back into SetupX using the same instrument-specific connector.

4. Workflow definition and surveillance
A workflow manager defines the execution sequence of the different modules in order to allow flexible adaptations to new laboratory requirements. In order to make the system independent from the current laboratory workflow definition, a workflow is compiled in a single configuration file. This allows easy update of workflows in case of changes of laboratory protocols or data processing modifications.

5. Persistence and document module
SetupX stores all documents such as experiment description, sample definition etc. as XML files. Consequentially a genuine XML database is used as repository for which XQuery [16] serves as powerful query language. We found XML structure an adequate choice given the fact that the definition of biological *classes* does not allow a unique structure. XML is known as a simple, very flexible text format, which allows the definition of the hierarchy used for the definition of the experiment in an excellent way. Storing this information in a relational database management system would be inappropriate because a large overhead would be generated for mapping this information from XML to the relational structure and back. Speed is not an important aspect for SetupX because no large computational queries are foreseen. Furthermore

both the input and export format is XML. Since the database stores the information as unmodified XML the data has never to be mapped.

6. Graphical user interface
One of the main requirements of the new developed LIMS system was that it had to be so user friendly that every user had to be able to use it without any introduction by the staff of the lab and even without reading manuals. The major request was technical – the user interface had to build and generate itself dynamically, because the structure which is represented by the graphical user interface will never, as mentioned above, be fixed to a final set of attributes. We have first explored using a Swing User Interface, similar to the PEDRo-approach. The experimental class structure was defined through an XML schema, and based on this schema the Client-Application created the graphical user interface. However, this schema driven client never matched the requirements of user friendliness and usability, because any fine tuning of *class* definitions and sample specifications were constrained by the technical limitations of the underlying XML schema. Instead we have implemented a server side dynamical created user interface based on Java Server Pages and Java Servlets. This solution is more independent from the experimental design than XML schema. It is therefore possible to add any functionality to this interface that can be implemented in code including functions like real time vocabulary checks or even the adoptions of the user interface to the selected items.

2.2 Experimental Metadata Supporting Other ArMet Modules

Some ArMet modules demand information that is usually stored in classic LIMS implementations such as user logins and user rights. For our case, slight adaptations were needed because many biological experiments are owned by more than one user: it is mandatory in our LIMS implementation to name the principal investigator (usually a faculty member), but also in addition to name the person who was responsible for performing the experiment (who may be research associate or staff). Other modules may be dependent on a given laboratory setup or a given *BioSource*: protocols to prepare samples from plants for metabolomic surveys may be totally inadequate for profiling of human blood plasma.

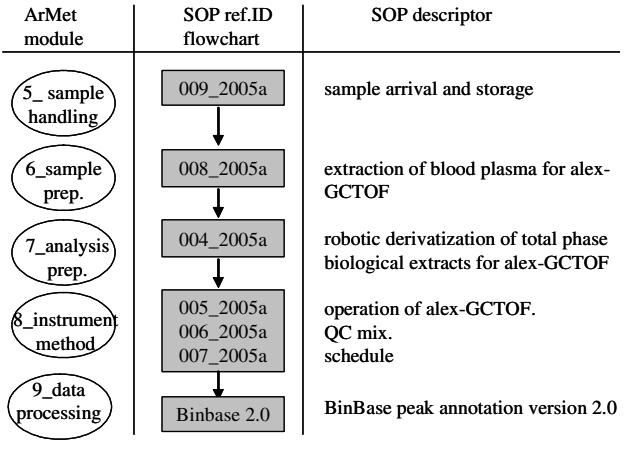


Fig. 6. Schematic flow diagram of Standard Operating Procedures (SOPs) in an example for an experiment with *BioSource*: human blood plasma samples

In order to ensure and monitor long term data quality and reusability, it is good laboratory practice

to perform any work by so-called 'Standard Operating Procedures' (SOPs), both in industrial and semi-industrial analytical environments such as academic core laboratories. Such SOPs include all characteristics needed for direct implementation of sources of metadata into the BinBase system: they include authoritative codes, identifier numbers, clear descriptions of necessary steps and also allowed deviations from protocols. An SOP differs from an academic laboratory protocol in that it must clearly lay out all aspects of a procedure. If a single item of the procedure is changed, it is necessary to state the reason for change, acquire data proving the validity of the change, reinstate permission by the laboratory authority (e.g. the principal investigator) and generate a new SOP number.

Once an SOP is laid out for e.g. sample preparation of a given *BioSource* or data acquisition procedure, it can be made mandatory in a LIMS workflow structure (figure 6). The validity area of SOPs is always clearly defined, but there may be features in the details of SOPs that are shared with external SOPs like the generalized type of the instrument (example in figure 6: a gas chromatography coupled to mass spectrometry) or the type of sample preparation (example in figure 6: cold protein precipitation, silylation). Such higher levels of metadata descriptions yet need to be developed and cannot be made mandatory at present. For example, it is an experience that some analytical instruments are affected by mid-term technical drifts (e.g. in sensititvity). Often, the factors underlying these technical drifts are not well understood and can only be partly controlled. The bottom line of metabolomic experiments is to derive structured information from the acquired data (e.g. by multivariate statistics) and to interpret resulting data clusters by biological metadata. It is obviously of utmost importance that this metabolomic data structure is not affected by non-biological factors such as machine drift. A means to ensure this (apart from instrument quality control) is a randomization of all samples in a sequence, so that each *class* is, on average, affected in the same magnitude as all other *classes*. The easiest way to ensure this is by a random number generator, however, in the laboratory this is almost impossible to put into practice. Therefore, SOP 007_2005a envisions a square root blocking schedule of all replicate samples of each class as compromise between total randomization and laboratory practicability:

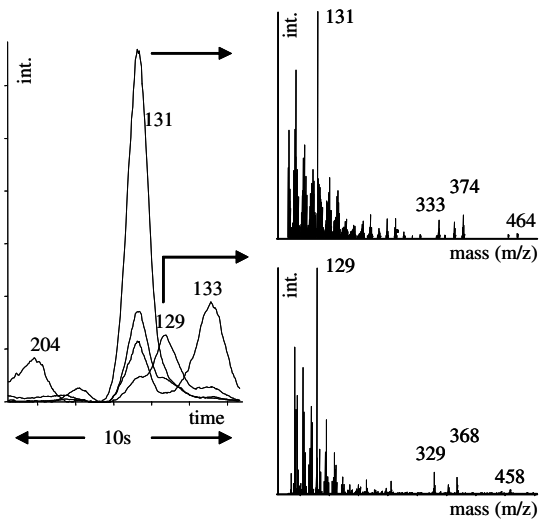
$$n_{block} = \sqrt{N_{class}} \quad (1)$$

If a *class* contains a total N=6 biological replicates, these would be randomized in three blocks of n=2 duplicates over the total instrument run sequence; if a *class* contains 16 biological replicates, these would be blocked into four blocks of four replicates. In summary, the SetupX module generates *classes* via biological metadata and enforces with this information a certain run sequence in the analytical laboratory.

3 Mass Spectral Annotation and Quantitation: BinBase 2.0

All samples are subjected to metabolome data acquisition by automatic liner exchange for gas chromatography/time of flight mass spectrometry (alex-GCTOF). The general output of this instrument is a three dimensional raw data matrix of (time x mass x intensity), which results in 10.8 mio. raw data points for a single sample (415 masses/spectrum x 1300 s x 20 mass spectra/s).

However, biological researchers can only interpret such data matrices if these are transformed into two dimensional data matrices (metabolite x intensity), since metabolite references are found in chemical or biochemical databases like CAS and KEGG and can thus be linked to other important biological objects like proteins and genes. The objective here is therefore to turn (time x mass) information into 'metabolite' annotations in a routine, but completely unbiased way, and to enable queries in experimental sets of such data matrices.



retention time	1135.23
unique mass	374
base mass	131
signal/noise	898.45
purity	0.067500
apex masses	82+83+84+85…
area	6,9E+07
height	2,2E+06

retention time	1136.58
unique mass	368
base mass	129
signal/noise	34.194
purity	0.62075
apex masses	129+255+353….
area	6,4E+05
height	1,1E+04

Fig. 7. Deconvolution of raw metabolomic data. Left panel: Overlay of 4 out of 415 measured mass elution profiles (10 s of a total run time 1350 s, profiles for ions m/z 129, 131, 133, 204). Mid panel: Deconvoluted mass spectra of two adjacent, co-eluting peaks with Δtime = 1.35 s. Right panel: instrumental metadata labelling these two peaks. Mass spectra and metadata serve as raw data input in Binbase 2.0

It is beyond the scope of this paper to outline theory and concepts of analytical mass spectrometry. It is important to know, though, that in the instrument each metabolite will fragment into more than one mass which will be detected in a finite time frame with an approximately Gaussian intensity time course and identical mass intensity ratios across this 'elution' time course. This time course is called a 'peak' with a unique mass/intensity pattern (called 'mass spectrum'). The peak intensity maxima define the first kind of instrumental metadata, called 'retention time' (fig. 7). It is unavoidable in metabolomics that peaks overlap (co-elute) since a metabolome of a given sample easily comprises over 1,000 different metabolites. Many mass fragments may be shared between co-eluting peaks. Therefore, the first step of the algorithm is to deconvolute [17] or purify mass spectra from co-eluting peaks, with appropriately assigning the intensity of shared masses to each peak. For this deconvolution we utilize the instrument vendor's software ChromaTOF 2.25. This software detects peaks in an unbiased way and exports one deconvoluted spectrum

per peak. In subsequent sections 'peaks' and 'spectra' are therefore used as synonyms. After deconvolution, a chromatogram comprises some 400-800 spectra, or a daily output of some 20,000 spectra per day and instrument. BinBase 2.0 then imports these spectra with accompanying metadata such as the 'unique (model) masses' that best describe the presence of a peak in the local environment. Further instrumental metadata are 'peak purity' (an estimate of the number, proximity and similarity of co-eluting peaks), 'signal/noise' (an estimate of peak abundance), 'apexing masses' (all masses that share maximum intensity with the peak maximum of the unique mass) and other.

3.1 The Filtering Algorithms in BinBase 2.0

Each sample will generate a different number of deconvoluted metadata-labelled spectra. Unfortunately, metabolomic mass spectrometry data sets contain numerous spurious and noisy spectra which need to be detected and deleted prior to annotating and aligning the remaining spectra, and this needs to be performed for multiple samples (n>1,000) and eventually, multiple of such large experiments. In addition, there may be deconvolution errors reported by ChromaTOF which need to be detected and eliminated. We therefore set out to develop a filtering algorithm that enables metabolite detection and quantification concurrently with automatic extension of metabolic libraries.

The objective of BinBase 2.0 therefore is to three-fold: (a) to annotate all exported spectra to known metabolic peaks that are already compiled as BINs in the database, (b) to automatically add new spectra to the list of BINs and (c) to allow dynamic user queries to export quantitative and qualitative metabolomic information after spectra of all *classes* have been annotated. A BIN is defined as a valid entry in the BinBase that has matched all mass spectral, instrumental and *class* metadata thresholds. In addition to the instrumental metadata, each BIN consists of a set of properties: mass spectrum, retention index (RI), quantification mass, list of unique masses, and a unique identifier number. BINs can be further qualified by super users with *1…n* properties that link further metadata such as 'metabolite name', 'ID code referring to external metabolic databases', 'list of synonyms' or else.

The general algorithm from spectra import to user query export is depicted in figure 8. It starts with importing and storing the .csv data files from all samples of an experiment. The algorithm proceeds by validating all spectra of a sample: check for presence and relative abundance of the unique ion, for presence of all apexing masses in spectrum, for deconvolution error dips, and for the number of spectra per chromatogram that exceed apex intensity thresholds and for the total number of thus detected deconvolution errors. Chromatograms that do not fulfil the latter two criteria will only be used for peak matching, but not for BIN generation. The algorithms then searches spectra of marker compounds that were physically spiked into the samples before data were acquired by using parametrized identification thresholds. With these marker compounds, retention indices (RI) are calculated from retention times to allow retention alignment. This is needed to counteract sample-to-sample retention shifts in the data acquisition procedure. The RI calculation is performed by polynomial regression because absolute and relative retention time shifts markedly differ from linear regressions at early and at late retention times. RIs are never altered or

manually adapted for a given data acquisition method, however, they will differ if chromatographic methods are changed. The algorithm then continues by sequentially (*seq*.) selecting all spectra by decreasing intensity (s/n) and testing, whether spectra can be annotated as existing BINs or, if they fail this annotation, if spectra could become new BINs. These decisions work through various filters: first, spectra need to fit into a retention index window, then they need to be labelled with a unique mass that is included in the BIN list of unique masses, afterwards they need to pass a mass spectral similarity filter (sim) that has different thresholds based on the intensity (s/n) and purity of the spectra, and last, spectra need to pass the isomer filter (iso) that selects the best of potentially several matching spectra for a given BIN. The similarity filter currently uses the INCOS algorithm [18], but in principle also other rules could be applied. Spectra that are sorted out in the isomer filter might still be able to match other (neighbouring) BINs and are therefore fed back into the annotation algorithm. Spectra that fail annotation to any existent BIN may generate new BINs. For this, they first need to pass mass spectral quality thresholds (MS) that are based on purity and intensity. Thresholds for the MS filter are more draconic than for the similarity filter to ensure that only abundant and pure spectra potentially become new BINs.

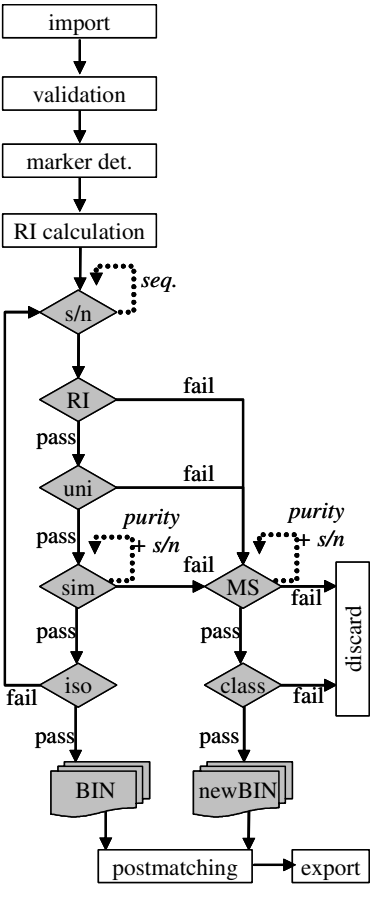


Fig. 8. Algorithm for peak annotation and BIN generation. For details, see text

Ultimately, a potential new BIN must pass the *class* filter before being validated. This filter demands that a new BIN is detected in at least 80% of all samples of a *class* in order to ensure that this BIN can be supposed to be a genuine metabolic entity and not a spurious contamination. This is also the basic reason why at least N=6 replicates of a given *class* need to be analysed, in order to ensure some level of statistical significance. Once all spectra of all *classes* of a given biological experiment have been annotated, the list of BINs is complete. Then, all spectra are again matched against the BIN list (postmatching) in order to warrant that all BINs (including the new BINs that were generated later in the process) are searched in all samples. Another reason for the postmatching process is that for some samples, spectra may not have passed the (higher) MS thresholds in the BIN generation but would pass the (lower) similarity thresholds in BIN annotation. Therefore, only by final postmatching the eventual result file can be regarded as complete. During the export process, each spectrum is quantified based on intensity of the BIN quantifier mass which is either

manually set by a super user or (as default value) it uses the 'unique mass' metadata during BIN generation. Various formats can be used for the final data export, depending on the user's needs. To our notion this is the first published attempt to align and annotate (biological) mass spectra by both instrument-related and biological metadata.

3.2 Technical Implementation of BinBase 2.0

Spectra filtering and BIN databasing is performed in separated modules: it is not advisable to calculate values within a database but use DBs exclusively for queries and data handling. We have employed an SQL 97 conforming database for an efficient data administration and query. The newer SQL 2003 specification was not yet supported by all open source databases. In order to be independent from a specific (supported) database type such as Oracle or SAPDB we have used Java database connectivity (JDBC). It was carefully avoided to program any functions that would be specific to a certain DB type.

BinBase 2.0 predominately consists of *1…n* table relationships. It is interesting to note that we have implemented the two modules, BinBase and SetupX, in two different database structures: for BinBase 2.0, an SQL structure was found to be advantageous due the faster access that is achieved by relational databases with fixed structure, compared to the more flexible but slower XML structure which was used for the (flexible) SetupX system. Furthermore are SQL based systems more mature, offering a wide variety of public or commercial products. For example it is unproblematic to use either Oracle or SAPdb because only minimal adaptations of SQL queries are needed (if programming was done conforming to standards, and if vendor-specific extensions were not used). The largest problems we have encountered were found in storing of all mass spectra. Spectra are imported into BinBase as strings which we first approached to be separated and stored in tables. However, we detected that query times exponentially slowed down with increasing numbers of rows. Therefore spectra are now stored as 'character large objects' (CLOB) which are dynamically transformed when needed. This procedure has also slowed down performance rates, however, it was found to be still faster than querying tables. The BinBase database itself is configured via XML files, which was found to be a simpler and more flexible solution compared to INI files. Furthermore this configuration offered the possibility to dynamically upload new implementations of the used interfaces *via* Class.forName().

Other components such as SetupX or web interfaces are linked via EJB (Enterprise Java Beans) and JMX (Java Management Extensions). The JMX components enable starting, stopping or querying the status of implemented servers. The EJBs allow querying which samples are being processed or exported during longer sequences. XDoclet was used for generating EJB/JMX configuration files and helper classes. Three servers are implemented: an import server (for importing, matching and BIN generation), a postmatching server (for regular postmatching over the complete database) and a transformation server (for exporting data and file formatting). Currently, plain text, MS Excel and XML is supported. These servers can run independently or together with the EJBs on the JBoss application server.

Finally, front ends have been implemented. A plugin based on Eclipse 3/SWT is used as administrative front end. It includes visualization based on JFreeChart and allows database queries *via* a Hibernate framework. The Hibernate framework supports mapping database documents to objects. Dynamic SWT-tables and visualizations are created from these objects via Java Reflection-API. Therefore, these tables visualize the database contents, for example, all BINs with corresponding metadata. BINs can be modified or manually erased by super users only. A persistence layer is used for user access and user defined queries.

4 Conclusions

This is the first description of a combined system which uses the description of biological experiments to validate metabolic peaks from mass spectra and corresponding mass spectral metadata. Earlier publications have not detailed algorithms how (processed) mass spectrometric peaks are automatically validated and added to a database, but rather focused on database query options [19] or on comparing chromatograms on the base of summing mass spectral intensities [20, 21], instead of alignments of deconvoluted mass spectra and annotation of individual metabolites. The implementation of BinBase 2.0 enables annotating up to 0.5 mio. spectra per day which is far higher than the current production rate of 20,000 spectra/day at the UC Davis Genome Center metabolomics facility. A comparison of manual and automatic validation of such chromatograms will be presented in a bioanalytical journal for the comparison of 1,200 potato tubers from a field trial.

Further improvements will work on parallelization of processes for peak detection and postmatching and on integration of further peak metadata (such as peak tailing factor or profile purity) for automatic flagging of problem cases. SetupX development will consist of further integration of ontologies with a focus on improvements in user friendliness and reducing the time needed for defining each experiment. Ideally, SetupX would parse the required biological metadata directly from strings that are pasted by users into a single web form, and would only ask for additional information if needed. To this end, however, the abilities of text mining approaches have not been developed far enough yet.

References

1. Fiehn, O.: Metabolomics – the link between genotype and phenotype. Plant Mol. Biol. 48 (2002) 155-171
2. Jenkins, H., Hardy, N., Beckmann, M. *et al.*: A proposed framework for the description of plant metabolomics experiments and their results. Nat. Biotechnol. 22 (2004) 1601-1605
3. Bino, R.J., Hall, R.D., Fiehn, O. *et al.*: Potential of metabolomics as a functional genomics tool. Trends Plant Sci. 9 (2004) 418-425
4. Garwood, K., McLaughlin, T., Garwood, C., *et al.*: PEDRo: A database for storing, searching and disseminating experimental proteomics data BMC Genomics 5 (2004) Art. No. 68

5. Jones, A., Hunt, E., Wastling, J.M., Pizarro, A., Stoeckert, C.J.: An object model and database for functional genomics. Bioinformatics 20 (2004) 1583-1590
6. Ball, C.A., Brazma, A. , Causton, H. *et al*.: Submission of Microarray Data to Public Repositories. PLoS Biology 2 (2005) e317, 1276-12773
7. Manduchi, E., Grant, G.R., He, H. *et al*.: RAD and the RAD study-annotator: an approach to collection, organization and exchange of all relevant information for high-throughput gene expression studies. Bioinformatics 20 (2004), 452-459
8. The Standard Metabolic Reporting Structure -An Open Standard for Reporting Metabolic Data . (http://www.smrsgroup.org/ - March 09, 2005)
9. Lindon, J.C. (ed.) Standardisation of Reporting Methods for Metabolic Analyses: A Draft Policy. Document from the Standard Metabolic Reporting Structures (SMRS) Group. *4.5. Summary*, p. 10. (http://www.smrsgroup.org/documents/SMRS_policy_draft_v2.3.pdf - February 01, 2005)
10. Wheeler, D.L., Barrett, T., Benson, D.A. *et al.*: Database resources of the National Center for Biotechnology Information. Nucl. Acids Res. (2005) D39-D45 Sp. Iss. SI
11. Rhee, S.Y., Beavis, W., Berardini, T.Z., *et al*.: The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community. Nucl. Acids Res. 31 (2003) 224-228
12. Bruskiewich, R., Coe, E.H., Jaiswal, P. *et al*: The Plant OntologyTM Consortium and Plant Ontologies. Comparative and Functional Genomics, 2002, 3(2), 137-142
13. Loranger, S., Higgins, G., Sen, S., Kelly H. The digital human: Towards a unified ontology. Omics 7 (2003), 421-424
14. http://tissuedb.ontology.ims.u-tokyo.ac.jp:8082/tissuedb/ May 16, 2005
15. Erich Gamma et al.: Design patterns : elements of reusable object-oriented software. (Addison- Wesley, Reading, Massachusetts 1995)
16. W3C XQuery 1.0: An XML Query Language. W3C Working Draft. (http://www.w3.org/TR/xquery/ - Feb. 12, 2005)
17. Stein, S.E.: An integrated method for spectrum extraction and compound identification from gas chromatography/mass spectrometry data. J.Am.Soc. Mass Spectrom. 10 (1999) 770–781.
18. McLafferty, F.W., Zhang, M.Y., Stauffer, D.B., Loh, S.Y.: Comparison of algorithms and databases for matching unknown mass spectra. J.Am.Soc. Mass Spectrom. 9 (1998) 92-95
19. Kopka, J., Schauer, N., Krueger, S. *et al.*: GMD@CSB.DB: the Golm Metabolome Database. Bioinformatics 21 (2005), 1635-1638
20. Jonsson, P., Gullberg, J., Nordstrom, A. *et al.*: A strategy for identifying differences in large series of metabolomic samples analyzed by GC/MS. Anal. Chem. 76 (2004), 1738-1745
21. Duran, A.L., Yang, J., Wang, L.J., Sumner, L.W.: Metabolomics spectral formatting, alignment and conversion tools (MSFACTs). Bioinformatics 19 (2003) 2283-2293