Design and development of a comprehensive data management platform for cytomics: cytomicsDB

Larios Vargas, E.

Chapter 2

CytomicsDB architecture

In this chapter, we propose a platform for managing and analyzing HTS images resulting from cytomics screens taking the automated HTS workflow as a starting point. This platform seamlessly integrates the whole HTS workflow into a single system. The platform relies on a modern relational database system to store user data and process user requests, while providing a convenient web interface to end-users. By implementing this platform, the overall workload of HTS experiments, from experiment design to data analysis, is reduced significantly. Additionally, the platform provides the potential for data integration to accomplish genotype-to-phenotype modeling studies.

This chapter is based on the following publications:

- K. Yan, E. Larios, S. LeDévédec, B. van de Water and F. J. Verbeek. **Automation in cytomics: Systematic solution for image analysis and management in high throughput sequences**. In Proceedings IEEE Conf. Engineering and Technology (CET 2011), volume 7, pages 195–198. 2011.
- E. Larios, Y. Zhang, K. Yan, Z. Di, S. LeDévédec, F. Groffen, and F.J. Verbeek. **Automation in cytomics: A modern rdbms based platform for image analysis and management in high-throughput screening experiments**. In Proceedings of the 1st Int. Conf. on Health Information Science, volume 7231, pages 76–87, 2012.

2.1 Introduction

Recent developments in microscopy technology allows various cell and structure phenotypes to be visualized using genetic engineering. With a time-lapse image-acquisition approach, dynamic activities such as cell migration can be captured and analyzed. When performed in large-scale via robotics, such approach is often referred to as a High-Throughput Screening (HTS). At the work floor this is often called "screen". In cytometry, HTS experiments, at both cellular and structural level, are widely employed in functional analysis of chemical compounds, antibodies and genes. With automated image analysis, a quantification of cell activity can be extracted from HTS experiments. In this manner, biological hypothesis or diagnostic testing can be verified via machine learning using the results from the image analysis. HTS experiments, supported by automated image analysis and data analysis, can depict an objective understanding of the cell response to various treatments or exposures.

In this chapter, we set our scope to the bioinformatics aspects of HTS. An HTS experiment starts with the design of a culture plate layout containing $N \times M$ wells in which the cells are kept, cultured and to which experimental conditions are applied. The response of the cells is then recorded through time-lapse (microscopy) imaging and the resulting time-lapse image sequence is the basis for the image analysis. The design of the plate layout is a repository of the experiment as a whole. From a study of the workflow of biologists, we have established an HTS workflow system.

Currently, spreadsheet applications are commonly used for bookkeeping the information generated during the workflow of HTS experiments. This approach has many drawbacks. It usually takes months to finish a complete experiment, i.e., from the plate design to the data analysis. Furthermore, images produced by the HTS experiments are not linked properly with their metadata and the analysis results. This scenario makes it difficult to do a proper knowledge discovery. So, most of the process within the workflow of HTS experiments are developed manually, which is highly prone to man made errors. Moreover, spreadsheets often differ in format and are not stored in a central place. This makes it hard for scientists from even the same institute to search, let alone to disclose their results in a uniform and efficient way.

To eventually tackle all these issues, we propose an *HTS platform* for managing and analysing cytomic images produced by HTS experiments. The platform seamlessly integrates the whole HTS workflow into a single system and provides end-users a convenient GUI to interact with the system. The platform consists of a layered architecture. First, an end-user layer that is responsible for the interaction with the scientists who perform different HTS experiments in cytomics. Then, the middleware layer that is responsible of the management of secure and reliable communication among the different components in the platform. Finally, a database-computational layer, in charge of the repository and execution of the image and data analysis.

Preliminary tests show that by using this platform, the overall workload of HTS

experiments, from experiment design to data analysis, is reduced significantly. This is because, among others, in the HTS platform, the design of plate layout is done automatically. Using spreadsheets, it takes an experienced biologist one week to manually finish the mapping of 400-600 gene targets, while it takes less than a day to use the plate design module in the HTS system. It also enables queries over datasets of multiple experiments. Thus, automation in cytomics provides a robust environment for HTS experiments. To sum up, the contributions of this work include:

1. Establishing a workflow system of the HTS experiments (Section 2.2).
2. An integrated platform to automate data management and image analysis of cytomic HTS experiments (Section 2.3).
3. The design of the database to store (almost) all data produced and used in the HTS experiments (Section 2.4).

Finally, we discuss related work in Section 2.5 and conclude in Section 2.6.

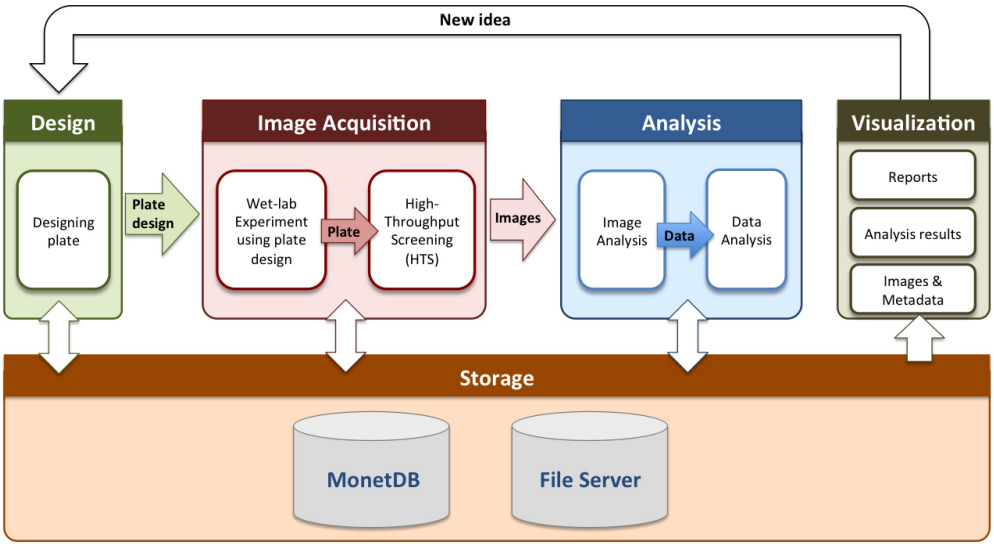


Figure 2.1: *Automated workflow of an HTS Experiment*

2.2 Automated Workflow of the HTS experiments

An automated workflow of a general HTS experiment is shown in Figure 2.1, where the typical stages are depicted separately. In this chapter, we describe the four functional modules in this *HTS workflow*: (1) plate design, (2) image analysis, (3) data management, and (4) pattern recognition (YLL+11).

2.2.1 Plate Layout Design Module

The design of a plate is considered as the cornerstone for an HTS experiment. Therefore, we have developed a Graphical User Interface (GUI) in our HTS platform to construct the layout for a plate (see Figure 2.2). The GUI allows end-users to rapidly deploy, modify and search through plate designs, to which auxiliary data such as experimental protocols, images, analysis result and supplementary literature is attached. In addition, the plate design provides a fast cross-reference mechanism in comparing data from various origins. This module is also used as the front end for the visualisation of results such as using heat maps, cell detection or motion trajectories.

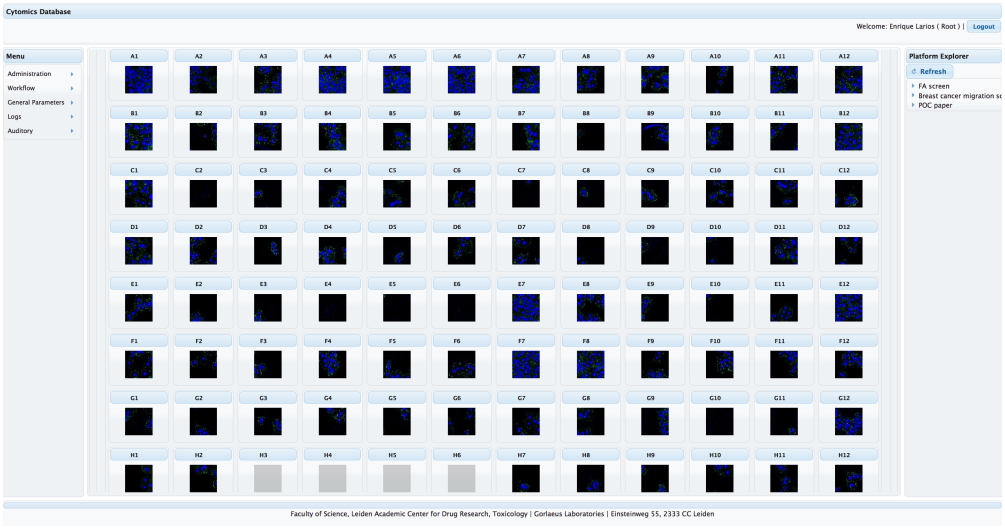
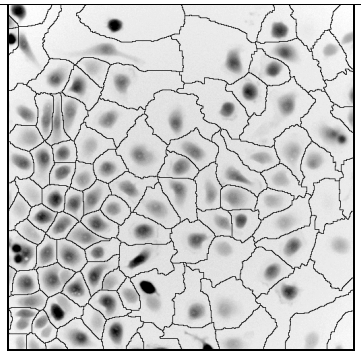


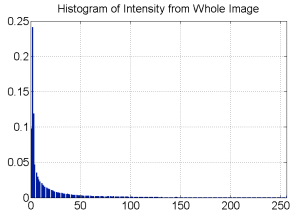
Figure 2.2: *Web plate layout design GUI*

2.2.2 Image Analysis Module

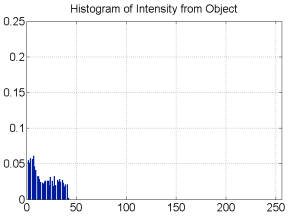
In the acquisition phase, the time-lapse sequences are connected to the plate design. Customized image processing and analysis tools or algorithms are applied on the raw images to obtain features for each of the different treatments. Our image analysis kernel is deployed to provide a customised and robust image segmentation and object tracking algorithm (YVDvdW09), dedicated to various types of cytometry. The current package covers solutions to cell migration, cellular matrix dynamics and structure dynamics analysis (see Figures 2.3, 2.4, 2.5). The package has been practised in HTS experiments for toxic compound screening of cancer metastasis (LYdB$^+$10) (QSY$^+$11), wound-and-recovery of kidney cells (QSY$^+$11) and cell matrix adhesion complex signaling, etc (CYW$^+$11).



(a) Image is divided into several coarse regions

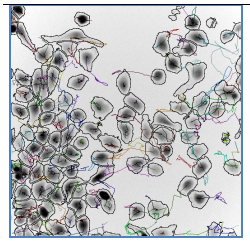


(b) The intensity histogram of the whole image

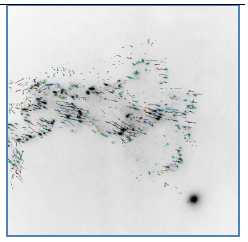


(c) The intensity histogram of one random coarse region

Figure 2.3: *Image and coarse regions* (CYW+11)



(a) Cell tracking results



(b) Adhesion tacking results

Figure 2.4: *Using our image analysis solution, the phenotypic measurements of (a) live cells and (b) adhesion can be extracted* (CYW+11)

The segmentation of objects is conducted using our watershed masked clustering algorithm, an innovative algorithm dedicated to fluorescence microscopy imaging. Frequently, the efficiency of fluorescence staining or protein fusion is subjective and highly unpredictable, which results in disorganized intensity bias within and between cells (see Figure 2.3(a)). The principle behind the algorithm is to divide such an extreme

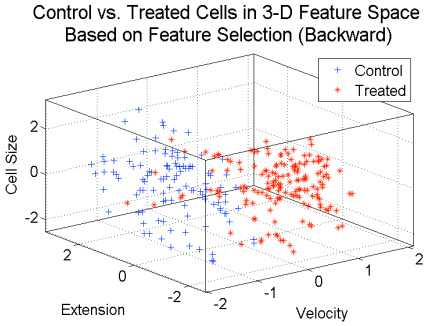


Figure 2.5: *Phenotypic characterization of the Epidermal Growth Factor (EGF) treatment using a highly aggressive cancer cell line, the illustrated features are picked by branch-and-bound feature selection. The EGF-treated cell group shows a significant increased migration velocity*
(CYW+11)

and multimodal optimization problem (Figure 2.3(b)) into several sub-optimal yet uni-modal optimization problems (Figure 2.3(c)). Such divided-and-conquer strategy provides an extended flexibility in searching intensity thresholds in each image. Contrary to bottom-up segmentation strategies such as the Otsu algorithm, our solution prevents undertraining by introducing a flexible kernel definition based on the congenital (intensity) homogeneity of an image. Unlike top-down segmentation strategies such as the level-set algorithm, our current algorithm prevents overtraining by a global overview of the intensity distribution of the region, therefore, it is less sensitive to local intensity distortion; in addition, our algorithm does not require any prior knowledge or manual interference during segmentation while it is mandatory for most existing top-down methods.

The tracking of objects is accomplished by customised algorithms deployed in the image analysis package. The principle behind this tracking algorithm is to estimate the minimum mean shift vector based on a given model (LYdB+10) (YVDvdW09).

With the binary masks and trajectories information obtained from image analysis, several phenotypic measurements are extracted for each object. Using the state-of-art pattern recognition and statistical analysis techniques, the effect of chemical compounds can be easily quantified and compared (Figure 2.5). Depending on the experiment setting, our package may employ up to 31 phenotypic measurements during the analysis.

The image analysis module is designed as a web service API module in the HTS platform. As the image analysis computation requires large image volumes to be processed, a high performance scientific cluster performs the image processing in order to obtain results in reasonable time.

2.2.3 Data Management Module

In cytomics bookkeeping of the information generated during lab experiments is crucial and the amount of image data can easily exceed the terabyte-scale. However, currently spreadsheets applications are commonly used for storing experiment data. The accessibility of large volume of image data already poses an obstacle in the current stage.

After scientists, having performed HTS experiments, it is necessary to store meta information, including the experiment type, the protocol followed, experimental conditions used and the plate design, and associate each well in the plate to the raw images generated during the experiments and the results obtained from the image and data analysis when these processes are completed.

Currently, the large volume of images are stored in a file server and they are accessed following a standard naming convention. The locations of the files are stored in the spreadsheet application used for the experiment, but this is not a practical solution for knowledge discovery later on or querying the results obtained in the analysis process.

The platform uses MonetDB, a modern column-based database management system (DBMS) as a repository of the experiments metadata that is used in the HTS Workflow System. Each component of the architecture communicate with the database through web services. This makes the future integration with other APIs more flexible.

2.2.4 Data and Pattern Analysis Module

Typical to the kind of analysis required for cytomics data is that the temporal as well as the spatial dimension is included in the analysis. The spatial dimension tells us where a cell or cell structure is, whereas the time-point informs us when it is at that particular location. Features are derived from the images that are time lapse series (2D+T or 3D+T). Over these features pattern recognition procedures are multi-parametric analysis. It is a basic form of machine learning solution, which is frequently employed in the decision-making procedure of biological and medical research. A certain pattern recognition procedure may be engaged in supporting various conclusions. For example, a clustering operation based on cell morphological measurements may provide an innate subpopulation within a cell culture (LYdB$^+$10) while a classification operation using temporal phenotypic profile can be used to identify of each cell phase during division (NWH$^+$10). The service that deals with the pattern recognition is based on the PR-Tools software package developed at the Delft University of Technology (`www.prtools.org`).

The PR-Tools library can be integrated in MatLab (MAT10) and we have used it in that fashion. In order to deal with the temporal dimension, the package was extended with specific elements to allow temporal analysis over spatial data (Yan13). The prototype data analysis module is implemented as a web service API based on output generated by MatLab deployment tools. The availability of MatLab with PR-

Tools within this architecture allows for rapid prototyping with a range of complex mathematical algorithms. In addition, PR-Tools in MatLab has its own GUI and in this manner data mining strategies can be explored by the end-user without in-depth knowledge of machine learning. The flexibility that is accomplished in this manner is efficient for the end-users as well as the software engineers who need to maintain and implement the services for machine learning.

2.3 System Architecture of the HTS Analysis Platform

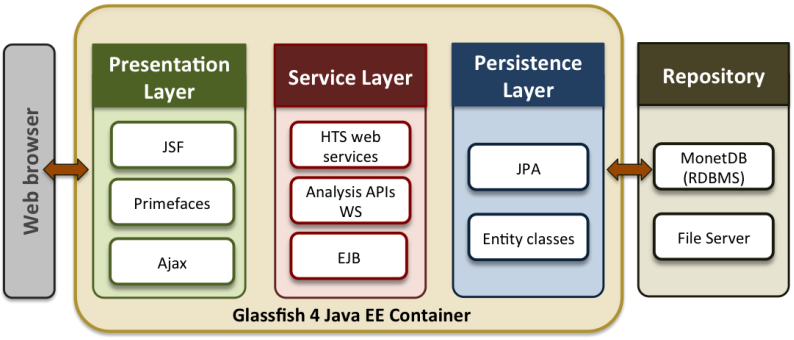


Figure 2.6: *The HTS analysis platform architecture*

To automate the workflow of HTS experiments and provide the users with a convenient interface to interact with the system, we have designed an *HTS analysis platform* (YLL+11) (for short: HTS platform), which has a layered architecture. Figure 2.6 depicts the components in each layer of the architecture.

2.3.1 The Presentation Layer

The HTS platform enables end-users to carry out complete HTS experiments using a single graphical user interface, i.e., the *HTS Analysis GUI*. This way, even for end-users without extensive knowledge in cytomics, it is easy to learn how to analyse HTS experiments in cytometry. In addition, data sets produced under different conditions or from different HTS experiments are available through one interface. This also counts for the resulting data from each step in an HTS experiment. As a result, end-users can easily view, compare and analyse the different data sets.

2.3.2 The Service Layer

The service layer, through web services, support every step in the HTS workflow that is done on the computers. The APIs are grouped in three modules in the *Web Service*

API layer, with each module corresponding to a module described in Section 2.2. This module structure allows quick development, error isolation and easy extending with more functional modules in the future.

We chose SOAP (Simple Object Access Protocol) messages for invoking the web services and receiving results, because of its approved interoperability in web applications and heterogeneous environments. In case of the HTS platform, because of the presence of legacy systems we must support different programming languages. Using SOAP makes it possible for various languages to invoke operations from each other. Transportation of the data generated by an experiment is integrated into web service calls. Large files are transmitted as attachments of the SOAP messages. To do this, the MTOM (Message Transmission Optimization Mechanism) feature (GMNR05) of the Glassfish Server is used. Ensuring error free data transmission and controlling user access permissions are done at the application level.

2.3.3 The Persistence Layer

The persistence layer is based on the principle of object-relational mapping (ORM) which involves delegating access to a relational database and, which in turn gives an object-oriented view of the relational data, and vice versa (O'N). The Java Persistence API (JPA) framework has been implemented in this layer to keep a bidirectional correspondence between the database and objects. Those Java objects used in this framework are known as Java Entities (KS06). The entities are objects that live shortly in memory but persistently in the database. Besides that, they have all the features of a Java class like instantiation, abstraction, inheritance, relationships and so on. The entities used in CytomicsDB follow the same structure as the tables they map to. CRUD operations are registered as named query methods which are written in Java Persistence Query Language (JPQL). These customized queries can be attached to entities as native queries via JPA.

2.3.4 The Repository Layer

There are two components in this layer. For the data management component, we made a conscious choice for MonetDB (`www.monetdb.org`), a modern column-based database system, after having considered different alternatives. For instance, in the initial design of the database schema, we have considered to use an XML supporting DBMS such as Oracle or Microsoft SQL Server in order to facilitate a flexible integration with other systems in the future. However, it is generally known that, compared with relational data, XML data requires considerable storage overhead and processing time, which makes it unsuitable as a storage format for the large volume of cytomic data. Moreover, traditional database systems are optimised for transactional queries, while in cytomics, we mainly have analytical queries. Traditional database systems generally carry too much overhead when processing analytical queries (Bon02). What we need is

a database optimised for data mining applications. MonetDB is a leading open source database system that has been designed specially for such applications (Bon02). It has been well-known for its performance in processing analytical queries on large scale data (BMK09). Thus, in our final decision, we use SOAP messages (i.e., XML format) to exchange small sized (meta-)data, but use MonetDB to store a major portion of the data produced and used during the HTS experiments, including all metadata generated during analysis. Additionally, a powerful scientific computer cluster is used to execute computing intensive image analysis tools. The future plan is to move also the raw data and as much as possible operations on them into the database system.

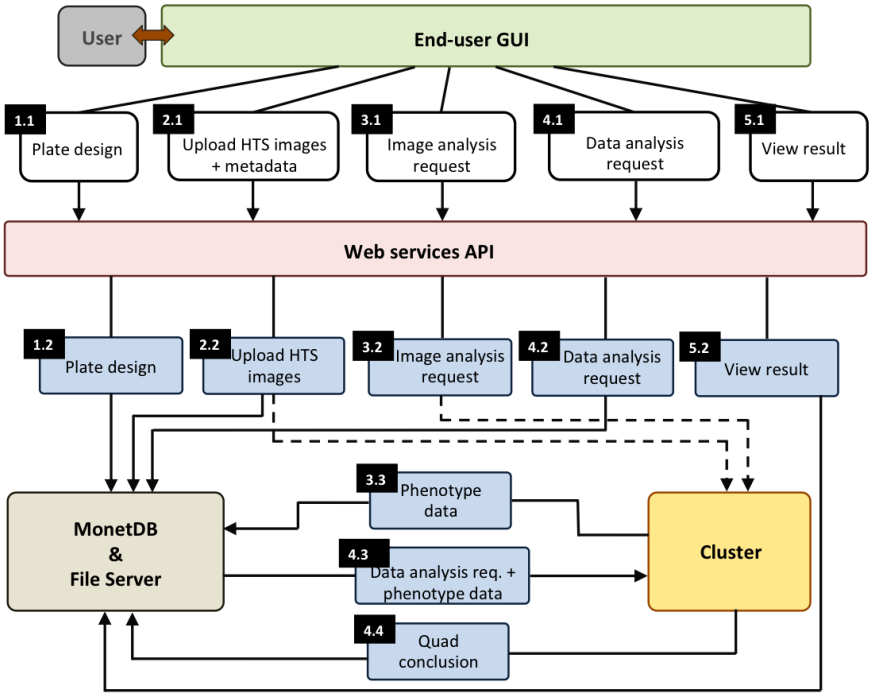


Figure 2.7: *Flow of control of the HTS platform*

2.3.5 Flow of Control

The diagram shown in Figure 2.7 illustrates the flows of the control in the HTS platform. How the main features of the platform are executed is shown by five sequences of annotated arrows starting from the end-user GUI. Arrows handling the same operation are grouped together by a major number, while the minor numbers corresponds to the order of a particular step that is called in its containing sequence. Below we describe each sequence.

Sequence 1 handles a new plate design, which is straightforward: the request is sent to MonetDB and a new entry is created. Sequence 2 handles uploading an HTS image request. Because currently the raw image data is stored separately, this request results in the metadata being stored in MonetDB while the binary data is stored on the file server. Sequence 3 handles an image analysis request, which is passed to the scientific super computer, since the tools for the analysis stage are there. Then the results are sent to MonetDB and stored there (step 3.3). Sequence 4 handles a data analysis request, which is first sent to MonetDB. Then, MonetDB passes both the request and the necessary data (obtained from the image analysis) to the scientific super computer for execution. The results are again stored in MonetDB. Since the most used data is stored at one place, in sequence 5, a view results request can be handled by just requesting data of both image analysis and data analysis from MonetDB. In the GUI, the results are displayed with the corresponding plate layout, as indicated in Figure 2.2.

Summary. In this section, we described the software architecture of the HTS platform, how its main features are processed, and how web services are used for the communication with the DBMS and the dedicated scientific computer cloud. In the next section, we present how all data is stored in the DBMS.

2.4 Database Design

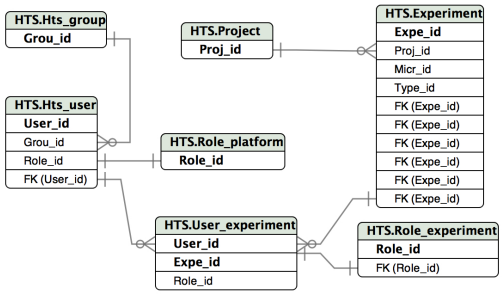


Figure 2.8: *Database schema for Project Metadata*

The complete relational database schema designed to store the metadata, location of images and binary data generated during the execution of the HTS workflow is shown in Figure 2.8, Figure 2.9, Figure 2.10, and Figure 2.11. The database schema can be roughly divided into five views: i) users and the experiment sets they work on (c.f. 2.8, c.f. 2.9), ii) the design of the culture plates (c.f. 2.10), iii) raw images acquired during a single HTS experiment (c.f. 2.11), iv) results of image analysis (c.f. 2.11), and v) results of data analysis (c.f. 2.11). In order to simplify the views, the tables show only the primary and foreign keys. Below we explain how the data is stored in each of

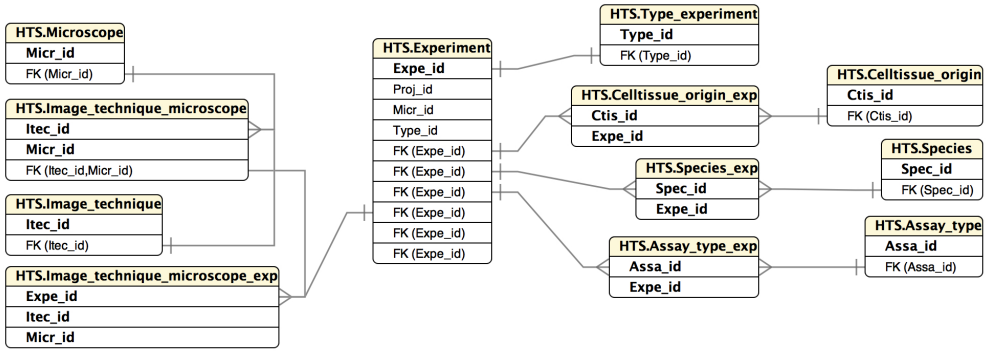


Figure 2.9: *Database schema for Experiment Metadata*

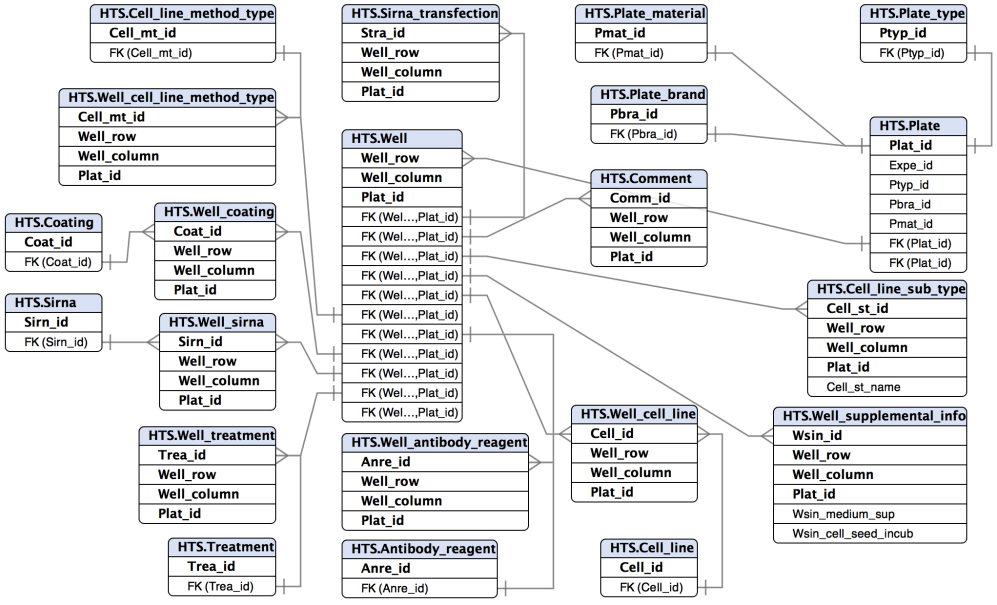


Figure 2.10: *Database schema for Plate-Well Metadata*

these views and the relationships among the tables.

2.4.1 Users and Experiment Sets

The basic information of a user is stored in the table `hts_user`. A user belongs to a `hts_group` (research group) and has a special privilege for accessing the platform according to `role_platform`, possible values are: system administrator, administrator and regular user. Additionally, every user can start with a new `Experiment`, and is

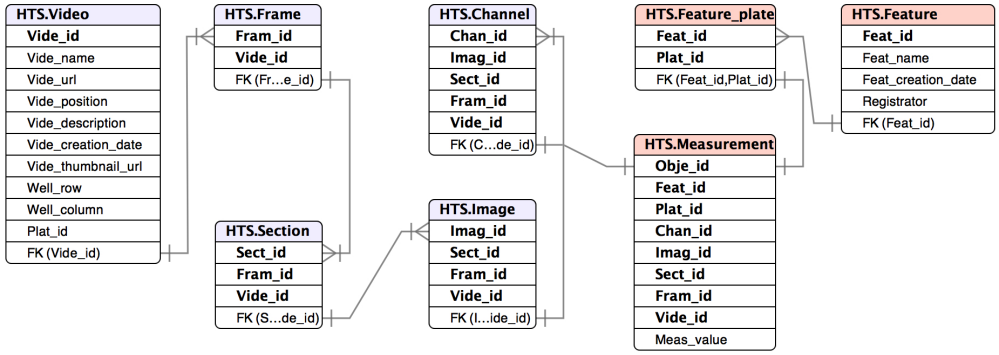


Figure 2.11: *Database schema for Raw Images and Measurement Metadata*

then also the author of this set of experiments.

Multiple users may work on the same experiment set, but only the author of an experiment set can grant another user the access to this set. The table `User_experiment` stores the data required for validating the access control of all users. Possible values of `Role_experiment` include: author, expert user, analyst user and guest user.

2.4.2 Plates and Wells

An HTS experiment starts with the design of the layout of a culture `Plate` of N × M `Wells` in which the cells are kept and cultured. An experiment set can contain multiple culture plates, which typically have sizes of (but not restricted to) 4 × 6, 6 × 8, 8 × 12 or 16 × 24 wells. A user can create `conditions` e.g. cell_lines, compounds, siRNA, coating, etc. to be applied to the wells. Similar to the `Experiment` table, restricted access to the `conditions` are denoted explicitly according to the privilege granted to the user in table `Role_platform`. Table `Well` keeps track of which conditions are used by them in each experiment. Thus, one condition can be used in multiple experiment sets and accessed by multiple users. However, by referring to the compound primary key `(User_id, Expe_id)` of `User_experiment`, a user is restricted to only have access to a condition, if he/she has access to an experiment set using the condition. Additionally, because conditions are applied on individual wells, the table `Well_(condition)` is designed to store this information.

2.4.3 Raw Images

A third step in the HTS workflow (the "HTS" step in Figure 2.1) is to process the cultured plates using automated microscopy imaging system. The response of the cells is recorded through time-lapse microscopy imaging and the resulting image sequences are the basis for the image analysis. The structure of an image file depends on the type

of experiment (denoted by `Type_id` in `Experiment`) and the microscopy used in the experiment. Currently, four types of structures are supported:

1. 2D (XY): this structure corresponds to one frame containing one image which is composed of multiple channels ([1]Frame → [1]Image → [1..n]Channels).

2. 2D+T (XY+T): this structure corresponds to one video with multiple frames. Each frame contains one image composed of multiple channels ([1]Video → [1..n]Frame → [1]Image → [1..n]Channels).

3. 3D (XYZ): this structure corresponds to one frame with multiple sections. Each section contains one image composed of multiple channels ([1]Frame → [1..n]Sections → [1]Image → [1..n]Channels).

4. 3D+T (XYZ+T): this structure corresponds to one video with multiple frames. Each frame can have multiple sections and each section contains one image composed of multiple channels ([1]Video → [1..n]Frame → [1..n]Sections → [1]Image → [1..n]Channels).

These four structures can be represented by the most general one, i.e., 3D+T. The 2D structure can be seen as a video of one frame containing one section. Each frame in the 2D+T structure can be regarded to contain one section. Finally, the 3D structure can be seen as a video of one frame. In the database schema, the generalised structures are captured by five relations, i.e., `Video`, `Frame`, `Section`, `Image` and `Channel`, connected to each other using foreign keys. Information stored in these relations is similar, namely a name and a description. Only the main table `Video` contains some extra information, e.g., a foreign key referring to the table `Well` to denote from which well the image has been acquired. Because currently only the metadata of the raw images are stored in these tables, the location of the image binary data is stored in `Vide_url`. The exact type of the video structure can be looked up using `Type_id` in the `Experiment`.

2.4.4 Results of Image Analysis

The results of image analysis are auxiliary images which, currently, are binary masks or trajectories. These images are result of the execution of quality enhancing filters and segmentation algorithms employed to extract region of interests (ROIs). The metadata of these images is stored in the table `Measurement`, including the location where the binary data is stored. Moreover, this table also store the phenotypic measurements gathered from ROIs and location of auxiliary images e.g. trajectories. The foreign key `Vide_id` links a measurement record to the raw video image file, on which the image analysis has been applied.

2.4.5 Results of Data Analysis

The goal of the data analysis stage is converting image data into comprehensive conclusions. For achieving this goal basic operations such as feature selection, clustering and classification are applied to the measurements extracted from the image analysis. The parameters used by the operation and the extracted features are respectively stored in `Feature`, and are connected to the corresponding `Measurement` record via foreign keys.

2.5 Related Work

Data management in microscopy and cytometry has been acknowledged as an important issue. Systems have been developed to manage these resources, to this respect the Open Microscopy Environment (`www.openmicroscopy.org`) and the OMERO platform is a good example. Another approach is connecting all kinds of imaging data and creating a kind of virtual microscope; such has been elaborated in the Cyttron project (KBK+09) (`www.cyttron.org`). The connection is realized by the use of ontologies. Both projects strive at adding value to the data and allow to process the data with plug-in like packages. These approaches are very suitable for the usage of web services. Both projects are also very generic in their architecture and not particularly fit for HTS and the volume of data that is produced. Important for data management in cytometry is that both metadata and bulk data are accommodated well. The accumulation of metadata is crucial; successful accommodation of both metadata and bulk data has been applied in the field of microarrays (SMS+02). Here, the interplay of the vendor of scanning equipment with the world of researchers in the life-sciences has delivered a standard that is proving its use in research. One cannot, one to one, copy the data model that has been applied in the field of microarrays. Like in cytometry, for microarrays the starting point is images in multiple channels. However, for cytometry, location and time components are features that are derived from the images whereas in microarrays the images are static from a template that is provided by the manufacturer. In cytometry there is a large volume of data that needs to be processed but this volume is determined by the experiment and it can be different each time; i.e. it depends very much from the experimental setup. This requires a very flexible approach to the model of the data. An important requirement for the metadata is that they can be used to link to other datasets. The use of curated concepts for annotation is part of the MAGE concept and is also embedded in the CytomicsDB project. We have successfully applied such approach for the zebrafish in which the precise annotations in the metadata were used to link out to other databases (BV08) and similarly, as mentioned, in the Cyttron projects the annotations are used to make direct connections within the data (KBK+09). For cytometry data linking to other data is important in terms of interoperability so that other datasets, i.e. images, can be directly involved in an analysis. For cytometry, there

are processing environments that are very much geared towards the volume of data that is commonly processed in HTS. The Konstanz Information Miner (KNIME) is a good example of such environment. It offers good functionality to process the data but it does not directly map to the workflow that is common in HTS and it does not support elaborate image analysis. Therefore, in order to be flexible, the workflow is directed towards standard packages for data processing and the processes are separated in different services rather than one service dealing with all processing. So, one service specifically for the image processing and analysis (e.g. ImageJ or DIPLIB) and another service for the pattern recognition and machine learning (e.g. WEKA or PRTools). In this manner flexibility is accomplished on the services that one can use.

2.6 Conclusions and Future work

In this chapter we presented the design of a platform for high content data analysis in the High-Throughput Screen cytomic experiments that seamlessly connects with the workflow of the biologists and for which all processes are automated. Based on the beta testing, this system increases the efficiency of post-experiment analysis by 400%. That is, by using this the framework, it now takes less than a week to accomplish the data analysis that previously easily took more than a month with commercial software, or a year by manual observation. Comparing with solutions such as CellProfiler (CJL+06) or ImagePro, our solution provides a unique and dedicated approach for HTS image analysis. It allows end-users to perform high-profile cytomics with a minimum level of a prior experience on image analysis and machine learning. The system is modular and all modules are implemented in the form of web services, therefore, updating the system is virtually instantaneous. Moreover, the framework is very flexible as it allows connecting other web services. Consequently, a fast response to new progress in image and data analysis algorithms can be realized. Further integration with online bio-ontology databases and open gene-banks is considered so as to allow integration of the data with other resources. Therefore, the platform can eventually evolve into a sophisticated interdisciplinary platform for cytomics. Having the screen information comprehensively organized in a sophisticated and scalable database is a fertile ground for knowledge discovery.