Author: Larios Vargas, Enrique
Title: Design and development of a comprehensive data management platform for cytomics : cytomicsDB
Issue Date: 2015-11-25

Design and development of a comprehensive data management platform for Cytomics

~ **CytomicsDB** ~

Enrique Larios Vargas

The research developed in this thesis was performed at the Imaging and BioInformatics group (Leiden Institute of Advanced Computer Science), the Division of Toxicology (Leiden Academic Center for Drug Research) and with the support of the Database Architecture Group (Centrum Wiskunde & Informatica - Amsterdam), the Netherlands.

This research project has been supported by the BAPE program, EMECW Lot 18, Erasmus Mundus.

Design and development of a comprehensive data management platform for Cytomics: CytomicsDB

Cover design by: Terry Zabel

ISBN: 978-94-6259-950-5

Printed by Ipskamp, www.ipskampdrukkers.nl

Design and development of a comprehensive data management platform for Cytomics: CytomicsDB

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden
op gezag van Rector Magnificus prof. mr. C.J.J.M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op woensdag 25 November 2015
klokke 10.00 uur

door

Enrique Larios Vargas

geboren te Lima, Peru
in 1977

Samenstelling promotiecommissie:

promotor:	prof. dr. J. N. Kok	LIACS, Universiteit Leiden
co-promotor:	dr. Ir. F. J. Verbeek	LIACS, Universiteit Leiden
overige leden:	prof. dr. Lucie Kubínová	Czech Academy of Sciences, Prague, Czech Republic
	dr. Ying Zhang	CWI, Amsterdam
	prof. dr. Bob van der Water	LACDR, Universiteit Leiden
	prof. dr. Stefan Manegold	LIACS, Universiteit Leiden

I dedicate this thesis to my mother Luveslinda, for her sacrifices and her effort for giving me the best education and also to my five angels on heaven: Padre Vicente, Madre Teodora, my cousin Ivan, my father Paco and my grandmother Rodolfa.

Table of Contents

Page

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter the thesis scope as well as the current key concepts related to this dissertation are introduced. In addition, the thesis structure is outlined.

1.1 Thesis scope

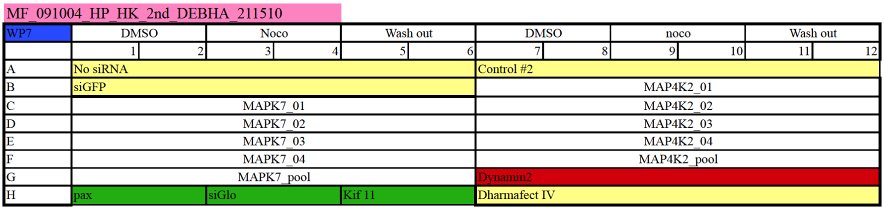
One of the main challenges in software engineering is to develop systems capable to adapt to the continuos changes in their environment. This problem is present in software systems for biomedical research in cell systems. In biomedical research, the study of cellular systems on large scale is called Cytomics, High-Throughput screening (HTS) is one of the most common techniques in Cytomics for target validation. HTS has to face that challenge due to the diversity and the emergence of new technologies in the context of data, roles, hardware, software tools, techniques, algorithms, protocols, etc. Additionally, the extremely large and growing volumes of data produced in these experiments complicate the exploration, interoperability, understandability, sharing and reusability of the data. Therefore, there is an urgent need to design and develop a comprehensive data management platform for experiments in Cytomics. This thesis focuses on the study of how to organize the data managed in *Cytomics* in an optimal and yet flexible manner, so that these challenges can be tackled while taking care of an appropriate organization of the heterogeneous data and then providing the necessary tools and software to facilitate the extraction of meaningful data from these large repositories.

1.2 HTS experiments workflow

In drug discovery, an HTS workflow is a sequence of operations and activities required to identify starting points for drug design or also called hits. These sequence consist of five stages: (YLL^{+}11): (1) *Experiment Design*, (2) *Wet-lab*, (3) *Image acquisition*, (4) *Image analysis* and (5) *Data analysis*.

1.2.1 Experiment Design

The key component in HTS is the "plate"; this is basically a small container that features a grid of wells and it is considered the principal information carrier of the experiment (c.f. 1.2). A spreadsheet application is used to create a layout of the plate in order to store practical information about the treatment introduced for each well in the cell culture plate. Figure 1.1 displays the design of a 96 well plate; details of the metadata are linked to each well therefore this is also called "plate design". The "plate design" contains the input data for the plate e.g. metadata and allows to link the output of the plate e.g. image files. Figure 1.1 shows a brief description of the plate design. Wells are identified by a row from "A" to "H" and a column from "1" to "12". A colour-coded notation in the plate design is used which facilitates the readability of the basic metadata linked to the experiment. For this experiment the MCF7 cell line is used and per well specific genes are switched off; i.e. knocked down, by the use of siRNA transfection which are represented by the not colour-coded wells. The negative controls are shown in light yellow, the positive control for transfection is in green, and the positive control for knockdown and the noco-assay is in red e.g. Dynamin2. The cells from wells B1 to B6 have been transfected with siGFP (light yellow). Wells B1 and B2 were additionally treated with DMSO while wells B3 and B4 have been treated with Nocodazole.



MF_091004_HP_HK_2nd_DEBHA_211510

WP7	DMSO		Noco		Wash out		DMSO		noco		Wash out	
	1	2	3	4	5	6	7	8	9	10	11	12
A	No siRNA						Control #2					
B	siGFP						MAP4K2_01					
C	MAPK7_01						MAP4K2_02					
D	MAPK7_02						MAP4K2_03					
E	MAPK7_03						MAP4K2_04					
F	MAPK7_04						MAP4K2_pool					
G	MAPK7_pool						Dynamin2					
H	pax		siGlo		Kif 11		Dharmafect IV					

Figure 1.1: *Design of a 96 wells plate for an experiment using MCF7 wt cells*

1.2.2 Wet-lab

Based on the design stage, one or more plates are used during the experiment. During the experiment, the *in vitro* cell lines are prepared and put into a culture well of the plate (c.f. 1.2). Plates for HTS can have either 24, 96, 384 wells, etc. i.e. multiples of 24. These wells will contain, depending on the type of experiment, culture medium and depending on the layout/design, some chemical compounds. These wells will also contain cell populations, and designed treatments are induced into them (YLL$^+$11). Upon completion of the wet-lab procedure, the plate is prepared for the further readout during the image acquisition stage.

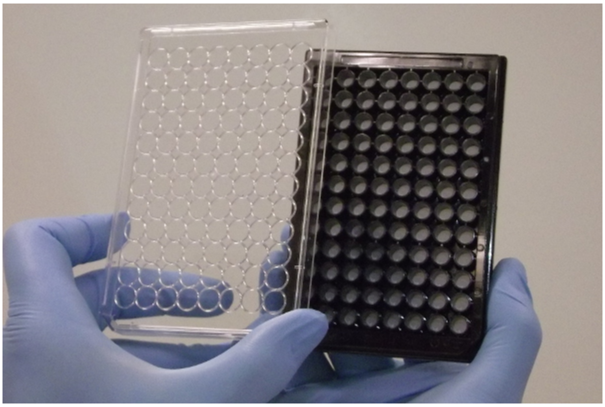


Figure 1.2: *96 wells plate for HTS.*
(Alv)

1.2.3 Image acquisition

In biomedical laboratories images are usually acquired by measuring photon flux in parallel, using a camera, or sequentially, using a point detector and equipment that scans the area of interest. However, capturing an image from a camera into the computer is straightforward, in most cases image acquisition needs to be tightly synchronized with other computer-controllable equipment such as shutters, filter wheels, x-y stages, z-axis focus drives and autofocus mechanisms; the controls for this are implemented in software and hardware. This automation is necessary to gather desired multiparametric information from a sample to allow the unattended acquisition of large numbers of images in time-lapse series, z-stacks, multiple spatial locations in a large sample or multiple samples in a large-scale experiment. Dedicated image acquisition software is therefore indispensable to communicate with these various components and coordinate their actions such that the hardware functions as quick and flawless as possible as well as allowing the researcher to easily execute the following sequence of acquisition events (EBG$^+$12), i.e.:

1. Manual or automated acquisition.

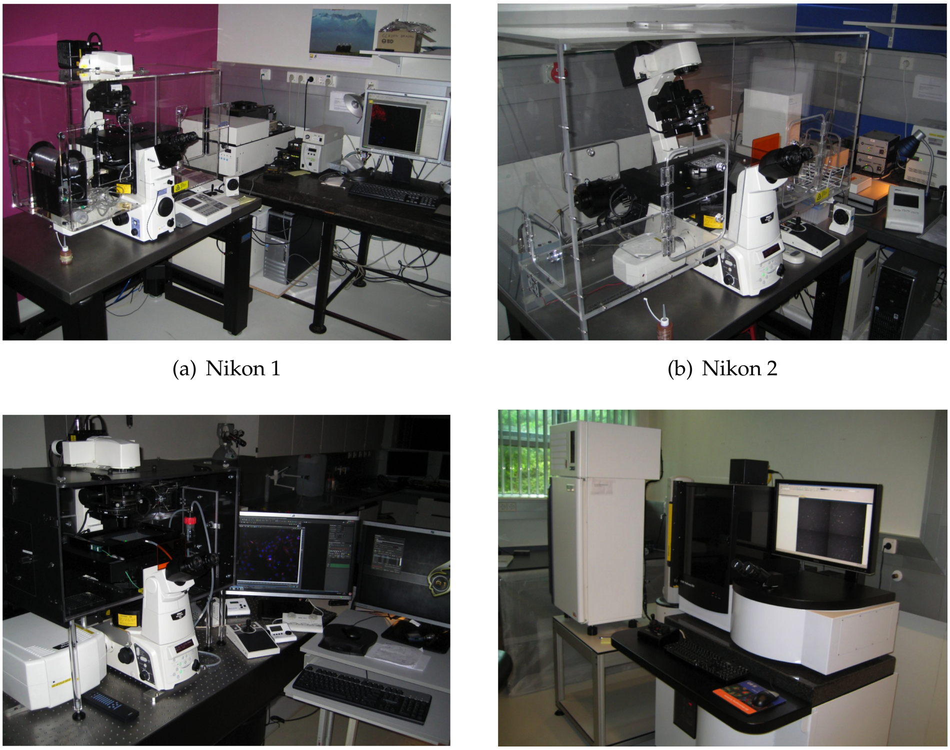
2. Single time point or time series.

3. Single focal plane or three-dimensional stack.

4. Single channel, multiple channels or hyperspectral.

5. Acquisition protocol is predefined or determined on the fly based on analysis of image data.

Figure 1.3 shows four different microscopes commonly used in Hight-Throughput Screening.



(a) Nikon 1

(b) Nikon 2

(c) Nikon 3

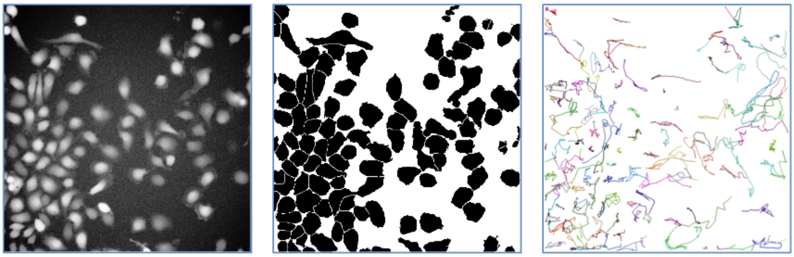
(d) BD Pathway

Figure 1.3: *Microscopes for HTS.*
(Tox)

1.2.4 Image analysis

Eliceiri et. al. (EBG+12) highlight the fact that biologists are increasingly interested in using image analysis to convert microscopy images into quantitative data (GM07) (LC09). Image analysis, in particular, is a necessary step for experiments in which hundreds or thousands of images are collected by automated microscopy, whether for screening multiple samples, collecting time or z-series data, or other technologies that generate vast volumes of image data. In addition to image analysis in a high-throughput context, image analysis is important for many biological studies: for example, quantifying the amount and localization of a signaling protein, measuring changes in structures over time, tracking invading cancer cells or looking at nonspatial data such as fluorescence-lifetime data (Lak99). Image analysis facilitates data reduction and can help to ensure that results are accurate, objective and reproducible.

For instance, during the image analysis process, raw images from the image acquisition (c.f. 1.4(a)) stage are converted to auxiliary image results (c.f. 1.4(b) and 1.4(c)). Quality enhancing filters and segmentation algorithms are executed to extract region of interests (ROIs). Moreover, other types of processing based on ROIs such as object tracking are also applied. For each image sequence, phenotypic measurements are gathered from ROIs and trajectories. Trajectories are not available for static images (YLL+11).



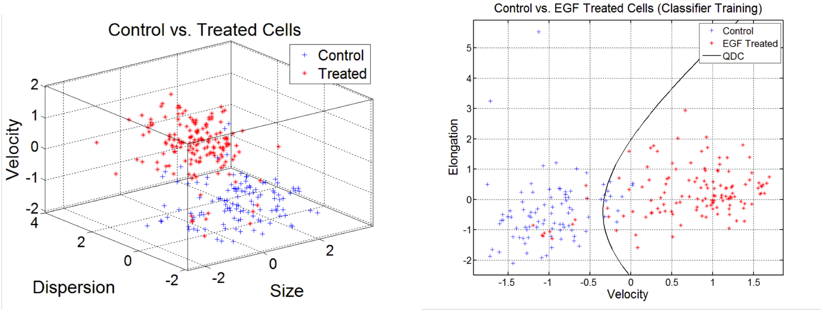
(a) Original image (b) Binary masks of (a) (c) Trajectories of (a)

Figure 1.4: *Segmentation and subsequent image analysis* (YLL+11)

1.2.5 Data analysis

Eliceiri et. al. (EBG+12) conclude that the interpretation of the results from a high-content imaging application requires proper presentation and visualization of the image data. Many data display options are currently available according to the type of microscope used. Image analysis results can be displayed graphically as heat maps, line graphs, bar charts, dose response curves, or scatter plots e.g. (c.f. 1.5(a)) and 1.5(b)). In addition, to interpret data within a plate or set of plates at a glance, thumbnail images of the entire plate can also be displayed.

The purpose of data analysis varies from experiment to experiment. One basic purpose is to identify "hits", meaning to identify genes or compounds which play a functional role in the cellular process that is under investigation (genes), or may positively or negatively affect the process (compounds). This is determined by comparing each sample with control treatments that are carried out under the same condition but induce no change in the celullar process. Before "hits" are identified, quality control and data normalization are performed to remove systematic errors and to allow comparison and combination of samples from different plates (Di13).



(a) Cell migration (b) Structure dynamics

Figure 1.5: *Data analysis*
(YLL^{+}11)

1.3 Data management in Cytomics

Finding "hits" contributes to identify targets for drugs. *Drug discovery* has a fundamental intrinsic dedication in increasing the number of novel medicines. However, in order to accomplish this goal, high costs are involved from discovery through clinical trials to approval (SKE07). Thus, the requirements for a more sophisticated IT infrastructure used to assist the drug development process has been exponentially increased during the last decade. *Cytomics* studies introduce information related to the behaviour of the cell systems, exploiting this information is a key factor for the drug development process. High-Throughput Screening (HTS) is one of the most common techniques used in *Cytomics*. In HTS the image dataset size per plate is approximately between 20Gb to 40Gb and depending on the type of experiment performed multiple plates can be used in one experiment and the number of images generated can reach around 100000 images per plate. The large volume of data generated in these experiments make its management a challenge and a necessity.

The current data management challenges in *Cytomics* can be summarized as following categories: (1) *Heterogeneity of the data*, (2) *Data exploration*, (3) *Interoperability*, (4) *Data sharing and reusability*, (5) *Understandability* and (6) *Transport of data*. In the next paragraphs we will discuss the categories.

1.3.1 Heterogeneity of data

In HTS experiments the diversity of data is one of the main issues to be tackled for the cytomics-based systems. The variety in the data is a consequence of the diversity of harware and software components involved in HTS (c.f.1.6). A summary of these components are:

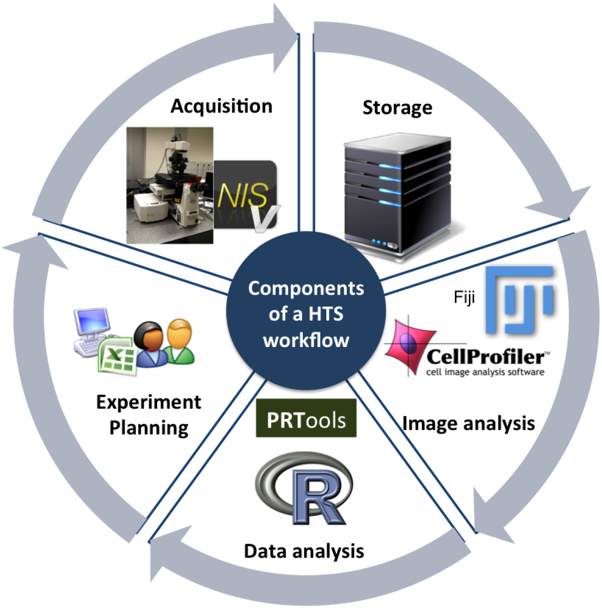


Figure 1.6: *Hardware and Software components in an HTS workflow*

Types of microscope. In modern biology, one of the most basic tools is the visual inspection of cells using a microscope and in modern optical microscopy for HTS is commonly used the following types of microscope: (1) Confocal microscopes, (2) Multiphoton microscopes (3) Multispectral microscopes and (4) Fluorescence microscopes. Each one managing different type of variables and parameters.

Image formats. The different types of microscopes have different file formats associated with them. This means that metadata is sometimes only available in propietary format. Consequently the image files generated will use a propietary format as well. However, these propietary software also includes plugins or tools for exporting or visualizing the images using more standarized format conventions e.g. tiff files.

Image and data analysis tools. There is a large number of open source tools and inhouse applications developed for image and data analysis in HTS. Depending on the programming language for the development and operating systems supported, it is necessary to create additional scripts or import data for further processing according to the file types supported by those tools. The output can also vary according to the analysis performed, for instance plain text files or binary data.

Spreadsheet applications. Spreadsheet applications are still very commonly used by biologist in the HTS workflow for bookkeping metadata related to the design of the experiment. The use of this type of applications have many drawbacks such as: susceptible to errors, difficult to maintain, limited security and accesibility, not shareable, and not suitable for querying.

Legacy systems. In the research groups, there are usually other systems which play a role in the management of laboratory data or other repositories which store logistic information of chemicals, compounds, or other components required in the lab. These systems work most of the time independent of the HTS workflow but the information contained needs to be synchronized with the experiment pipeline.

1.3.2 Data discovery

The drug discovery and development process depend on informed decision making. A Cytomics platform needs to increase the quality and pertinence of information for that decision-making process (SKE07). Therefore, identifying the location of meaningful data in the large datasets generated in HTS experiments is the most important challenge.

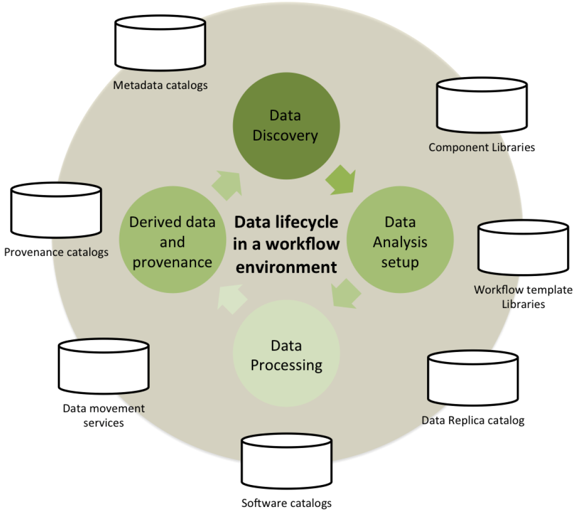


Figure 1.7: *Data Lifecycle in a workflow*
(DC08)

E-Science represents the increasing global collaborations of people and shared resources that will be needed to solve the new problems in science and engineering. These e-Science problems range from the simulation of whole engineering or biological systems, to research in bioinformatics, proteomics and pharmacogenetics (HT03). Hey et. al. address the imminent flood of scientific data expected from the next generation of experiments and the critical needs for analytical tools capable of exploiting and automating the process of going from raw data to information to knowledge.

Deelman et. al. (DC08) present from the point of view of data, the lifecycle in a workflow for e-Science expressed in four transformations. This includes the following transformations (c.f. 1.7):

1. Data discovery.
2. Setting up the data processing pipeline.
3. Generation of derived data.
4. Archiving of derived data and its provenance.

Data analysis is often a collaborative process or is conducted within the context of a scientific collaboration. Data discovery is highlighted due to the fact that scientists in a collaboration frequently submit workflows to process datasets and derive scientific knowledge. These collaborators may submit related workflows and build upon earlier work by other scientists. Thus, scientists need to be able to discover information about workflows that have been executed in the past, identify datasets of interest, and locate analysis code and workflow templates.

1.3.3 Interoperability

Interoperability represents the posibility to exchange information between repositories. Additionally, interoperability is a domain specific concept and the heterogeneity of ICT implementations is such that there exist different solution spaces depending on the combination of existing systems and in many cases such solutions are not directly transferable to other cases (Kos06). According to Chen and Doumeingts (CDV08), interoperability has the meaning of coexistence, and autonomy. If systems are interoperable, this mean that their components are connected by a communication network and can interact; they can exchange services while continuing locally their own logics of operation.

Each of the various types of data handled in the drug discovery process poses its own specific problems for integration into the information systems and decision-making processes (SKE07). The role of standards is thereby becoming crucial. Standarization processes should be present in each stage of the scientific data lifecycle.

The development and wide adoption of common standards is extremely difficult despite the fact that they represent the most effective tool to deal with interoperability. In HTS, one of the main challenges is the standarization of the metadata i.e. naming conventions in order to facilitate the validation, verification, exchange with external platforms and the exploration of annotated information. Moreover, it is necessary to create standards to represent scientific data such as data models and standards for query languages.

Developing comprehensive approaches is complex because data interoperability is a problem that goes beyond technical aspects. Data interoperability approaches, in order to be complete, should reconcile all the differences arising between data providers and data consumers with respect to organizational, semantic, and technical characteristics governing their "exchange" of data. The majority of existing solutions focus on technical aspects only with very limited efforts and guidelines being oriented to reconcile differences at the organizational level, probably because this domain presents more complexities than others do (PCC13).

1.3.4 Understandability

In e-Science, the capability of the user to understand the information or knowledge contained in the large repositories generated by each experiment is becoming a critical issue. An HTS dataset is very complex. Therefore there is a huge challenge in organizing all this information so that it remains interpretable. With respect to the mass of data produced, we do not believe that a solution is solely linking together databases containing all the screening, laboratory testing, and chemical information. The scientists whose collaborative work build the "road to the lead" need to efficiently communicate. Only clear-cut analyses at each stage can optimize the process and facilitate such communication (Hey02).

The approach proposed by Heyse et. al. (Hey02) consist of: (1) perform thorough quality control to include only reliable data in further processing stages, (2) standardize, process, condense, and annotate data at each stage as much as needed by the respective experts involved, (3) keep data related to its context using metadata on assays, analyses, and compounds, and (4) provide tools to analyze and interpret large datasets following statistical and biophysical models and standard processes. This makes the overall screening process transparent and traceable.

1.3.5 Data sharing and reusability

Demchenko et. al. (DGDLM13) emphasize that the emergence of computer aided research methods is transforming the way how research is done and scientific data are used. Four types of scientific data are defined:

- Raw data collected from observations and from experiments according to an initial research method.

- Structured data and datasets that have gone through data filtering and processing.

- Published data that support one or another scientific hypothesis, research result or statement.

- Data linked to publications to support the wide research consolidation, integration and openness.

Collaboration is possible thanks to the contribution of new results as a consequence of the validation performed by scientists after scientific data has been published. Collecting information about the processes involved in the transformation of raw data to published data becomes an extremely important aspect of scientific data management.

There is a close link between interoperability and collaboration. The interoperability achieved by the use of standards for the metadata in HTS experiments, facilitates the understandability, discoverability and thus promotes the collaboration among scientists. Reseachers will be able to improve their contributions thanks to the use of domain-specific metadata information standards.

Reusability of published data within the scientific community is still another aspect to take into consideration. Understandability of the semantics of published data is a key factor for reusability and this has been done manually. However, given the volume of the data, this becomes, so far, impractical in the scale of Big Data science.

1.3.6 Transport of data

Kaisler et. al. (KAEM13) describe the storage and transport issues for big data. Current disk technology limits are about 4 terabytes per disk. So, 1 exabyte would require 25,000 disks. Even if an exabyte of data could be processed on a single computer system, the current systems do not allow to directly attach the requisite number of disks. Access to those data would overwhelm current communication networks. Assuming that a 1 gigabyte per second network has an effective sustainable transfer rate of 80%, the sustainable bandwidth is about 100 megabytes. Thus, transferring an exabyte would take about 116 days, if we assume that a sustained transfer could be maintained. It would take longer to transmit the data from a collection or storage point to a processing point than it would to actually process it.

Two solutions manifest themselves. First, process the data “in place” and transmit only the resulting information. In other words, “bring the code to the data", vs. the traditional method of “bring the data to the code." Second, perform triage on the data and transmit only that data which is critical to the downstream analysis. In either case, integrity and provenance metadata should be transmitted along with the actual data, this will ensure traceability and validation of the data before the processing stage.

Cytomics based platforms must take special care to the six categories of the current data management challenges in Cytomics in order to facilitate the interaction of researchers with their data. This means that platforms should ensure that research data is FAIR (Findable, Accessible, Interoperable and Reusable).

1.4 LIM Systems

A typical approach to the management of large volume of data in a laboratory is a Laboratory Information Management System (LIMS). LIMS is a computer application designed for the analytical laboratory that is designed to administer samples, acquire and manipulate data, and report results via a database. It automates the process of sampling, analysis and reporting (Mah91) (DAJ12).

A more complete definition is provided by Skobelev et. al. (SZK$^+$11), Laboratory information management systems belong to the class of application software intended for storage and management of information obtained in the course of the work of the laboratory. The systems are used to control and manage samples, standards, test results, reports, laboratory staff, instruments, and work flow automation. Integration of laboratory information management systems with the enterprise's information systems will make it possible to promptly transmit required data to the laboratory and the enterprise administration.

High-throughput screening (HTS) due to the overwhelming amounts of data is pushing forward the need for sophisticated IT infrastructure in order to enhance the laboratory performance. LIM systems have been designed as a first step to face this dilemma. However, the continuous progress in tools, hardware, and heterogeneity of the components involved in HTS make still difficult to completely tackle this problem.

Under the term Laboratory Information Management System, industry-related IT solutions for research, development, service, and production in the fields of chemistry, biology, environmental protection, or medicine are combined. The interdisciplinary object of the life sciences meanwhile characterizes a broad, large class of laboratory information management systems and offers a potential for new general system concepts in laboratory automation before the background of innovative Internet technologies (TGDS04) (AMF00).

Figure 1.9 (DAJ12) describes how LIMS assist laboratories in tracking and managing its information resources, particularly the data that represents the laboratory product.

LIMS has contributed in improving the administration of labs data from the perpective of a production environment. However, with respect to the research environment, LIMS has limitations such as: (1) dealing with unstructured data, (2) ensure interoperability between components in an HTS environment, (3) facilitate experiment data sharing and (4) promote reusability.

1.4.1 LIMS environment

A LIMS is more complex than just a single application and it is better to consider the term environment to specify the set of elements who interact within a laboratory. These elements are shown in Figure 1.8 (DAJ12), i.e:

1. LIMS application.

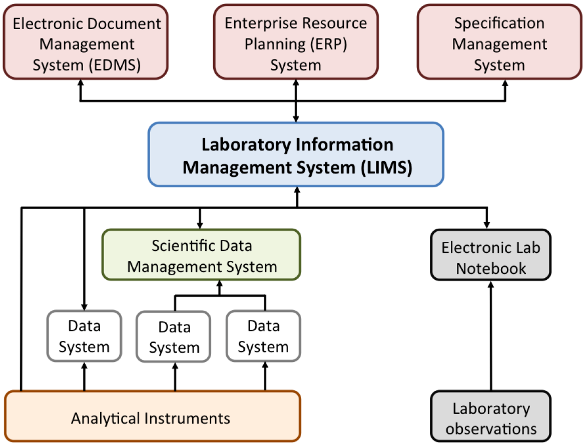

Electronic Document Management System (EDMS)
Enterprise Resource Planning (ERP) System
Specification Management System
Laboratory Information Management System (LIMS)
Scientific Data Management System
Electronic Lab Notebook
Data System
Data System
Data System
Analytical Instruments
Laboratory observations

Figure 1.8: *Diagram of a tyical LIMS environment* (DAJ12)

2. Data Analysis tools interfaced directly with the LIMS.

3. Legacy systems and computer systems interfaced with the LIMS.

4. Applications external to the laboratory that are also interfaced to the LIMS e.g. an enterprise resource planning system.

1.4.2 How a typical LIMS works

The LIMS is in charge of the analysis and distribution of the data generated in the laboratory. It becomes the single point of interaction between the data and the different tools and systems used in the laboratory. The data is stored in a single repository and this allows to centralize its flow and access through the organization (DAJ12).

Once the sample is logged into the LIMS package, it automatically takes over the further operations and generates the work sheet/protocol sheet/analytical work record. The workflow contains all the diagnostics required to be carried out for the sample (Figure 1.9).

While carrying out the actual testing, the readings and observations are noted down in the workflow. After completing the diagnostics all data are fed into the LIMS package. It automatically calculates the results, compares the findings with the standards, and also decides about the compliance with the standards. After this it prints out the certificate of analysis in the prescribed format that is already fed into the software.

It is the LIMS that decides about the conformance and non-conformance of the sample with the standard and not the analyst. The technician just has to enter the required inputs (readings/observations); the desired output format can be selected, and results can be generated in desired options without giving any external formulae for calculation (KGG10).

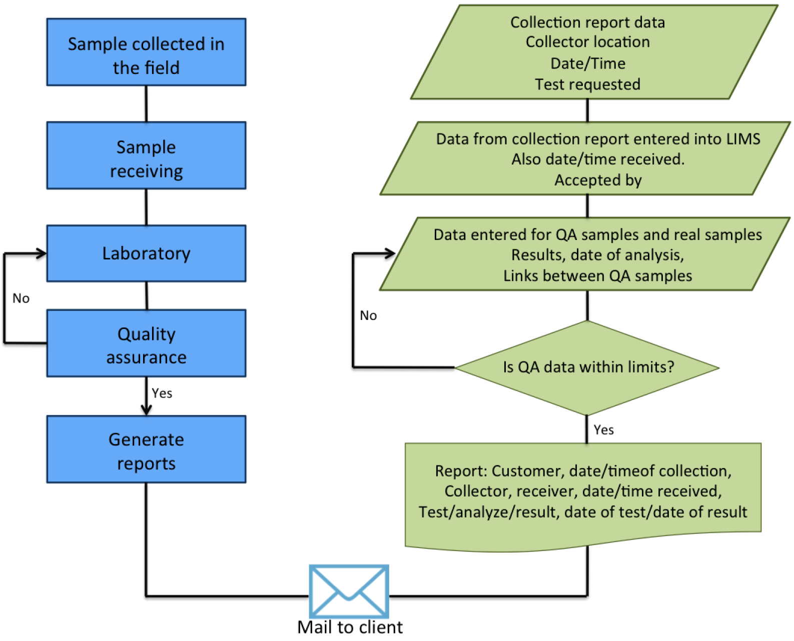

Sample collected in the field
Sample receiving
Laboratory
No
Quality assurance
Yes
Generate reports
Collection report data
Collector location
Date/Time
Test requested
Data from collection report entered into LIMS
Also date/time received.
Accepted by
Data entered for QA samples and real samples
Results, date of analysis,
Links between QA samples
No
Is QA data within limits?
Yes
Report: Customer, date/timeof collection,
Collector, receiver, date/time received,
Test/analyze/result, date of test/date of result
Mail to client

Figure 1.9: *Sample work flow diagram* (DAJ12)

1.4.3 Benefits of LIMS

A LIMS provides benefits for many of the users of a laboratory. However, a LIMS does represent an expense that must be considered. This expense will almost certainly have to be justified by a level of higher management. The following is a brief outline of several of the main benefits identified and realized from current users of LIMS (DAJ12).

1. Information can be obtained with the click of a button rather than having to dig through files.

2. Years of data can be kept easily without the need for traditional archiving.

3. The improvement of business efficiency.

4. Improvement of data quality (all the instruments are integrated).

5. Automated log-in, tracking, and management.

6. Automated customer reports (Turnaround Time, Work Load).
7. Automated Integration of Hand-held LIMS devices.
8. Automated Quality Control.
9. Daily Quality Reports.
10. Easily accessible data via the web.

LIMS is a useful tool to manage structured information produced in the laboratory. However, an HTS environment is constantly changing due to the diversity of technologies involved which produce continuosly more volume of data. Current datasets generated in HTS are commonly unstructured and not suitable for being stored in a rigid database system. The need to align research data to the FAIR paradigm force the urgent need to design new software architectures which are able to integrate the benefits of LIMS with platforms capable to manage unstructured data in Cytomics.

1.5 Thesis structure

This thesis consists of six chapters: (1) *Introduction*, (2) *CytomicsDB architecture*, (3) *Metadata management*, (4) *Metadata Validation*, (5) *Cluster Integration* and (6) *Conclusion and Discussion*.

Chapter 1 introduces the scope of thesis as well as its background. In addition, here we review the existing challenges in the management of large volumes of data in Cytomics. Thereafter, the lab management systems that currently manage lab data and their issues for dealing with high-throughput data is briefly described. Moreover, the typical workflow in High-Throughput Screening (HTS) experiments is presented.

Chapter 2 describes our research in developing CytomicsDB, a modern RDBMS based platform, designed to provide an architecture capable of dealing with the computational requirements involved in high-throughput content.

Chapter 3 introduces a semantic approach for organizing the metadata involved in High-Throughput Screening experiments. The main goal is to facilitate the exploration process in the HTS workflow, scientists are aware of semantics and they are pushing forward the need for new approaches in organizing the metadata according to which queries are mostly applied on the scientific data.

Chapter 4 provides a semantic-based metadata validation approach applied in CytomicsDB. The objective is to ensure the integrity, consistency and reliability of the data stored in the platform. This is a critical requirement for image and data analysis and further data exploration of the experiment‘s results.

Chapter 5 discusses how the integration with a computational cluster to CytomicsDB architecture can speed up the image analysis stage in the workflow of an HTS experiments.

Finally, **Chapter 6** provides the general conclusions and recommends future directions of research.

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 × 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.

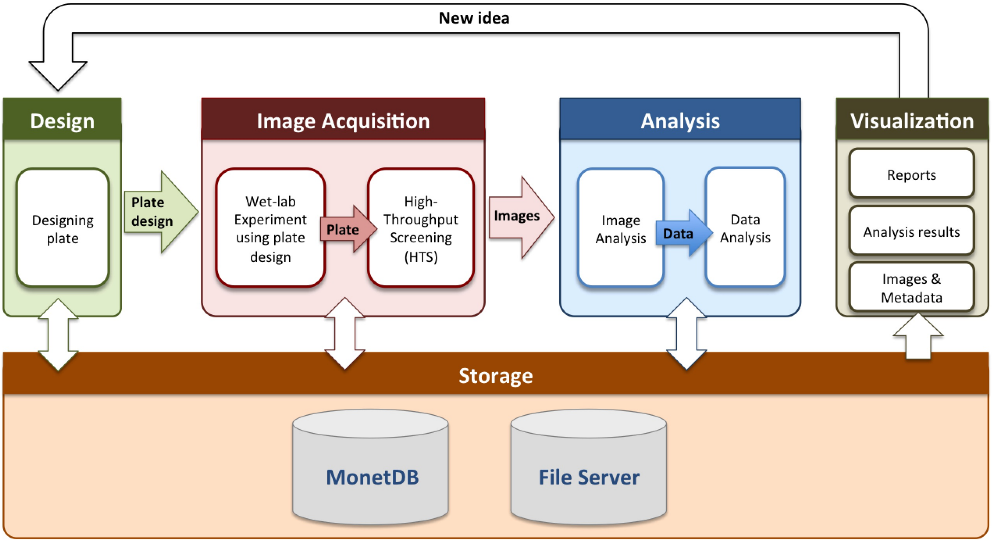


Figure 2.1: *Automated workflow of an HTS Experiment*

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.

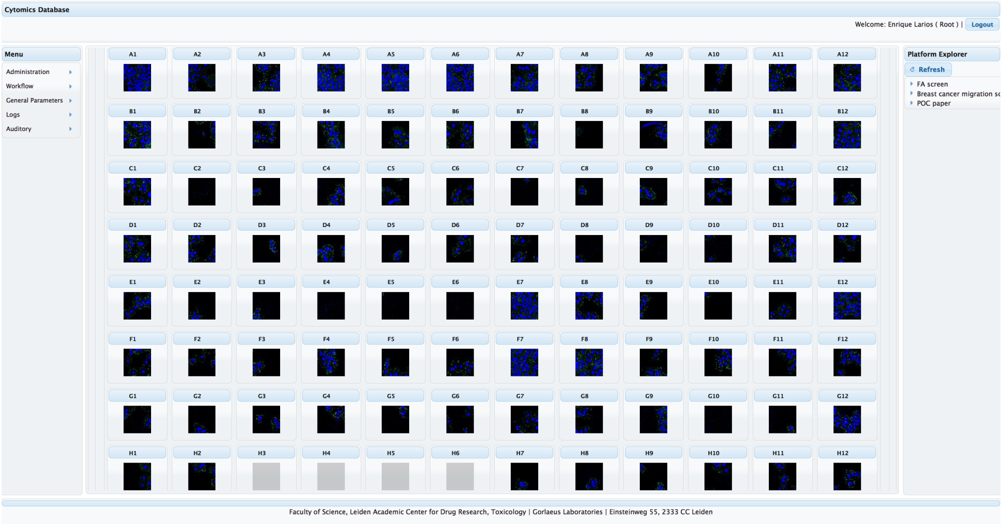
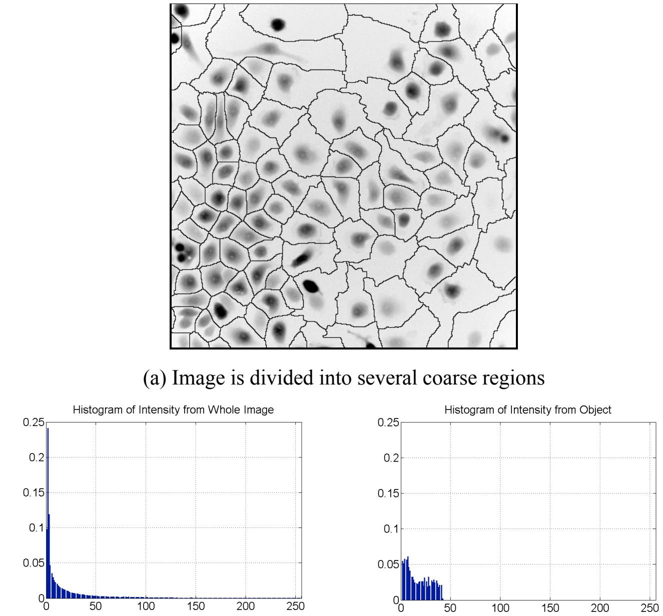

Cytomics Database
Welcome: Enrique Larios (Root) |
Logout
Menu
Administration
Workflow
General Parameters
Logs
Auditory
A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12
C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12
D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
G1 G2 G3 G4 G5 G6 G7 G8 G9 G10 G11 G12
H1 H2 H3 H4 H5 H6 H7 H8 H9 H10 H11 H12
Platform Explorer
Refresh
FA screen
Breast cancer migration sc
POC paper
Faculty of Science, Leiden Academic Center for Drug Research, Toxicology | Gorlaeus Laboratories | Einsteinweg 55, 2333 CC Leiden

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).

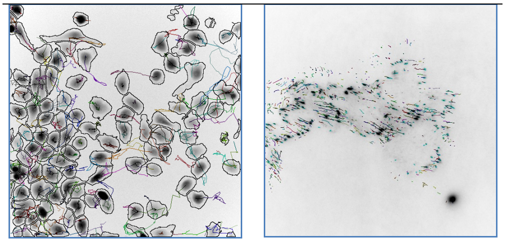


(a) Image is divided into several coarse regions

(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

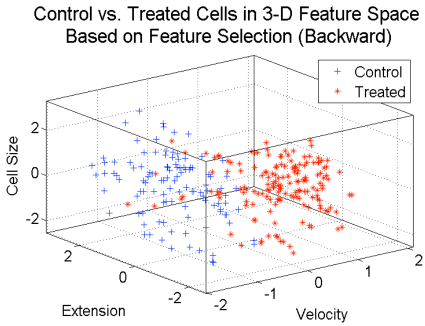


Figure 2.5: *Phenotypic characterization of the Epidermal Growth Factor (EGF) treatment using a highly aggressive cancer cell line, the illustrated features are picked by branch-and-bound feature selection. The EGF-treated cell group shows a significant increased migration velocity*

(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

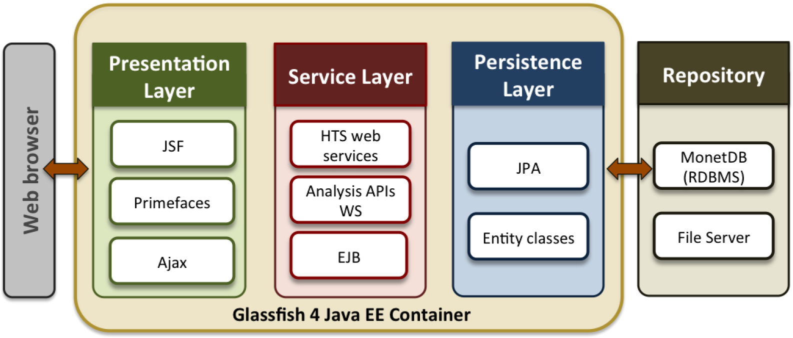


Figure 2.6: *The HTS analysis platform architecture*

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.

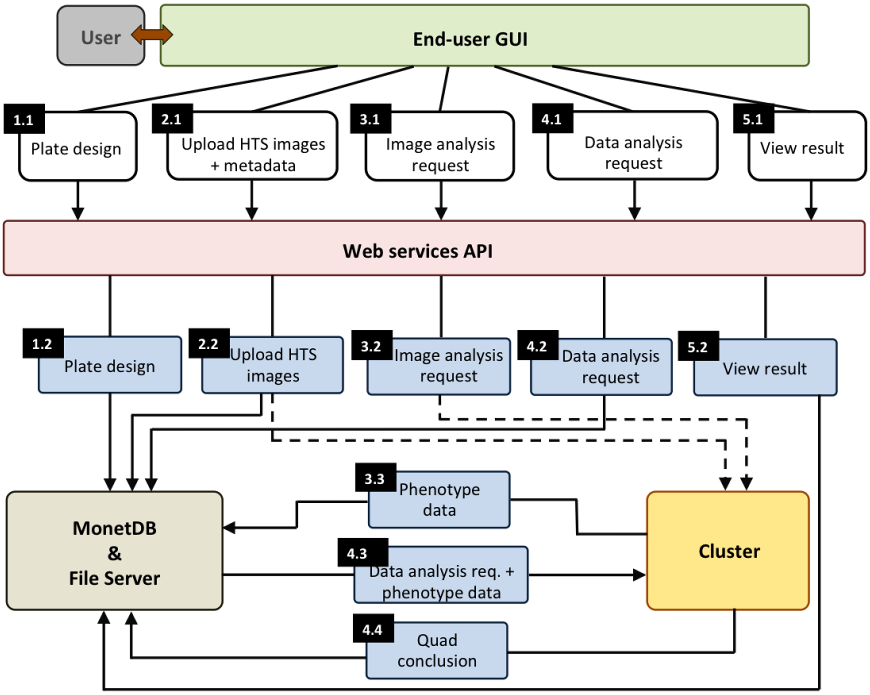


Figure 2.7: *Flow of control of the HTS platform*

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

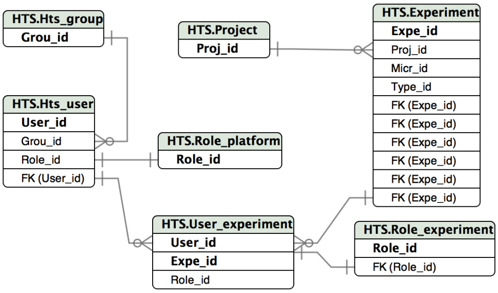


Figure 2.8: *Database schema for Project Metadata*

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

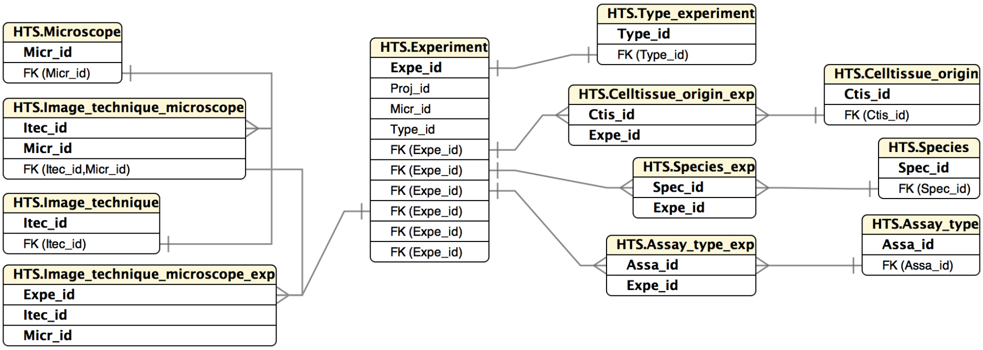


Figure 2.9: *Database schema for Experiment Metadata*

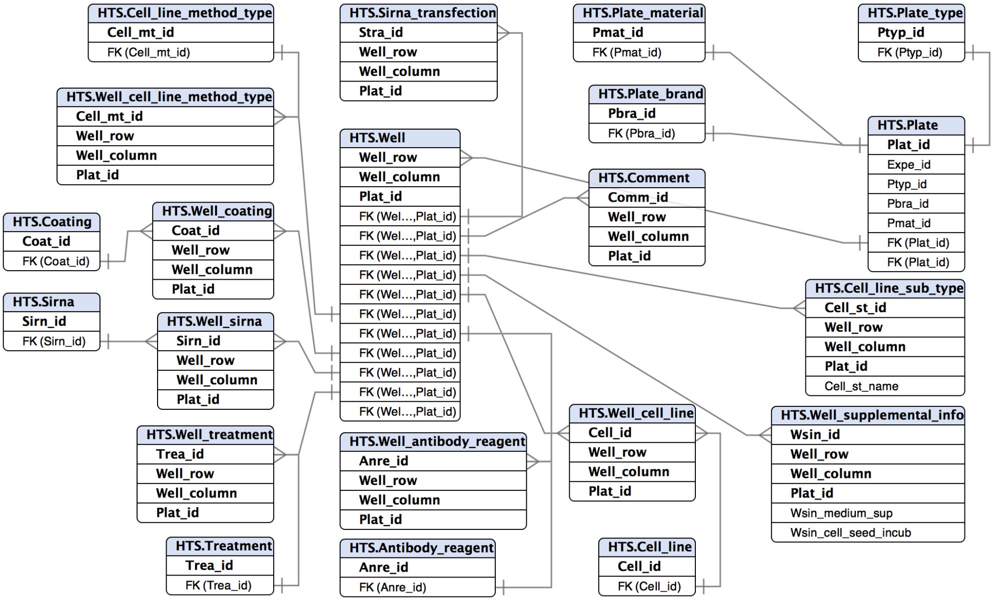


Figure 2.10: *Database schema for Plate-Well Metadata*

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

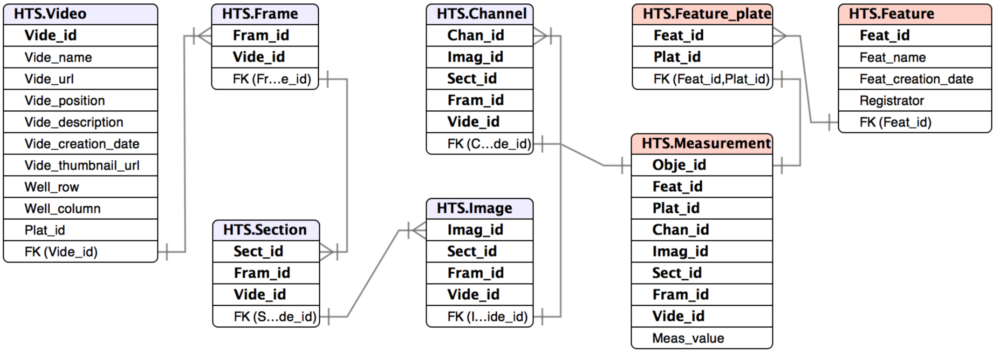


Figure 2.11: *Database schema for Raw Images and Measurement Metadata*

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.

Chapter 3

Metadata Management

This chapter describes our approach in CytomicsDB for building a semantic layer over the data so as to enable querying metadata and at the same time allowing scientists to integrate new tools and APIs taking care of the image and data analysis. These analysis results will become part of the metadata of the whole HTS experiment and will be available for semantic post analysis.

This chapter is based on the following publication:

- E. Larios, Y. Zhang, L. Cao, and F.J. Verbeek. **Cytomicsdb: A metadata-based storage and retrieval approach for high-throughput screening experiments**. Pattern Recognition in Bioinformatics (PRIB 2014). Lecture Notes in Computer Science, vol. 8626, Springer International Publishing, pages 72–84. 2014.

3.1 Introduction

High-Throughput Screening (HTS) is a well-established process in drug discovery for pharma and biotechnology companies and is now also being set up for basic and applied research in academia and some research hospitals (MF08). Recent developments in microscopy systems and robotics enabled large-scale screening of cellular systems. A popular screen setup is automated time-lapse confocal image acquisition which enables capturing of e.g. high content subcellular information (derived as features) or dynamic aspects like cell migration. Cells are exposed to hundreds and even thousands of different conditions using one or several multiwell (96, 384, 1536) plates. This typically results in 20-40 GB of data consisting of in the order of 100,000 - 200,000 images in an overnight experiment.

In cytometry, HTS-experiments are usually employed in the context of functional analysis, closing the gap between genomics-proteomics and functional responses on the cellular level. Examples are genome wide siRNA screens, where all existing genes are lowered in activity one at a time using siRNA mediated knock down followed by some cellular-level phenotypic readout, e.g., cell migration speed, focal adhesion dynamics, subcellular morphological changes, cell death.

A next step in the HTS-experiment pipeline is image quantification using image analysis software tools. In this manner, biological hypothesis can be statistically tested using the quantification results from the image analysis stage, and can depict an objective understanding of the cell response to various treatments or exposures.

In a typical HTS workflow, spreadsheet applications are commonly used for bookkeeping all information related to the design of the multiwell-imaging plates, image analysis quantification results and even statistical analysis results. This approach has many drawbacks. Firstly, it is extremely difficult to link the data produced during the different stages of an HTS experiment, such as linking the images generated in the HTS experiment and the metadata collected during the design of the plate layout. Secondly, it is highly prone to man made errors. The lack of standards, formats and a centralized place for storing the information makes it difficult to promote a collaborative environment within or between research groups. Finally, spreadsheet applications are not suitable for knowledge discovery, as they do not allow to combine sophisticated visualization and querying of the (meta)data previously stored.

In our previous work (LZY$^+$12), we presented the initial design of 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. Using this platform, the overall workload of HTS experiments, from experiment design to data analysis, can be significantly reduced. Additionally, the platform provides the potential for data integration to accomplish genotype-to-phenotype modeling studies. In this work, the initial design, particularly, the database model, has been rigorously revised and generalised to manage all kinds of metadata produced by automated HTS systems. We call our system *CytomicsDB*, which is designed as a user oriented platform but considers the HTS workflow as a template for managing, visualizing and querying the metadata.

Current software and architectures for HTS are mostly based on generic Lab Information Management Systems (LIMS) (WSP$^+$07), which face significant challenges for accessing, analyzing, and sharing the data required to drive day-to-day processes within the laboratory. Furthermore, the limited connectivity to other legacy systems and poor visualization of the data is an obstacle to extract new insights from the data stored, and cause a deep impact in the efficiency of the HTS experiment. Comparing with the existing LIMS systems, CytomicsDB has a number of important advantages:

1. Ease of promoting scientific collaborations. Since all data in CytomicsDB are centralized, granting access to collaborators or sharing information has been made simple.

2. Flexibility for integration with other legacy systems. It it common to use external APIs for performing image and data analysis results, such as Weka, PRTools. In the design of the architecture of CytomicsDB, special care has been taken to assure the possibility of invoking external API through web services.

3. The web-based architecture allows users easy access to their experiments data from wherever and at any time. The architecture also allows the whole or parts of the system to be smoothly moved to a Cloud based environment.

4. The capability to drill-down through experiments metadata due to the metadata-based approach.

5. A single interface for visualization of all experiments data, including raw images, metadata and analysis results.

6. Pattern recognition (PR) within an experiment and PR across HTS experiments.

To sum up, the contributions of this work include:

1. Metadata organization in an HTS experiment (Section 3.2).

2. A case study in endocytosis of epidermal growth factor receptor (EGFR), describing how a Metadata-based RDBMS approach can facilitate the identification of EGFR dynamics and classification of EGFR phenotype stages (Section 3.3).

Finally, in Section 3.4 we discuss related work and in Section 3.5 we present our conclusions.

3.2 Metadata organization in an HTS experiment

The metadata of an HTS experiment consists of a variety of types and formats and has been grouped in five levels as showed in Figure 3.1: Project, Experiment, Plate - Wells, Video/Images and Measurements. These levels contain each other in a cascade fashion, for instance: [1] Project contains [1..n] Experiments, [1] Experiment contains [1..n] Plates, [1] Plate contains [24,48,96,384] Wells, [1] Well contains [1..n] Video/Images and finally [1] Well contains [1..n] measurements.

3.2.1 Project

This level contains a title which describes the aim of the project, the duration, the author, etc. When a project is created, its creator becomes its administrator and is possible to grant access to another scientist in order to promote a collaborative environment.

3.2.2 Experiment

Figure 3.2 shows the structure of the metadata contained in the Experiment level. This level is divided in Hardware and Type of Experiment. Firstly, the metadata associated to the hardware correspond to the microscope and the imaging technique used. Depending on which microscope is used, the set of imaging techniques differs. For instance, the imaging techniques available for a Becton Dickinson (BD) Pathway microscope are EPI, Spinning disk or Bright Field, but in a Nikon TE 2000-e microscope it is possible to use: FRAP, FRET, EPI, Confocal, Spectral or DIC. Secondly, the metadata associated to the type of experiment can be separated in four groups: (1) Fixed or Live experiment including a 2D or 3D option for each case; (2) Assay type, in this case there are the following options: migration/invasion, proliferation, primary tumor, apoptosis and sub cellular perturbations; (3) Species, the options available are: human, rat, mouse and zebrafish; and (4) Cell / Tissue origin, considering in this area: primary, cell line, iPSC, stem cel, biopsy, etc.

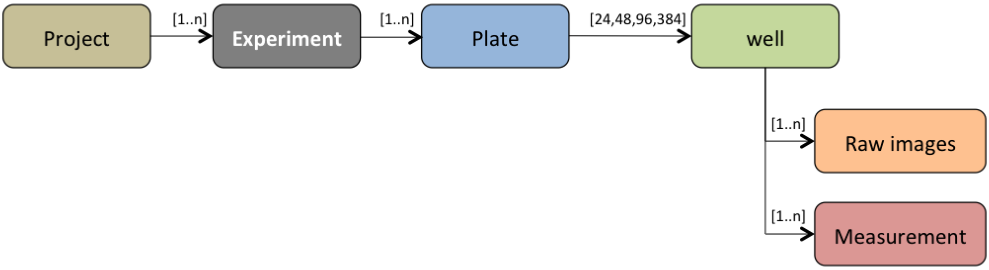


Figure 3.1: *Structure of the metadata in an HTS experiment*

3.2.3 Plate-Wells

This level is also divided into two groups: metadata about the Hardware and about the Parameters used. Figure 3.2 shows a diagram of the structure of these two groups. The Hardware sub level includes information about the plate type, the brand and the fabrication material. The level of Parameters includes information about (1) Coating, (2) Cell-line / tissue, (3) treatment, (4) siRNA, (5) Antibodies / reagents and (6) Parameters of control or comments. The metadata of Wells is a subset of metadata of the plate level. For instance, in a 8x12 wells plate, different wells can have a subset of the parameters assigned to the whole plate. This level is also associated with the output of the HTS process (Raw Video / Images) and with the results of the image and data analysis phase which is also called measurements.

Part of the metadata at this level is critical information that should be verified and validated when it is uploaded. For instance, the parental cell line/tissue, or the treatment and its concentration are just two cases which the entry is verified in a

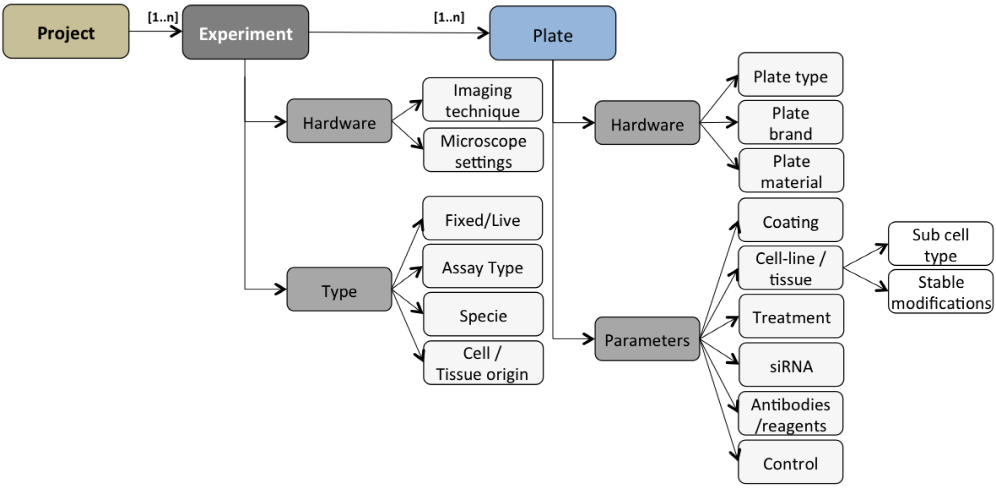


Figure 3.2: *Structure of the Experiment and Plate metadata*

first instance (obligatory data) and then they are validated with the information pre loaded in the imaging database. In order to keep the consistency of the metadata it is necessary to validate each entry and when a new value is detected the administrator of the platform is in charge of accepting this new entry as valid or correct to the right value if it is necessary. The consistency in the metadata is a key task in the imaging database because the obligatory data will be further used as a controlled vocabulary for querying.

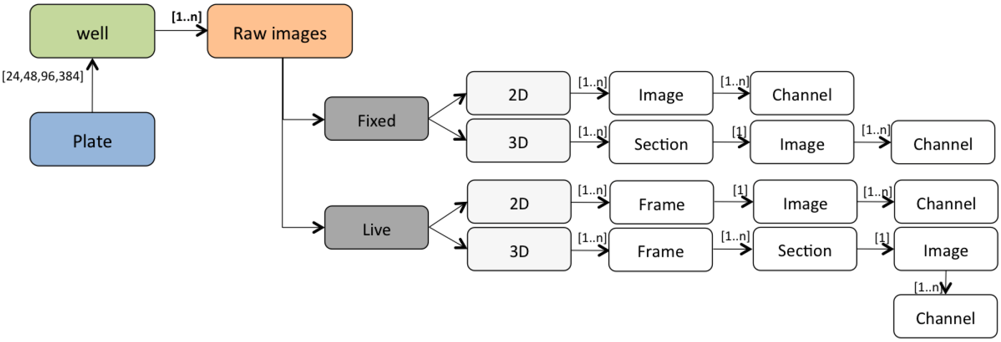


Figure 3.3: *Structure of the Raw Images metadata*

3.2.4 Raw Images

Raw images are obtained after image acquisition with automated microscopy systems. These images are the basis for the image analysis which results in quantitative data used for hypothesis testing. 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 (Fixed/Live) and the microscopy technique used in the experiment. Section 2.4.3 describes four types of structures supported (cf. Figure 3.3) (LZY+12): (1) *2D (XY)*, (2) *2D+T (XY+T)*, (3) *3D (XYZ)*, and *3D+T(XYZ+T)*.

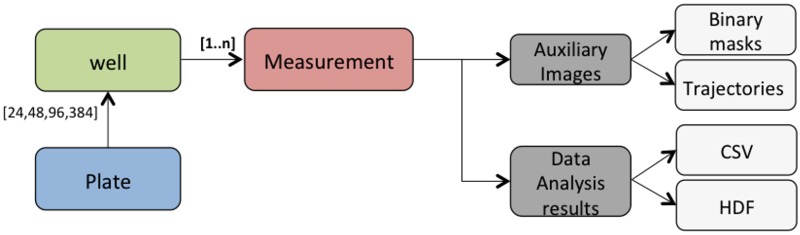


Figure 3.4: *Structure of the Measurements metadata*

3.2.5 Measurements

This level contains the results of the Image and Data Analysis process (cf. Figure 3.4):

Results of Image Analysis

Section 2.4.4 describes the type of output in the image analysis stage. The goal in this stage is to convert images into measurements by using image segmentation and object tracking procedures. These measurements are part of the metadata and are also linked to the raw video image file, on which the image analysis has been applied.

Results of Data Analysis

Section 2.4.5 describes the results of the data analysis stage. The measurements obtained from the analysis step are stored in the database, and can be queried in order to perform further analysis using pattern recognition tools. A CSV file is generated with the results accompanied by a HDF file (Gro10) with information of the structure of the CSV file (features).

3.3 Case Study in Endocytosis of EGFR: Identification of EGFR dynamics and classification of EGFR phenotype stages

In this section we describe a case study on how the structure of the metadata and RDBMS are applied in order to identify the EGFR dynamics and classify the different EGFR phenotypes.

Endocytosis is regarded as a mechanism of attenuating epidermal growth factor receptor (EGFR) signaling and of receptor degradation. Increasingly, evidence becomes available showing that cancer progression is associated with a defect in EGFR endocytosis (dGCW^{+}13). Functional genomics technologies combine high-throughput RNA interference with automated fluorescence microscopy imaging and multi-parametric image analysis, thereby enabling detailed insight into complex biological processes, like EGFR endocytosis. The experiments produce over half a million images. Such a volume of images is beyond the capacity of manual processing and therefore, image processing and machine learning are required to provide an automated analysis solution for HTS experiments (CYW^{+}11). The total size in average can vary between 500 Mb to 20 Gb of raw images per experiment and CytomicsDB is designed to cope with the growing data size due to the scalable architecture for storing the images in a File Server and the metadata of the entire experiment in the database (see Figure 2.6).

According to the methodology described in (CYW^{+}11), three stages are identified: (1) Image Acquisition, (2) Image Analysis and (3) Data Analysis. We describe each stage as follows:

3.3.1 Image Acquisition

The experiment "Endocytosis of EGFR" and its respective plates is created in the CytomicsDB platform. The type of metadata required for creating an experiment and the plates in our platform is described in Section 3.2. The respective values associated to each type of metadata have been detailed in (CYW^{+}11). After designing the plate in the platform, the wet-lab experiment is initiated, which includes the following steps: (1) cell culturing, siRNA transfection and EGF exposure, (2) fluorescent staining of proteins of interest and (3) image acquisition. Upon completion of the acquisition process 960 images are uploaded to the platform which size in total is 767 Mbytes. These images correspond to a 96 wells plate (cf. Figure 3.5) and for each well, images are captured from ten randomly selected locations. However, an experiment can consist of more than one plate and the number of samples per well can differ per case.

3.3.2 Image Analysis

The API in charge of the image analysis, request from the database the location of each image to process. The query executed is:

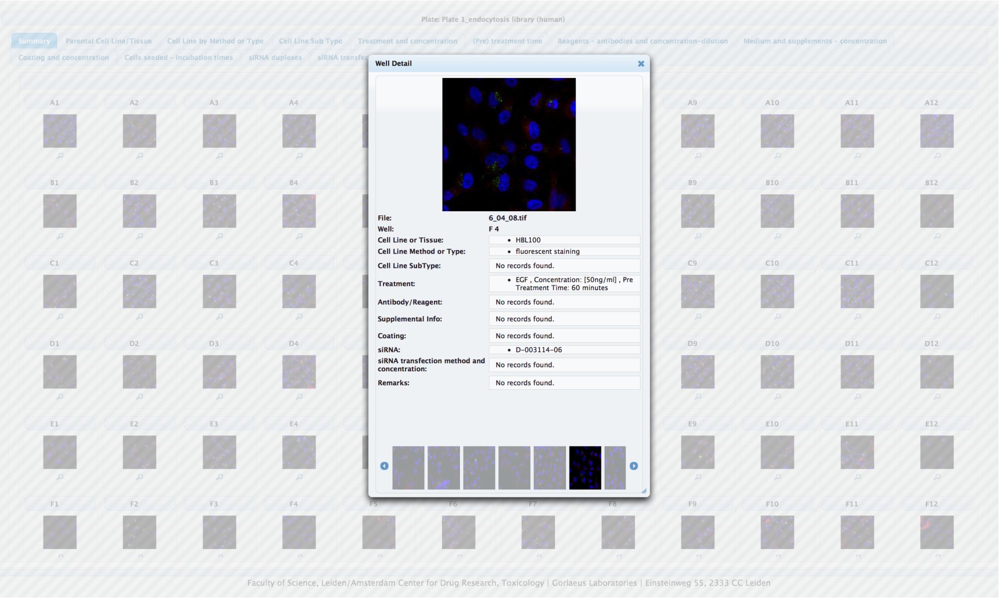


Plate: Plate 1_endocytosis library (human)
Summary
Parental Cell Line/Tissue
Cell Line by Method or Type
Cell Line Sub Type
Treatment and concentration
(Pre) treatment time
Reagents - antibodies and concentration-dilution
Medium and supplements - concentration
Coating and concentration
Cells seeded - incubation times
siRNA duplexes
siRNA transfec
Well Detail
A1 A2 A3 A4 A9 A10 A11 A12
B1 B2 B3 B4 B9 B10 B11 B12
C1 C2 C3 C4 C9 C10 C11 C12
D1 D2 D3 D4 D9 D10 D11 D12
E1 E2 E3 E4 E9 E10 E11 E12
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
Faculty of Science, Leiden/Amsterdam Center for Drug Research, Toxicology | Gorlaeus Laboratories | Einsteinweg 55, 2333 CC Leiden

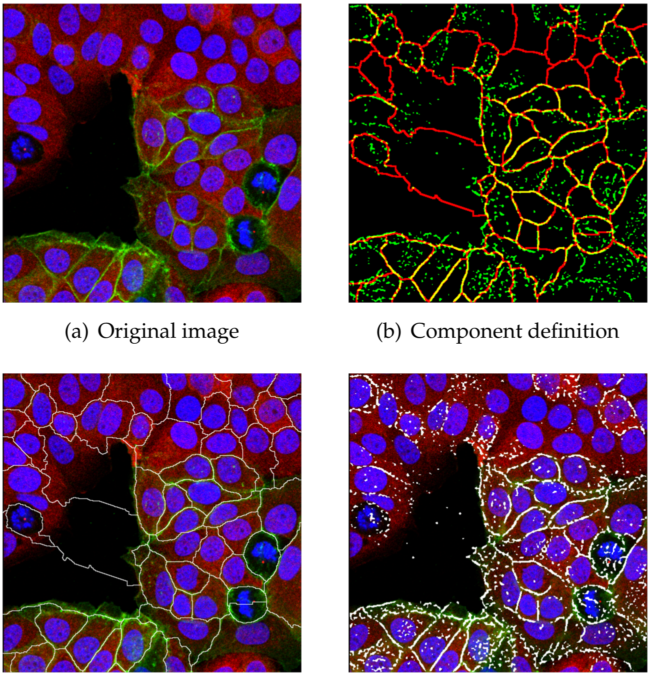
File:	6_04_08.tif
Well:	F 4
Cell Line or Tissue:	• HBL100
Cell Line Method or Type:	• fluorescent staining
Cell Line SubType:	No records found.
Treatment:	• EGF , Concentration: [50ng/ml] , Pre Treatment Time: 60 minutes
Antibody/Reagent:	No records found.
Supplemental Info:	No records found.
Coating:	No records found.
siRNA:	• D-003114-06
siRNA transfection method and concentration:	No records found.
Remarks:	No records found.

Figure 3.5: *Web plate layout*

```
SELECT v.vide_id, v.vide_name, v.vide_url, v.vide_position, v.well_row, v.well_column
FROM HTS.Video v
WHERE v.plat_id = 17;
```

The value of column *plat_id* is in this case “17” and it was assigned after selecting *the plate for endocytosis* in the web interface. Three steps are performed by this API: (1) noise suppression, (2) image segmentation and (3) phenotype measurement. The algorithms and process details are described in (CYW$^+$11). Upon completion of the image analysis process, the API returns two outputs: (1) The location in the database of a new set of images and (2) currently, a CSV file containing the features and the phenotype measurement respectively. The set of images generated are: (a) Original image: PERK (red), EFGR (green) and nucleus (blue) (cf. Figure 3.6(a)), (b) Component definition: artificial cell border (red) and binary mask of protein expression (green) (cf. Figure 3.6(b)), (c) Cell border reconstruction: artificial cell border (W-V) (cf. Figure 3.6(c)), (d) Image segmentation: binary mask of EFGR channel by WMC (cf. Figure 3.6(d)) (YV12). The phenotype measurements (CSV file) are parsed first and then stored in the database by a web service. For instance, in case of the first measurement the following query is executed:

```
INSERT INTO HTS.Measurement
(Obje_id, Feat_id, Plat_id, Chan_id, Imag_id, Sect_id, Fram_id, Vide_id)
VALUES (0,1,17,1,1,1,1,1,14.0);
```



(a) Original image

(b) Component definition

(c) Cell border reconstruction

(d) Image segmentation

Figure 3.6: *Example of set of images generated during the Image Analysis stage* (YV12)

In this example, the column *Feat_id=1* corresponds to *Area* in the entity Feature and the measurement obtained for this feature is 14.0. The column *Plat_id* is still 17 because we refer to the same plate.

The measurements are categorized in two subgroups: (1) basic measurements of the phenotypes covering shape descriptors and (2) the localization phenotype describing the assessment of the correlation between two information channels. The basic phenotype measurement includes a series of shape parameters such as: size, perimeter, extension, dispersion, elongation, orientation, intensity, circularity, semi-major axis length, semi-minor axis length, closest object distance and in nucleus, these can be extended as the experiments so dictates. In addition to the basic phenotype measurement, localization measurements can be derived for a specific experimental hypothesis. The localization phenotypes are quantifications of comparative measurement between information channels such as relative structure-to-nucleus distance or structure-to-border distance. The features in EGFR-screen based localization phenotypes used are: nucleus distance, border distance and intactness. On the basis of the phenotype measurements, objects are classified into phenotypic stages. For the assessment of significance statistical analysis is performed (CYW$^+$11). Upon completion of the image analysis, it is possible to visualize the results in a web plate layout and export the measurements to files.

3.3.3 Data Analysis

The aim of the endocytosis study is to quantify the process of EGF-induced EGFR endocytosis in human breast cells and to identify proteins that may regulate this process. The EGFR endocytosis process can roughly be divided into three characteristic episodes: i.e. (1) at the onset EGFR is present at the plasma-membrane; (2) subsequently, small vesicles containing EGFR will be formed and transported from the plasma-membrane into the cytoplasm; and (3) finally, vesicles are gradually merging near the nuclear region forming larger structures or clusters. The characteristic episodes are the read-out for HTS. Based on this model it is believed that EGFR endocytosis regulators may be potential drug targets for EGFR-induced breast cancer. Studying each of the stages, i.e. plasma-membrane, vesicle and cluster, may provide a deeper understanding of the EGFR endocytosis process (CYW$^+$11).

When the data analysis process is triggered, a web service request from the database (entities feature and measurement) the results from the image analysis process. The output of this web service is the location of a file which contains the results of the test for each siRNA regulator. This file will be requested for the API PRTools for generating classifications and graphs with the comparison of the results, such as: (1) Weighted classification error curve, which represents a combination of a feature selection/extraction method and a classifier algorithm, (2) Results of the feature extraction and (3) Average number of plasma-membrane (a) and vesicle (b) per nucleus (CYW$^+$11). Consolidating in CytomicsDB the experiment's metadata, raw images and images/data analysis results, facilitates further comparison with the result of other HTS experiments.

3.4 Related work

In the current area of -omics research, various systems/tools have emerged to try to solve the problem that the existing practice of keeping metadata does not allow for effective data searching and mining. They are generally referred to as Laboratory Information Management System (LIMS).

The work proposed by Colmsee et. al. (CFK$^+$11) is probably the closest to CytomicsDB. The authors defined central requirements for a primary lab data management and aspects of best practices to realise those requirements. As a proof of concept, the authors implemented a pipeline to manage primary lab data of crop plants. The pipeline consists of i) data storages including a Hierarchical Storage Management system, an RDBMS and a BFiler package to store primary lab data and their meta information; ii) the Virtual Private Database for the realisation of data security and the LIMS Light application to iii) upload and iv) retrieve stored primary lab data. Compared with this work, CytomicsDB has a more sophisticated data model to cope with different types of data (i.e., images, videos, and data produced in different steps in an HTS experiment), pays special attention to the extensibility of the architecture to enable adding new tools.

In (NSD$^+$10), the authors presented three open-source, platform independent software tools for genomic data: a next generation sequencing / microarray LIMS and analysis project center (GNomEx); an application for annotating and programmatically distributing genomic data using the DAS/2 data exchange protocol (GenoPub); and a standalone Java Swing application (GWrap) that provides a GUI for the command line analysis tools. CytomicsDB provides similar functionalities as these tools, but focuses on dealing with Cytomic data. Moreover, for the design of CytomicsDB, we have deliberately chosen for a single integrated system to include all features required for conducting HTS experiments and analysis, instead of individual tools and enabling high profile pattern recognition.

In (WSP$^+$07), the authors describe a general modeling framework for laboratory data. The model utilises several abstraction techniques, with focus on the concepts of inheritance and meta-data. In this model, distinct regular entity and event schemas can be defined and fully integrated via a standardized interface. The design allows definition of a processing pipeline as a sequence of events. A layer above the event-oriented schema integrates events into a workflow by defining processing directives, which act as automated project managers of items in the system. This LIMS is built on the Oracle RDBMS, and is maintained by multiple database administrators (DBAs). While with CytomicsDB, our goal is to meet the needs of HTS experiments with a more light-weight, flexible system. By adapting modern web and database technologies, CytomicsDB is easy to maintain and extend (i.e., allowing integrating new tools naturally).

The work by Chan et al. (CMS06) focuses on interactive visualization methods for data generated by HTS experiments. The visualization methods might be adapted by CytomicsDB. However, CytomicsDB is a much more comprehensive information system for HTS data, because it integrates both experiments and analysis data into a single system, and allows various types of users and groups to be defined.

Based on the Golm Plant Database System, Köhl et. al. (KBL$^+$08) devised a data management system based on a classical LIMS combined with web-based user interfaces for data entry and retrieval to collect this information in an academic environment. This system stores plant cultivation units in an MS ACCESS database, which would quickly run into scalability issues as the data size grows.

3.5 Conclusions and Future Work

In this chapter, we have presented a metadata based storage and retrieval approach for organizing data in High-Throughput Screening experiments. Our goal is to facilitate the exploration process in the HTS workflow, scientist are aware of semantics and they are pushing forward the need for new approaches in organizing the metadata according to which queries are mostly applied on the data. In HTS, images by itself do not have any meaning, but linking images to their respective metadata allows researchers to learn from their experience and help them in mentalizing semantic structures of the metadata. Finally, we plan to extend CytomicsDB architecture to a more sophisticated interdisciplinary platform for cytomics. The structure of the metadata proposed in this chapter will further evolve to an ontology based framework. A new layer to the architecture will be added in order to perform semantic queries, turning the architecture to a web based interactive semantic platform for cytomics (BSV11).

Chapter 4

Metadata Validation

This chapter describes our research in the validation process as performed in CytomicsDB. This system is a modern RDBMS based platform, designed to provide an architecture capable of dealing with the strict validation requirements during each stage of the HTS workflow. Furthermore, CytomicsDB has a flexible architecture which support easy access to external repositories in order to validate experiments data.

This chapter is based on the following publication:

- E. Larios, Z. Xia, J. Slob and F. J. Verbeek. **A semantic-based metadata validation for an automated High-Throughput Screening workflow: Case study in CytomicsDB**. NETTAB 2015 - Integrative Bioinformatics 2015. (Submitted).

4.1 Introduction

High-troughput screening (HTS) assays provide a way for researchers to simultaneously study interactions between large numbers of potential drug candidates with a target of interest. Significant volumes of data are generated as a consequence of conducting HTS experiments, making it cumbersome to store, interact, and throughly mine the data for biological significance (MCV^{+}06).

Due to the large amount of resources invested and the complexity of the processes performed during an HTS experiment, it is convenient and necessary to build an automated workflow system which will be in charge of the management, supervision and validation of the data in every stage of the workflow. In our previous work

(LZY+12), we presented the initial design of an automated workflow system and in (LZCV14) we have given more detailed description about how it works in a case study. Our automated HTS workflow (cf. Figure 2.1) is divided in 5 stages: (1) Design, (2) Image Acquisition, (3) Analysis, (4) Visualization and (5) Storage. This last stage is performed in parallel with the first 3 stages. It is extremely relevant to give special care to the information stored in the platform in order to have reliable output available to display in the Visualization stage.

CytomicsDB is our metadata-based storage and retrieval approach for HTS experiments and it *seamlessly* integrates the whole HTS workflow into one single system. The platform relies on a modern relational database system to store experiments data and process user requests, and at the same time it provides a convenient web interface to end-users. Using this platform, the overall workload of HTS experiments, from experiment design to data analysis, can be significantly reduced (LZCV14).

Management of the HTS information is one of the key challenges for drug discovery and in order to ensure consistency, integrity and reliability of the data stored in the platform it is compulsory to perform a strict validation process in every stage of the HTS workflow. CytomicsDB facilitates this validation process using web services which will prove each critical entry with an internal or external repository. The metadata that we store become a key parameter for performing further image/data analysis and drill down the results of different experiments datasets.

To sum up, the contributions of this chapter comprehend:

1. The automated HTS workflow managed by CytomicsDB, identifying the key validation issues (Section 4.2).
2. The validation strategies applied in CytomicsDB (Section 4.3).
3. The validation workflow developed by CytomicsDB (Section 4.4).
4. The CytomicsDB architecture, identifying the components in charge of the validation process (Section 4.4).
5. The analysis of the strategies performed in CytomicsDB for solving inconsistency conflicts (Section 4.5).

4.2 Critical validation in an Automated HTS workflow

In Figure 2.1 is possible to notice that each stage provides critical input data to the next stage in the workflow and the urgent need to validate the data involved in these steps become a key factor in order to obtain accurate and consistent results in the whole process.

In the following sections, it will be described the stages of the HTS workflow which are subjected to validation activities. The main validation activities are performed in the first two stages of the HTS workflow: (1) Experiment design and (2) Image acquisition. In this way, the metadata strictly validated can be succesfully used for the next stages e.g. image and data analysis.

4.2.1 Design

This stage entails the most important validation step, i.e the design of the plate and layout, it is also in charge of linking the experiment metadata with the previously plates designed. In the design stage the scientist uploads a spreadsheet file to CytomicsDB containing the experiment metadata at both plate and well level. The use of spreadsheet applications is susceptible to errors, thus a strict validation process is performed in two steps:

Pre design validation

CytomicsDB stores critical metadata in special entities called *Master Tables*, everytime a new value is stored in these tables, a validation is done by accessing external biological databases which are supported and validated by the scientific community.

Design validation

The spreadsheet file containing the experiment metadata is validated with the *Master Tables* and in case of any inconsistency, a warning is displayed and registered in the system. The role administrator will authorize the new entry in case of the metadata was not identified in the master tables or select one of the possible solutions provided by the system. The administator is also able to track record of changes in the platform.

4.2.2 Image Acquisition

Upon completion of the plate design stage, the image acquisition process begins. This stage is divided in two steps: Wet-Lab Experiment and HTS Process. In this stage, it is necessary to validate the image metadata such as: image type, format type, images naming convention and the settings used by the microscope. These critical metadata is used for linking the images to their respective wells.

4.3 Validation strategies

The validation process can be abstracted as a strategy. In this strategy, each object which objectively exists in the real world (e.g. a *Compound*, or a *siRNA*) is defined as an entity E. For each entity, several attributes are assigned to it, like the name, the ID number and the publisher. The attributes that are used to describe one entity can be defined as a set: $A=\{a_1,a_2,...,a_n\}$, in which a_i $(1 \leq i \leq n)$ means the i^{th} attribute of the entity. In this strategy, multiple data sources are involved as well. They can be categorized as two types. One set is from the lab in which researchers use CytomicsDB to manage their experiment data. In CytomicsDB, this set is uploaded by researchers and stored as *master tables* in the database. Another group of sources are from external databases. They are used to validate the metadata uploaded by researchers. All these data sources can be expressed as a collection S={s_1,s_2,...,s_m} in which s_i ($1 \leq i \leq m$) represents the i^{th} data source among the *m* data sources. Adopted from (YXQZZH12), the data source s_i offers a fact value f for the attribute a_j of an entity E. different data sources may have different fact values for a same attribute of the entity. For the entity *E*, if $a_j \in [a_1, a_n]$, $f_{(s_i,a_j)} \neq f_{(s_l,a_j)}, i \neq l$, then a conflict or inconsistency is found between data source s_i and s_l. In all fact values from all data sources, those who correspond to the attribute value in the real world are referred to as *true value*. So the validation process is in fact a process for identifying *true values* among all conflicts between data sources. The relationships among the entity, its attributes and fact values from different data sources in CytomicsDB are sketched in Figure 4.1. In CytomicsDB, the idea is to validate the metadata in S_2 by data from data sources S_1 who have the same attributes as S_2. Conflicts found among data sources during this process indicate potential inconsistency. There are several available strategies to perform the conflict resolution which are described in the following section.

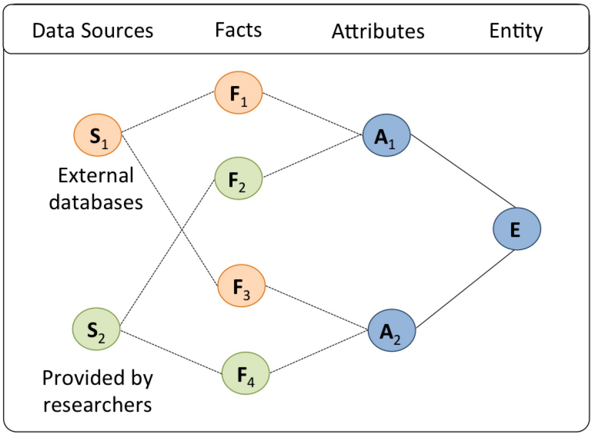


Figure 4.1: *Relationship between the entity, its attributes and fact values from data sources*

4.3.1 Strategies and algorithms

It is common to have an implicit assumption that the content of all the information sources should be mutually consistent (Mot99) in the approaches for solving inconsistency issues between different data sources. A data inconsistency exists when two entries (or tuples in the relational data model) coming from different information sources are identified as versions of each other i.e. they represent the same real-world object and some of the values of their corresponding attributes differ (Mot01). The available conflict resolving strategies are then identified by the ways they elaborate this basic assumption e.g. *Trust your friends* strategy assumes that the tuples from trustful data sources are most likely to be the real-world object, then the data under validation should be consistent with them. Some standard inconsistency resolutions from (BN09) are listed hereafter.

No gossiping strategy

The basic idea here is that if multiple objects are retrieved from the excution of queries from different data sources and it is unsure about which object or which value for an attribute in the object matchs to the real-world one, then just leave them out and only report on the sure facts, or directly report all of them. This is the strategy used by the consistent query answering approaches (FFM05). This strategy leaves the decision force to users if all query results are reported, which makes it simple to be implemented.

Trust your friends

In the *Trust your Friends* strategy it is required to trust a third party who will provide the correct entries. An assumption is considered that the fact values from reliable external databases can be treated as true values. Especially when those fact values are identical to the ones given by researchers, the possibility that those fact values are true values becomes quite high which can be assumed as 100%. This conflict resolution strategy is referred as *Trust Your Friends* (BN06). The key point of this strategy is having reliable data sources.

Cry with the wolves

The *Cry with the wolves* strategy pursues a different approach, the entries which correctly describe the real-world object prevail over the incorrect ones, given enough evidence. It reflects the principle of following the decision of the majority, of choosing the most common entries among candidates from all data sources and compare to the one under validation (BN06). The more data sources involved in the validation process, the better the strategy work.

Meet in the middle

In contrast with the previous strategies, the *Meet in the middle* strategy follows the principle of compromise and does not prefer one value over the other but instead tries to invent a value that is as close as possible to all present values from all data sources and compare it to the one under validation (BN09).

Keep up to date

This strategy uses the most recent entries from external data sources to compare to the one under validation. Some additional time-stamp information about the recentness is required in order to do the comparison (BN06).

4.4 Implementation

There are usually two types of data inconsistency (BN06): contradictions and duplications. For the contradictions, they may be caused by typos, version updates, shuffle of attributes, etc. Perceiving duplications for entities with unique identities is easy. The validation can be performed on the identifier attributes. Otherwise, additional identifier attributes are needed to perceive duplications, which add complexity to the problem. The goal of the validation process implemented in CytomicsDB is to detect and correct the inconsistent data in the entities of the metadata.

4.4.1 Validation subjects

In CytomicsDB, metadata attributes are mapped as fields in each table which are called *master tables*. All the metadata stored in CytomicsDB can be validated for internal consistency to increase the accuracy and reliability of the metadata for HTS experiments. In this paper, compounds which have only one attribute without unique identifier and siRNAs which have several attributes beside unique identities are used as test cases for the implementation of the validation process.

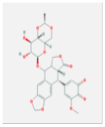
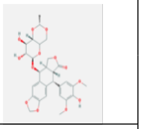
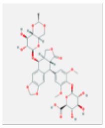
Compounds

Compounds is one of the key entries in the HTS experiments metadata. In CytomicsDB, only the name of each compound is adopted. The consistency of this name can be validated by using the NCBI PubChem Compound database (BWTB08). For instance, in case of validating the compound *ETOPOSIDE*. The researchers just offer the name of the compound. The validation process needs to check if the compound's real name is *ETOPOSIDE* and if it has been registered in the *master table* by another name. The following results are shown in Table 4.1 after querying in the NCBI database using the compound given name.

It is possible to see in Table 4.1 that the name for a compound is not an unique identifier. A compound entity can be described with several kinds of names like the source name, the Medical Subject Headings (MeSH) terms, the synonym names, etc. A compound entity can have multiple names in each category as well (e.g "71316630" compound has two synonym names).

The CID (PubChem Compound Identification) is a non-zero integer PubChem accession identifier for a unique chemical structure. So it can be used as additional

Table 4.1: *Query result after checking for ETOPOSIDE in the NCBI database.*

CID	Name	Name type	Molecular weight	Molecular formula	Structure
71316630	Etoposide o-Quinone	synonym	572.514120	$C_{28}H_{28}O_{13}$	
	Etoposide 3′,4′-Quinone				
59360017	Etoposide	MeSHHeading	588.556580	$C_{29}H_{32}O_{13}$	
		synonym			
46173784	Etoposide glucuronide	synonym	764.680700	$C_{35}H_{40}O_{19}$	
		MeSHHeading			
		MeSHTerm			

information to detect duplications. As shown in Table 4.1, the molecular weight and formula are the most distinguishable attributes for the researchers. So these two attributes are included into the process to assist the researchers to take a final decision. The 2D structure of each entity is also used in a similar way.

siRNAs

Small interfering RNA (siRNA) (Han02) is a class of 20-25 base pairs in length, double–stranded RNA molecules. In common cases, the siRNA is designed as a gene knockdown tool to interfere with the expression of specific genes with complementary nucleotide sequences. siRNA inhibits expression from its homologous gene, i.e. the sequence of siRNA is a sub-sequence of its homologous DNA's sequence. The symbol, ID, accession number and GI number of the siRNA follows the homologous gene as well. Usually one of the double strands of a siRNA sequence is recorded in the database. The sequence of a siRNA referenced in the rest of this paper refers to a one strand sequence if it is not specified. For HTS experiments, the siRNA target is of crucial importance. One typical example of a siRNA provided by the researchers is listed below in Table 4.2.

Table 4.2: *siRNA example provided by the researchers.*

Duplex number	Gene ID	Gene Symbol	Accession Number	GI Number	Sequence
D-004105-01	7272	TTK	NM_003318	34303964	P.M.

The consistency of this siRNA can be validated using external databases such as NCBI Nucleotide (Miz02), HGNC (HUGO Gene Nomenclature Committee), Gene symbols/IDs database (BLW$^+$08), BLAST+ (Basic Local Alignment Search Tool) sequence alignment application (JZR$^+$08) and EMBL database. The result of searching all attribute values of the siRNA using the external repositories is shown in Table 4.3.

Table 4.3: *Query results from all external data sources.*

Query Keyword	External Data Source	Gene ID	Gene Symbol	Accession Number	GI Number	Sequence
GeneID: 7272	HGNC IDs	7272	TTK	NM_003318	262399359	100% aligned
GeneSymbol: TTK	HGNC Symbols	7272	TTK	NM_003318	262399359	100% aligned
Accession Number: NM_003318	NCBI Nucleotide	7272	TTK	NM_003318.4	262399359	100% aligned
GI Number: 34303964	NCBI Nucleotide	7272	TTK	NM_003318.3	34303964	100% aligned
Sequence: P.M.	BLAST+	100969041	TTK	XM_008969441.1	675737708	100% aligned
		7272	TTK	NM_003318.4	262399359	100% aligned
		7272	TTK	NM_001166691.1	262399360	100% aligned

Table 4.3 shows that querying with each fact value of attributes in the siRNA from researchers in external data sources may get different results. For example, as querying with the *fact value* “NM_003318" of Accession Number attribute in the NCBI Nucleotide database, the result entry of siRNA has a different GI number (“262399359") to (“34303964") the one provided by researchers. This is because the siRNA has a new version, GI Number “34303964" corresponds to Accession Number “NM_003318.3" which means the third version of the siRNA while GI Number “262399359" corresponds to Accession Number “NM_003318.4" which is the 4th version of the siRNA, stored in the NCBI Nucleotide database. Conflicts like this or conflicts like typos, e.g. the gene symbol given by the researchers might be miss-spelled or be using a synonym name, will be detected and presented to the user along with the best matches from the external data sources, thereupon the user can select one of the options presented or continue using his own version. It is possible to notice that the result of querying with some fact value from non-unique attribute may get non-unique results, e.g. searching the short sequence against BLAST+ application. Searching with Gene ID, Gene symbol and sequence in external data sources may get multiple results. These results will be treated as potential candidates. In order to detect duplications of siRNAs in the *master tables*, only the attribute *duplex number* is used due to its unique value.

4.4.2 The validation workflow

Theoretically, data sources can not be 100% accurate in describing all entities in the real world. Since the researches should be experts in the metadata that they upload, the researcher's decisions are involved as a part of the conflict resolving strategy. The validation result from *"Trust Your Friends"* strategy is "passed on" (*Pass it on* (BN06)) to other researchers to let them decide how to handle possible conflicts. For an entity, the validation result includes the status of its correctness and some possible solutions when some conflicts are detected in some attributes. For the entity, the *"Highest Quality"* (BN09) entries (in all attributes) obtained from external databases are given as recommended possible solutions to conflicts. The researchers have the final word on the conflict resolutions. For entities with only one unidentifiable attribute, additional fact values from external data sources should be used in the validation strategy. For entities with several attributes, some multi-objective decision algorithms should be implemented during the selection of the *"Highest Quality"* options. The validation workflow follows *"Trust Your Friends"* and *"Pass It On"* strategies while using *"Levenshtein distance"* and *"Multi-Objective Decision"* algorithms. There are two branches separately focusing on single-attribute and multi-attributes situations in the validation workflow. The two branches follow a common principle of the validation workflow. The principle is first parsing each fact value, i.e. the attribute value of metadata from researchers, into a standard unique identifier value by querying it as a keyword in an external data source, then getting the entries from an external database, considered as reliable, which uses the unique identifier as a primary key. The validation of *Compounds* and the validation of *siRNAs* can be viewed as two scenarios corresponding to the two branches of the workflow, respectively.

Compound validation

The whole workflow can be divided in four stages: (1) *Get candidates*(c.f. Figure 4.2), (2) *Screening and marking duplex* (c.f. Figure 4.3), (3) *Updating duplex marks* (c.f. Figure 4.5) and (4) *Cleaning up the validation result table* (c.f. Figure 4.4). When the user wants to insert or update a compound into the master table, the process starts to check whether the compound name exists in the master table. If so, the name will not be stored. Otherwise, the process will call the components for validation in this order, i.e. *get candidates → screening → update duplex marks*. If the user wants to ignore the validation results and keep the compound in the master table as it is, the user can select "ignore" which will call the "*clean up*" component. If the user wants to delete the compound from the master table, then after calling the "clean up" component, the deleting action in the master table will follow.

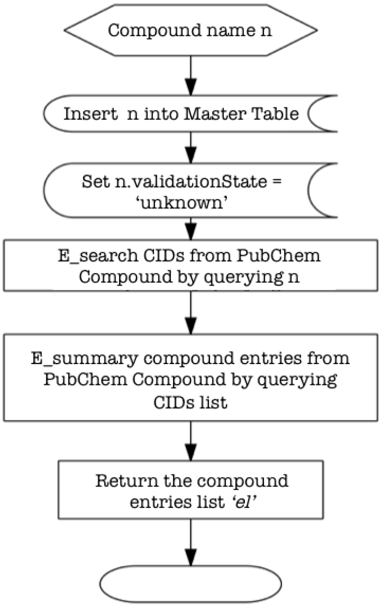

Compound name n
Insert n into Master Table
Set n.validationState = 'unknown'
E_search CIDs from PubChem Compound by querying n
E_summary compound entries from PubChem Compound by querying CIDs list
Return the compound entries list 'el'

Figure 4.2: *Get candidates*

siRNAs validation

This is the case for validating multi-attributes entities. 5 types of siRNA attributes can be validated with external data sources. They are: the *Gene ID*, *the Gene Symbol*, the *Accession Number*, the *GI number* and the *sequence*. Since the *Duplex Number* (it is only used on the master table as a unique identifier which cannot be validated with external data source) attribute is registered on each siRNA entry in the master table, only a basic duplex validation of siRNAs is adopted, i.e. assuming that the Duplex Number of the siRNA provided by the user is reliable, then only checking if the duplex number exists in the master table is enough. To do the validation, not all the 5 kinds of attribute values can be directly used as input fields for query in external data sources. Meanwhile, not all the 5 attributes are included in the query output for each external data source. The supported types of attributes for query input/output in external data sources are listed below in Table 4.4.

As shown in Table 4.4, each data source has some specified input fields (e.g. only BLAST+ application can search with a sequence and only HGNC databases can search with Gene ID/symbol, etc.). And not all five fields are available in the output siRNA entities (in fact only NCBI Nucleotide database support all fields in the output). However, all these data sources do have a common field, "Accession number", in the output. The Accession Number or also called "*GenBank Accession Number*" is a unique identifier

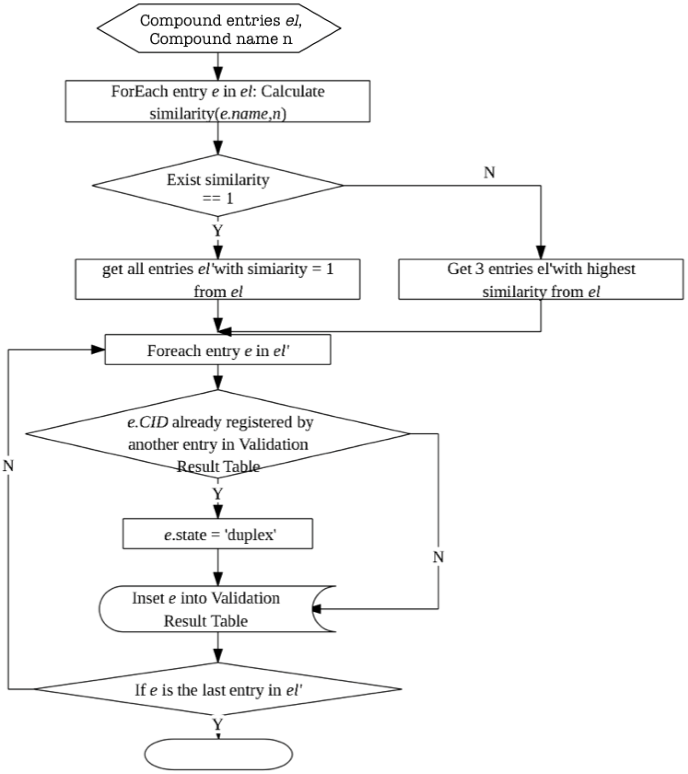

Compound entries *el*,
Compound name n
ForEach entry *e* in *el*: Calculate similarity(*e.name*,*n*)
Exist similarity == 1
N
Y
get all entries *el'*with simiarity = 1 from *el*
Get 3 entries el'with highest similarity from *el*
Foreach entry *e* in *el'*
e.CID already registered by another entry in Validation Result Table
N
Y
e.state = 'duplex'
N
Inset *e* into Validation Result Table
If *e* is the last entry in *el'*
Y

Figure 4.3: *Screening and Marking*

Table 4.4: *Supported types of attributes in external Data Sources*

External Data Source	Supported types of attributes for query	Query output field
HGNC IDs	GeneID	GeneID, GeneSymbol, Accession Number
HGNC Symbols	GeneSymbol	GeneID, GeneSymbol, Accession Number
NCBI Nucleotide	Accession Number, GINumber	Accession Number, GINumber, GeneID, GeneSymbol, sequence
BLAST+	Accession Number, GINumber, sequence	Accession Number, GINumber, sequence

given to a DNA entity record to track versions and associated entities over time of the entity record in a data repository (BKMC+12). A standard example of an accession number in Table 4.3 is “NM_003318.4" (MKSP00). It is a combination of an accession prefix (“NM_003318") and a version number (“4"). If the sequence of the DNA entity changes, the accession prefix will remain the same but the version number will be

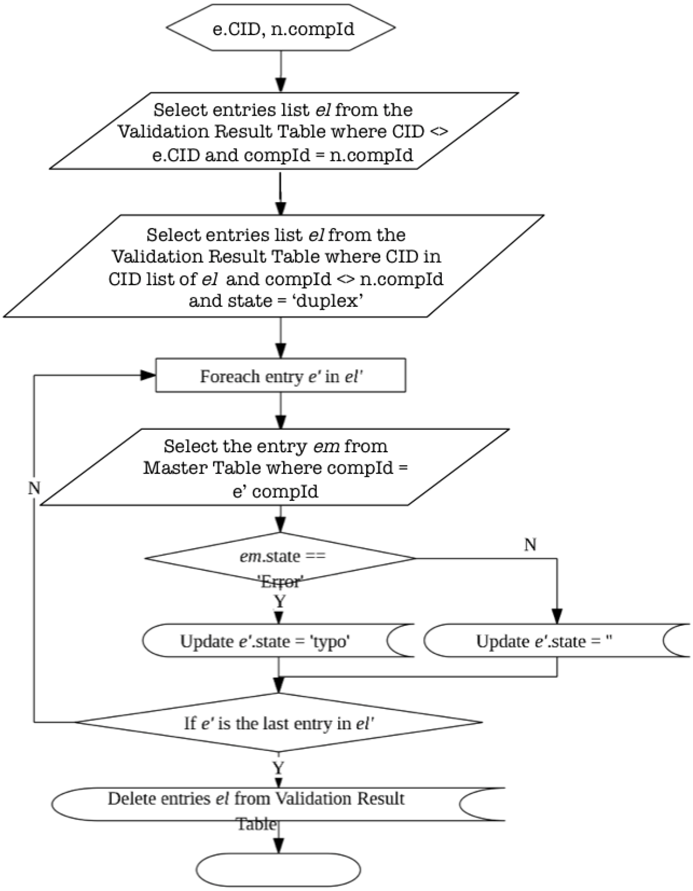

e.CID, n.compId
Select entries list *el* from the Validation Result Table where CID <> e.CID and compId = n.compId
Select entries list *el* from the Validation Result Table where CID in CID list of *el* and compId <> n.compId and state = 'duplex'
Foreach entry *e'* in *el'*
Select the entry *em* from Master Table where compId = e' compId
em.state == 'Error'
N
Y
N
Update *e'*.state = 'typo'
Update *e'*.state = ''
If *e'* is the last entry in *el'*
Y
Delete entries *el* from Validation Result Table

Figure 4.4: *Clean up*

incremented. GenBank GI number, however, will change each time the sequence changes – even if only one base is affected. So the *accession number* is used as a common identifier in the validation process.

An example of a user-provided siRNA is given in Table 4.5. Attribute values from *"Oder Number*", *"Pool Catalog Number*" and *"Duplex Number*" are not possible to be validated with external data sources as they are only defined and used internally in the laboratory. So they are not considered in the validation process. In Table 4.5, these fields are in gray.

In the first step to validate this siRNA, each attribute in the entity is parsed by a separate parser. The multi-attributes validation problem is then transformed into a single-attribute validation problem. Similar to the the Compund validation, the first step of the validation process is to parse each attribute into an unique identifier

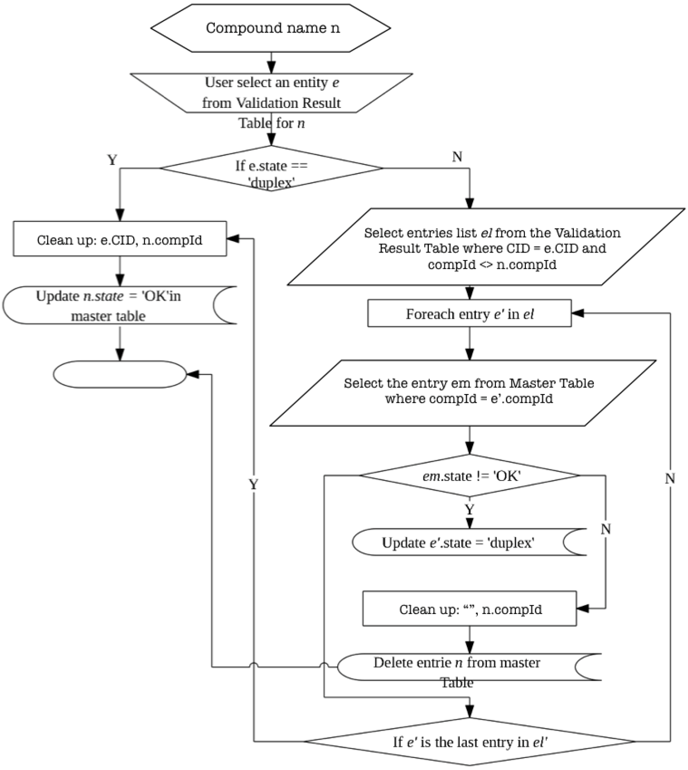

Compound name n
User select an entity *e* from Validation Result Table for *n*
Y
If e.state == 'duplex'
N
Clean up: e.CID, n.compId
Select entries list *el* from the Validation Result Table where CID = e.CID and compId <> n.compId
Update *n.state* = 'OK'in master table
Foreach entry *e'* in *el*
Select the entry em from Master Table where compId = e'.compId
Y
em.state != 'OK'
N
Y
N
Update *e'*.state = 'duplex'
Clean up: "", n.compId
Delete entrie *n* from master Table
If *e'* is the last entry in *el'*

Figure 4.5: *Update duplex marks*

Table 4.5: *siRNA example uploaded by the user*

Gene ID	7272
Gene Symbol	TTK
Order Number	191376
Pool Catalog Number	D-004105-01
Accession Number	NM_003318
GI Number	34303964
Duplex Number	D-004105-01
Sequence	P.M.

i.e.,"*Accession Number*" by using external data sources. The list of Accession numbers obtained varies from database to database according to the filters . For instance, the accession number returned from the "*HGNC* database omits the version suffix number. For a given "*Accession number* without a suffix number queried in the NCBI Nucleotide database (i.e. “NM_003318"), the returned *accession number* is always the latest version. The Accession number provided by the scientist is usually without suffix version. The result of the first stage is listed in Table 4.6.

Table 4.6: *List obtained after first stage of siRNA validation*

Input field and value	Parsed to	External data source
Gene ID: 7272	NM_003318	HGNC Gene ID
Gene Symbol: TTK	NM_003318	HGNC Gene Symbol
Accession Number: NM_003318	NM_003318	NCBI Nucleotide
GI Number: 34303964	NM_003318	NCBI Nucleotide
Sequence: P.M.	XM_008969441 NM_003318 NM_001166691	BLAST+

The duplicated results are omitted from the list before sending the list to the next step. If there is some inconsistence between the siRNA entry provided by the user and the siRNA entry found in external data source, it is due to the version update. This means there is a comparatively higher chance to find a perfect match in one of the versions of the siRNA. So the next step is to find all existed version suffixes for accession numbers retrieved from the first step. The RNAs associated to those accession numbers (with version suffixes) are used as candidates for the next step. The latest version of the accession numbers are fetched from the NCBI Nucleotide database by NCBI Eutilities EFetch web service (Say08). An iterator then is applied on the version suffix to get a list of accession numbers from version one to the latest one. For those accession prefixes in the first step, the list of associated accession numbers is showed in Table 4.7.

Table 4.7: *List obtained after second stage of siRNA validation*

Accession numbers	Latest version
NM_003318.1 NM_003318.2 NM_003318.3 NM_003318.4	4
XM_008969441.1	1
NM_001166691.1	1

Using these accession numbers, queries are executed in the NCBI Nucleotide database obtaining the results displayed in Table 4.8.

Some candidates have several names in Gene Symbol attribute (e.g. “TTK; ESK;

Table 4.8: *List of siRNAs candidates retrieved from the NCBI db.*

Accession numbers	GI numbers	Gene ID	Gene Symbol	Sequence
NM_003318.1	4507718	7272	TTK; MPS1L1	P.M.
NM_003318.2	23308721	7272	TTK; ESK; MPS1L1; PYT	P.M.
NM_003318.3	34303964	7272	TTK; CT96; ESK; FLJ38280; MPS1; MPS1L1; PYT	P.M.
NM_003318.4	262399359	7272	TTK; CT96; ESK; MPH1; MPS1; MPS1L1; PYT	P.M.
XM_008969441.1	675737708	100969041	TTK	P.M.
NM_001166691.1	262399360	7272	TTK; CT96; ESK; MPH1; MPS1; MPS1L1; PYT	P.M.

MPS1L1; PYT" for the "NM_003318.2" entry). This is because one siRNA can have several synonym names. When candidates are retrieved, all these synonym names will be collected and linked to the official gene symbol in the *Gene Symbol* field. Subsequently, if the siRNA provided by the researchers use a synonym name, the validation process is able to detect it and give a warning to the user.

The fourth step is to use several comparators in order to calculate the similarity of each attribute between the candidate and the siRNA provided by the researcher. The result of this step is a matrix of size $n \times 5$ where *n* is the number of candidates. Each row in this matrix corresponds to an entry of candidates and each column corresponds to each attribute. The similarity score is stored in each cell of the matrix. Given one candidate *sc*, the siRNA *s* from the researcher and the Levenshtein distance similarity grading function *ld (attr1: string, attr2: string)* (Lev66). The matrix obtained from the fourth step is displayed in Table 4.9.

Table 4.9: *Fourth step result - Matrix of similarities*

	GI numbers similarity	Gene ID similarity	Gene Symbol similarity	Accession Number similarity	Sequence Similarity
siRNA provided	Compare to: 34303964	Compare to: 7272	Compare to: TTK	Compare to: NM_003318	Compare to: Sequence provided
Candidate 1	0.0056	1	$< 1, 0 >$	1	1
Candidate 2	0.1051	1	$< 1, 0 >$	1	1
Candidate 3	1	1	$< 1, 0 >$	1	1
Candidate 4	0.0792	1	$< 1, 0 >$	1	1
Candidate 5	0.0792	0.07	$< 1, 0 >$	0.0056	1
Candidate 6	0.0903	1	$< 1, 0 >$	0.0056	1

The fifth step of the process is to use the multi-objective decision method (MA04) to screen best candidates and feed it back to the user for a decision. If a perfectly matched candidate is found then the user will get a positive feedback, for instance the candidate 3 in Table 4.9 scores 1 for all comparators except the Gene Symbol comparator. For the *Gene Symbol* comparator, other cells in the returned tuple is 1, this is a perfect match for the siRNA. If no perfectly matched candidates are found, then an error message along with the top 3 best matched candidates decided by the multi-objective decision method will be sent to the user. If the perfectly matched candidate is targeted, it is still needed to check if a new version of the gene name exists in an external database (e.g. the candidate 4 in Table 4.9) or if the candidate is using a synonym name (the value of the first cell in the Gene Symbol comparator returned tuple is less than 1 while the second cell value is 1). If so, a warning message will be sent to the user. The user can choose to ignore the solutions from the validation process or select one as the correct one. In both cases the decision is stored in a log where the Administrator can trace the changes performed in the *Master Table*.

4.4.3 The architecture

The architecture is designed to accurately follow the HTS experiment workflow described in section 2.2, considering four key activities: Acquisition, Visualization, Integration and Exploration. The acquisition consists of two steps: image acquisition and metadata formulation. The visualization is a key factor in this architecture due to the large amount of images produced in the HTS process. All these images should be linked to the plates designed in the acquisition activity. The integration is concerned with the tasks of image and data analysis which are performed by external applications. However, the output related to certain metadata is also included in the experiment. Therefor it is necessary to integrate external APIs to the architecture through web services. Finally, the exploration is associated to querying the data and the results. This task is also a key step in order to verify if the experiment requires a new iteration, possibly making some adjustments to the plate design.

Following the workflow of an HTS experiment described in Section 2.2, CytomicsDB uses a four-layered architecture (Fig. 2.6). The performance, stability, speed and scalability are main concerns for the architecture due to the overhead of connecting to external web services and loading a BLAST+ local sequence database into the RAM which are relatively heavy tasks during the validation process. Besides that, since the workflow relies on external applications (e.g. BLAST+ gets different I/O schema in Windows and Linux), it is needed to concern about the compatibility across different operation systems.

The validation process consists of four main activities: retrieving candidates, screening candidates, reporting inconsistences, and updating master tables according to user‘s decision. The four activities are distributed in several components which interact with each other. The component diagram in Fig. 4.6 shows the interaction between these components. These components operate within CytomicsDB using the four-layered architecture.

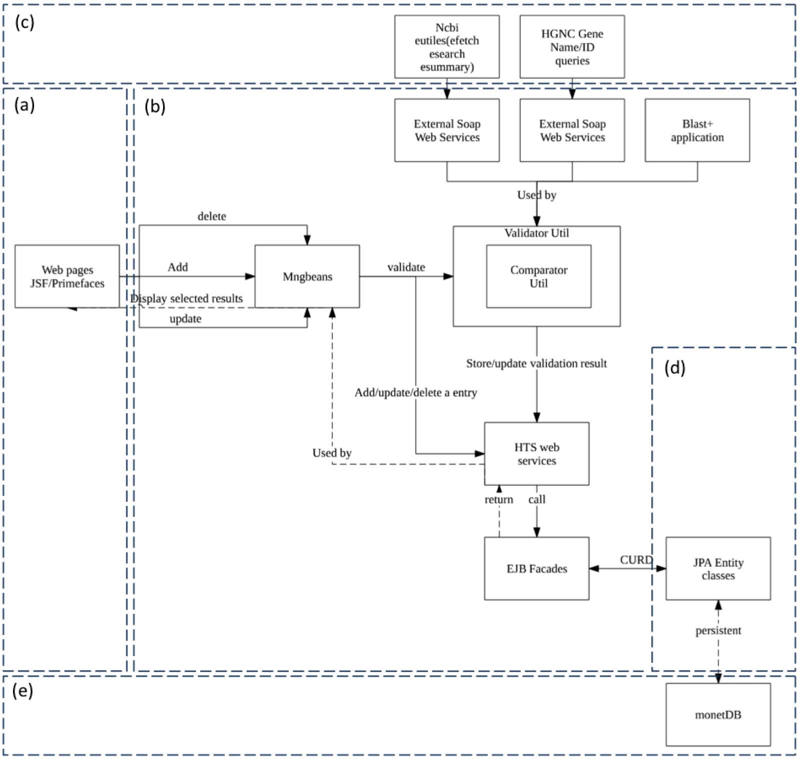
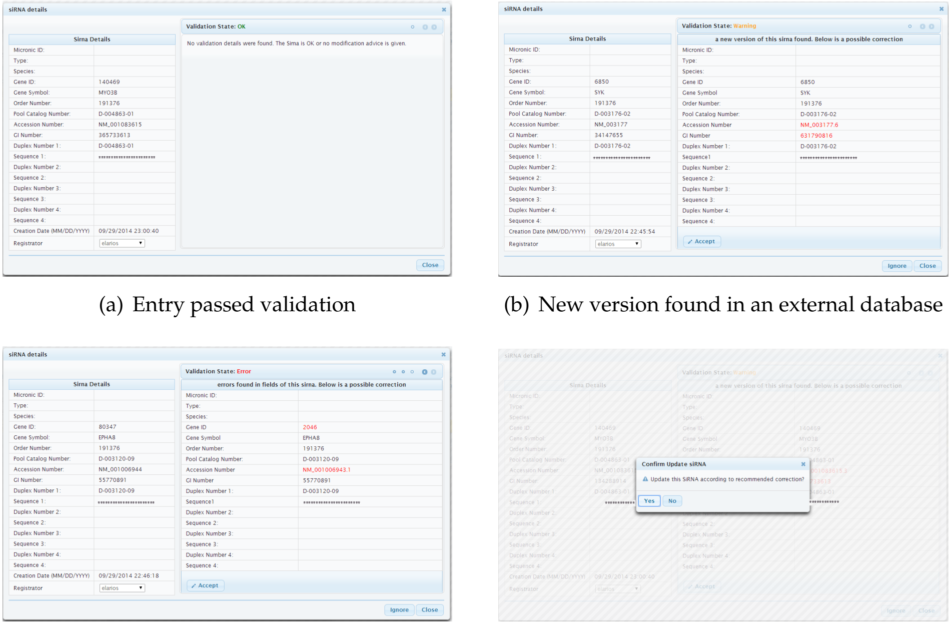


Figure 4.6: *Components diagram of the validation workflow*

The presentation layer

The validation process is functionally enabled for users using a single web based graphical user interface. The presentation layer supports the GUI for users. Coded in the JSF 2.0 and PrimeFaces 4.0 front end framework which fully support HTML5 and JavaScript/AJAX (see fig. 4.6(a)), the presentation layer makes it easier for users to interact with data in master tables (LZY$^+$12). Users can send requests to batch-upload a list of entries into the master tables, create or update a single entry in a master table, view, or delete one entry. The validation process will be triggered by the user operations on the presentation layer. During uploading or creating a new entry in the data repository, the validation process will be triggered to validate it at the back end. While the user is viewing the detail of one entry, the validation result and recommended solutions are showed synchronously. For instance, Figures 4.7(a), 4.7(b), 4.7(c), and 4.7(d) show the interfaces for the case of siRNA validation. Users with the right privileges can directly select one candidate the web interface to correct the detected inconsistency or keep the current one in the master table (see fig. 4.8). The selection will be updated into the data repository by the validation process. After the user has deleted or updated an entry from the master table, the validation process will be called to update the duplicated information in the validation result table if it is necessary. Besides that, the presentation layer is the first stage of validation in the platform by considering mandatory fields for uploading experiment metadata. The presentation layer also manages the messaging, thus the errors detected during the validation process will be reported to the users by an alert message on the interface.



siRNA details

Validation State: OK

No validation details were found. The Sirna is OK or no modification advice is given.

Sirna Details	
Micronic ID:	
Type:	
Species:	
Gene ID:	140469
Gene Symbol:	MYO3B
Order Number:	191376
Pool Catalog Number:	D-004863-01
Accession Number:	NM_001083615
GI Number:	365733613
Duplex Number 1:	D-004863-01
Sequence 1:	**********************
Duplex Number 2:	
Sequence 2:	
Duplex Number 3:	
Sequence 3:	
Duplex Number 4:	
Sequence 4:	
Creation Date (MM/DD/YYYY)	09/29/2014 23:00:40
Registrator	elarios ▾

(a) Entry passed validation

siRNA details

Sirna Details	
Micronic ID:	
Type:	
Species:	
Gene ID:	6850
Gene Symbol:	SYK
Order Number:	191376
Pool Catalog Number:	D-003176-02
Accession Number:	NM_003177
GI Number:	34147655
Duplex Number 1:	D-003176-02
Sequence 1:	**********************
Duplex Number 2:	
Sequence 2:	
Duplex Number 3:	
Sequence 3:	
Duplex Number 4:	
Sequence 4:	
Creation Date (MM/DD/YYYY)	09/29/2014 22:45:54
Registrator	elarios ▾

Validation State: Warning

a new version of this sirna found. Below is a possible correction	
Micronic ID:	
Type:	
Species:	
Gene ID	6850
Gene Symbol	SYK
Order Number:	191376
Pool Catalog Number:	D-003176-02
Accession Number	NM_003177.6
GI Number	631790816
Duplex Number 1:	D-003176-02
Sequence1	**********************
Duplex Number 2:	
Sequence 2:	
Duplex Number 3:	
Sequence 3:	
Duplex Number 4:	
Sequence 4:	

(b) New version found in an external database

siRNA details

Sirna Details	
Micronic ID:	
Type:	
Species:	
Gene ID:	80347
Gene Symbol:	EPHA8
Order Number:	191376
Pool Catalog Number:	D-003120-09
Accession Number:	NM_001006944
GI Number:	55770891
Duplex Number 1:	D-003120-09
Sequence 1:	**********************
Duplex Number 2:	
Sequence 2:	
Duplex Number 3:	
Sequence 3:	
Duplex Number 4:	
Sequence 4:	
Creation Date (MM/DD/YYYY)	09/29/2014 22:46:18
Registrator	elarios ▾

Validation State: Error

errors found in fields of this sirna. Below is a possible correction	
Micronic ID:	
Type:	
Species:	
Gene ID	2046
Gene Symbol	EPHA8
Order Number:	191376
Pool Catalog Number:	D-003120-09
Accession Number	NM_001006943.1
GI Number	55770891
Duplex Number 1:	D-003120-09
Sequence1	**********************
Duplex Number 2:	
Sequence 2:	
Duplex Number 3:	
Sequence 3:	
Duplex Number 4:	
Sequence 4:	

(c) Typo detected in the GeneID and Accession Number

siRNA details

Validation State: Warning

Confirm Update siRNA

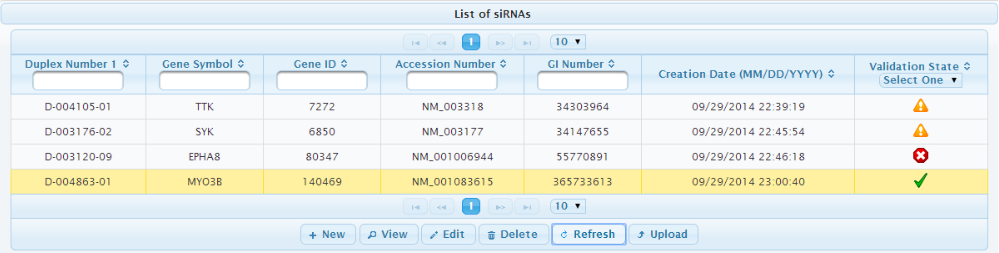
⚠ Update this SiRNA according to recommended correction?

Sirna Details	
Micronic ID:	
Type:	
Species:	
Gene ID:	140469
Gene Symbol:	MYO3B
Order Number:	191376
Pool Catalog Number:	D-004863-01
Accession Number:	NM_00108361
GI Number:	134288914
Duplex Number 1:	D-004863-01
Sequence 1:	************
Duplex Number 2:	
Sequence 2:	
Duplex Number 3:	
Sequence 3:	
Duplex Number 4:	
Sequence 4:	
Creation Date (MM/DD/YYYY)	09/29/2014 23:00:40
Registrator	elarios ▾

a new version of this sirna found. Below is a possible correction	
Micronic ID:	
Type:	
Species:	
Gene ID	140469
Gene Symbol	MYO3B
Order Number:	191376
	04863-01
	001083615.3
	733613
	04863-01
	************
Duplex Number 2:	
Sequence 2:	
Duplex Number 3:	
Sequence 3:	
Duplex Number 4:	
Sequence 4:	

(d) Confirm update

Figure 4.7: *Interfaces for validating a new siRNA entry*



List of siRNAs

Duplex Number 1 ⇕	Gene Symbol ⇕	Gene ID ⇕	Accession Number ⇕	GI Number ⇕	Creation Date (MM/DD/YYYY) ⇕	Validation State ⇕ Select One ▾
D-004105-01	TTK	7272	NM_003318	34303964	09/29/2014 22:39:19	⚠
D-003176-02	SYK	6850	NM_003177	34147655	09/29/2014 22:45:54	⚠
D-003120-09	EPHA8	80347	NM_001006944	55770891	09/29/2014 22:46:18	✖
D-004863-01	MYO3B	140469	NM_001083615	365733613	09/29/2014 23:00:40	✓

Figure 4.8: *Validation result interface*

The service layer

The service layer is described in Figure 4.6(b), it includes *manage beans* and several *utilities* such as parsers and comparators. They work as the pivot in the validation process to control the generation of candidates, the screening of candidates and the responding actions after the researcher makes a choice on the presentation layer. The *manage beans* are controllers which request to the *utilities* to visit resources from external web services (or applications) and do calculations (c.f. Fig. 4.6(c)). They also control the calling of internal web services in the service layer to do CRUD (create, read, update and delete) actions in the master tables. The results obtained are collected and sent back to the presentation layer (c.f. Fig. 4.6(a)). The purpose of designing the *utilities* as independent components from the *manage beans* is to make the parallel execution of the *utility* instances easier.

The service layer also consists of multiple web services that support every step in the HTS workflow. These web services invoke different APIs which are in charge of the Experiment design, Image Analysis and Data Analysis (LZY^{+}12). This structure allows easy scalability adding more functional modules. For instance, parsers in the *utility module* use web services to access external data sources. The Simple Object Access Protocol (SOAP) messages are selected for invoking the web services and receiving results because of its approved interoperability in web applications and heterogeneous environments. For these web services, one big portion of work is keeping the persistence in the database by using modules from the persistence layer (c.f. Fig. 4.6(d)).

The persistence layer

In section 2.3.3, it is described in detail the role of the persistence layer in CytomicsDB architecture. In the persistence layer is managed all the SQL code and configuration for accessing MonetDB database. This layer provides a common data model to the service layer. This style of architecture facilitates the flexibility and reusability of components in the web aplication.

The repository layer

The *master tables* are managed by an open source column-based database system, MonetDB (`www.monetdb.org`) (Bon02), which is used as the data repository (c.f. Fig. 4.6(e)). In the validation process three tables are needed . First, the *master table* which contains the object to validate, for instance: “sirna table" or “compound table"). Second, the *validation state* table, where the validation state (“OK", “Warning" or “error") for each entity is stored. Finally, the *validation details* table, where the solutions to correct the inconsistencies in each entity are stored.

MonetDB has proven to have superior performance in processing analytical queries on large scale data (BMK09) which is suitable for the complex data manipulation in the validation workflow. Thus, in CytomicsDB, MonetDB is used to store the experiment metadata and validation intermediate results.

4.5 Analysis of conflict solving strategies

In section 4.3 several conflict solving strategies were listed which are available to solve the data inconsistency issue while doing data integration or validation. This section explains how the analysis was performed in CytomicsDB for selecting the best strategie according to the metadata managed by the platform.

4.5.1 Criterias for selecting a strategy

According to (BN09), choosing a specific strategy for a particular data validation issue can be done by analyzing the following four aspects: (1) System availability, (2) Information availability, (3) Cost considerations and (4) Quality considerations.

System availability

The *"system availability*" constraints mainly come from the data properties. One of the major properties is the input/output value types accepted by the function that implements the strategy. The four most common types are: numerical, strings, categorical and taxonomical (**?**). The functions available are: (1) *Vote*, (2) *Average/Sum/Median*, (3) *First/Last*, (4) *Most Complete*, (5) *Most General*, (6) *Most Similar*, (7) *Choose Corresponding*, and (8) *Most Recent*.

The functions that can be used to implement each strategy are listed in Table 4.10.

Table 4.10: *Functions used to implement each strategy*

Strategies	Functions
Trust your friends	First/Last, Most similar, Most complete, Choose corresponding
Cry with the wolves	Vote
Meet in the middle	Average, Median, Most general, Vote
Keep up to date	Most recent, first
No gossiping	Not applicable

Each function has some preference on value types and data entry types. These preferences are listed in Table 4.11.

All attributes for siRNA and compound entries are strings, and the functions in each strategy should support both single and multiple-columns situations, thus some functions are not applicable for the test cases any more. Table 4.12 shows the list of available functions and strategies that are supported under the system constraints.

Table 4.11: *Conflict handling functions and their applicable properties*

Functions	Supported input/output types	Supported data entry types
Vote	All	Single-Column/Multi-Column
Average/Sum/Median	Numerical	Single-Column
First/Last	All	Single-Column
Most complete	All	Single-Column
Most general	Taxonomical	Single-Column
Most similar	String	Single-Column/Multi-Column
Choose corresponding	All	Multi-Column
Most recent	All	Single-Column/Multi-Column

Table 4.12: *Strategies and Functions available for siRNAs and Compounds*

Strategies	Functions	System availability
Trust your friends	Most similar	Implementable
Cry with the wolves	Vote	Implementable
Meet in the middle	Vote	Implementable
Keep up to date	Most recent	Implementable
No gossiping	Not applicable	Non Implementable

Information availability

There are three major nucleotide databases: GenBank (National Centre for Biotechnology Information, or also called *"NCBI*", *EMBL* (European Molecular Biology Laboratory) and *DDBJ* (The DNA Databank of Japan). All of them can be queried using SOAP based web services which their integration to the CytomicsDB platform makes more easy. Although these three data sources are available to all strategies, the strategies *CRY WITH THE WOLVES*, *MEET IN THE MIDDLE* and *KEEP UP TO DATE* may need more data sources in order to be implemented according to the principle of "more data sources the better result".

Beside these three databases, there are other databases which affiliate to smaller research groups. They may not have soap based web services which makes their integration to other systems more complex. Therefore, these data sources are more complex to be used by the strategies *CRY WITH THE WOLVES*, *MEET IN THE MIDDLE* and *KEEP UP TO DATE*.

In case of the siRNAs, the sequences can be queried locally in the web server using the BLAST+ application with its embedded database which is synchronized to the NCBI repository.

The smaller databases may have comparatively less maintenance and data are less reviewed than the other three major data sources. Since there are a lot of them, if they are included as valid data sources for *"CRY WITH WOLVES"* or *"MEET IN THE MIDDLE"* strategies under the principle of more data sources are better, then the list of candidates from all data sources will likely includes entries which do not fully correct describe the real-world facts.

The information availability for the five strategies is listed in Table 4.13.

Table 4.13: *Strategies and their information availability*

Strategies	Functions
Trust your friends	Available
Cry with the wolves	Less available
Meet in the middle	Less available
Keep up to date	Less available
No gossiping	Available

Cost considerations

CytomicsDB architecture supports multi-threading with minimized I/O (input/output) operations to all web services. Thus, this makes CytomicsDB comparatively light weighted. The cost considerations are calculated in terms of the overhead needed in order to decide the "correct" result for each strategy.

Costs for single-column data attributes The *VOTE* function or the *MOST RECENT* function does not require some particular complex algorithms and have $O_{(n)}$ as time and space complexity respectively, where n is the number of candidates. The *MOST SIMILAR* function can be implemented with other similar algorithms. The available algorithms for implementing the *MOST SIMILAR* function are:

1. *Longest Common Subsequences (LCS) algorithm*: The time complexity is $O_{(m_1 m_2 n)}$ where m_1, m_2 represent the length of the string and n is the number of candidates. An optimized implementation is built in The *BLAST* application for sequence similarity calculation (Cor88). The space complexity is expected to be $O_{(m_1 m_2 n)}$ as well.

2. *Levenshtein Distance (LD) algorithm*: Levenshtein (Lev66) proposed the edit distance algorithm which calculates the minimum numbers of operations (insertions, deletions and substitutions) for editing one string. The algorithm is sensitive to local changes in a single word. The time complexity for this algorithm is $O_{(m_1 m_2 n)}$

where $m1$, $m2$ are the length of strings and n is the number of candidates. The space complexity is also $O_{(m_1 m_2 n)}$.

3. *RKR-GST*: The major drawback of this algorithm is the time complexity which is $O_{(m^3 n)}$ (assuming the length of the two strings are almost the same value m) in the worst scenario where n is the number of candidates. RKR-GST uses a hash function to calculate the hash number for each division in the two strings. This optimized the complexity of GST to $O_{(m^2 n)}$. The space complexity of this algorithm is $O_{(n m_{max})}$ where m_{max} is the length of the longest string in the two strings.

Costs for multi-column data attributes When the *MOST SIMILAR* function runs in multi-column mode, it can be attached with different multi-objective decision algorithms as well. A multi-objective decision algorithm is used to model the decision-maker preferences. Algorithms are categorized depending on how the decision-maker articulates these preferences. Considering the overhead and easy-implementing factors, according to (MA04), in the class of algorithms with no articulation of preferences, the *Objective Sum Method* is one of the most computationally efficient, easy-to-use, and common approaches whose time complexity and space complexity are both $O_{(n)}$. Therefore, this work uses this approach during the accuracy test and implementation.

The calculation costs for different strategies in single and multi-column mode are listed in Table 4.14.

Quality considerations

The quality of each strategy is based on the accuracy of the functions used to implement the strategy. The quality is measured from two perspectives. First, whether the strategy is able to identify inconsistency in the data under validation. Second, whether the strategy-selected candidate really reflects the real world object. To do these evaluations, 300 Compounds and 300 siRNAs are used as test cases. 150 items in each category are randomly modified, forcing them to be invalid by adding, modifying or deleting one character or digit on each attribute value. To make these manual errors uniformly distributed, for the 150 modified compound name, every 50 names are modified by deletions, insertions or substitutions respectively. Similarly for the 150 modified siRNAs, they are divided into three groups (50 siRNAs in each group) and each group is modified by using one of the three different operations (deletion, insertion or substitution). In each group, the operation is performed on a different attribute (Gene Symbol, Gene Id, Accession Number, GI Number or Sequence) for every 10 entries. The *F-measure* which is a common measure for test accuracy is adopted as an indicator to identify the accuracy in finding inconsistency for each strategy. The F-measure considers both the precision p and the recall r of the test to compute the score.

$$p = \frac{tp}{(tp + fp)}$$

$$r = \frac{tp}{(tp + fn)}$$

$$F = \frac{(2)(p)(r)}{(p + r)}$$

The unmodified 150 consistent compound names or siRNAs are considered as the positive class, and tp, fp and fn denote the number of true positives, false positives, and false negatives, respectively. For the second criteria, a percentage number named "*True-hit*" is considered as the indicator. In this case, 600 (300 Compound names for the single-column case and 300 siRNA for the multi-column case respectively) proved, unchanged test cases are considered as the representation of real-world objects. Then the measure evaluates if each strategy yielded a "correct" candidate based on the 600 real objects (if not, it means that the "correct" candidate is not consistent with the real-world object). The percentage number shows the percent of matched ones in the 600 real-world objects. In the test, the NCBI database is used as the trustful data source for *TRUST YOUR FRIENDS* strategy. The NCBI database and EMBL database are used as the data sources for *CRY WITH THE WOLVES*, *MEET IN THE MIDDLE* and *KEEP UP TO DATE* strategies. As the *NO GOSSIPING* strategy does not work with the candidates retrieved from external data sources, it is not applicable for measuring accuracy in this case.

Table 4.14 shows the results of measurement on functions and algorithms in both single-column and multi-column mode.

4.5.2 Selection of strategies

The analysis for each strategy is listed in Table 4.14. In CytomicsDB, special care has been taken for considering the highest quality result with as little cost as possible. From the perspective of the quality, the larger value for *F-measure* and *True-hit* indicates the best quality of the result. According to Table 4.14: The average score for *F-measure* in a single-column mode for *TRUST YOUR FRIENDS* strategy is 0.963 which is larger than *CRY WITH THE WOLVES* (0.918) and *MEET IN THE MIDDLE* strategies, and also larger than *KEEP UP TO DATE* strategy (0.902).

The average score for *True-hit* in a single-column mode for *TRUST YOUR FRIENDS* strategy is 0.556 which is larger than *CRY WITH THE WOLVES* (0.513) and MEET IN THE MIDDLE strategies, and also larger than *KEEP UP TO DATE* strategy (0.507).

The average score for *F-measure* in a multi-column mode for *TRUST YOUR FRIENDS* strategy is 0.966 which is larger than *CRY WITH THE WOLVES* strategy (0.894), *MEET IN THE MIDDLE* strategy (0.136), and *KEEP UP TO DATE* strategy (0.355).

The average score for *True-hit* in a multi-column mode for *TRUST YOUR FRIENDS* strategy is 0.917 which is larger than *CRY WITH THE WOLVES* strategy (0.823), *MEET IN THE MIDDLE* strategy (0.053), and *KEEP UP TO DATE* strategy (0.193).

These results indicate that *TRUST YOUR FRIENDS* strategy with *MOST SIMILAR* function gets the highest values among all the strategies. Among all the algorithms in single-column mode, the *LD* algorithm gets 0.983 which is slightly higher than *LCS* algorithm (0.98) and *RKR-GST* algorithm (0.925) in *F-measure*, and it gets 0.573 which is higher than *LCS* algorithm (0.567) and RKR-GST algorithm (0.527) in True-hit measurement.

Among all the algorithms in multi-column mode, the *LD* algorithm scores 0.974 which equals to the score for *LCS* algorithm and is higher than *RKR-GST* algorithm (0.949) in *F-measure*, and it gets 0.933 in True-hit measurement which equals to the True-hit for *LCS* algorithm and better than 0.89 for *RKR-GST* algorithm.

These statistics show that *LD* algorithm has better quality results than the other two algorithms (*LCS* and *RKR-GST*) which are available for implementing the *MOST SIMILAR* function.

The drawback of *TRUST YOUR FRIENDS* strategy is the cost concern. For the space overhead, the memory use when the implementations are running for all strategies does not have any significant difference in a machine with 8 Gb of RAM. For the time complexity, *TRUST YOUR FRIENDS* strategy has the highest time among all strategies ($O_{(m^2n)} + O_{(n)} > O_{(n)} > O_{(1)}$). However, the complexity cost could be fetched up a little bit by using the architecture of CytomicsDB. A rough performance test shows, running with un-optimized *RKR-GST* algorithm (whose time complexity is $O_{(m^3n)}$) in a multi-threads pathway with minimized I/O (input/output) to all web services, it will take 200ms (not including the time for fetching candidates from external data sources) to validate one single-attribute entry or 600ms (not including the time for fetching candidates from external data sources) to validate a multi-attributes entry in a Linux system with a stable Internet connection. In the same environment, the time for fetching all the candidates from all the external data sources takes less than 1 second in a single-column mode and around 4 seconds in a multi-column mode for *CRY WITH THE WOLVES* strategy. So the calculation overhead is a comparatively small portion in the whole overhead for the validation process. Therefore, the time overhead for *TRUST YOUR FRIENDS* strategy is still in an acceptable range compared to its accuracy. As the validation process is performed during the design stage, once the metadata is uploaded and validated, changes on metadata will be very limited (i.e. "once created, use forever"). So, the overhead of one round validation can be a less important issue than its accuracy.

Another major concern is the information availability criteria. As shown in Table 4.14, only *TRUST YOUR FRIENDS* strategy (except *NO GOSSIPING* strategy) has full available data sources because theoretically it requires only one data source, it makes *TRUST YOUR FRIENDS* strategy more implementable than other strategies. The results for system availability are the same for all the strategies except *NO GOSSIPING* strategy.

Out of above considerations, the strategy *"TRUST YOUR FRIENDS*" implemented with the *MOST SIMILAR* function (using Levenshtein distance algorithm) is chosen for the validation process.

Table 4.14: *Strategies and criterias*

Strategies	**Functions**	**System availability**	**Information availability**	**Cost considerations**		**Quality considerations**			
				Time complexity	**Space complexity**	**Single column**		**Multi column**	
						F-meas.	**True-hit**	**F-meas.**	**True-hit**
Trust your friends	Most similar	Yes	Available	$O_{(m^2n)} + O_{(n)}$	$O_{(mn)} + O_{(n)}$	LCS: 0.98 LD: 0.983 RKR-GST: 0.925	LCS: 0.567 LD: 0.573 RKR-GST: 0.527	LCS: 0.974 LD: 0.974 RKR-GST: 0.949	LCS: 0.93 LD: 0.93 RKR-GST: 0.89
Cry with the wolves	Vote	Yes	Less available	$O_{(n)}$	$O_{(n)}$	0.918	0.513	0.894	0.823
Meet in the middle	Vote	Yes	Less available	$O_{(n)}$	$O_{(n)}$	0.918	0.513	0.136	0.053
Keep up to date	Most recent	Yes	Less available	$O_{(n)}$	$O_{(n)}$	0.902	0.507	0.355	0.193
No gossiping	—	No	Available	$O_{(1)}$	$O_{(1)}$	—	—	—	—

4.6 Conclusions and Future work

In this chapter, we have presented a semantic-based metadata validation approach for an automated High-Throughput Screening workflow. Our main goal is to ensure the integrity, consistency and reliability of the data stored in the platform. This is a critical requirement for image and data analysis and further data exploration of the experiment's results. CytomicsDB architecture has been designed to facilitate the integration to external repositories. The use of web services makes possible to have flexibility to access to external databases and validate the key metadata with this public repositories. Furthermore, aligning the metadata in CytomicsDB to public databases allow the platform to become an ontology based framework capable to handle semantic queries and turning the architecture to a web based interactive semantic platform for cytomics. Finally, we plan to optimize the recomendation results by tuning the multi-objective decision formula including weights associated to the user's previous decisions. Currently we are giving the same weight to all attributes during the metadata validation, but including user decision will allow the users to have more accurate candidates to select.

Chapter 5

Cluster Integration for Image processing and Pattern recognition

In this chapter is described how CytomicsDB supports the integration of cluster computing for the image analysis stage and how is performed the data processing and pattern recognition on MonetDB database. It contains also a detailed case study for two commonly used images analysis algorithms that have been adapted for efficient use on a computing cluster, and explore the effect on their performance.

This chapter is based on the following publication:

- E. Larios, D. van Es, K. Rietveld and F. J. Verbeek. **Cluster integration in CytomicsDB**. (Manuscript in preparation).

5.1 Introduction

In cytomics, the large-scale study of cell systems there is an urgent need to use high performance and parallel computing technologies due to the large volume of data managed in different type of experiments. One of the most used techniques in this area is High-Throughput screening (HTS) where thanks to the use of sophisticated microscopes a large set of cell images are acquired with taking pictures at a certain temporal interval. The resulting images can be studied individually, to observe cell characteristics such as morphology, or analysed as a time lapse series, to observe cell migration and motility for example.

It was mentioned in Chapter 1 that due to the complexity and variety of types of data managed in HTS experiments it was necessary to design an automated workflow capable of handling the data properly in each stage of the experiment (LZY$^+$12). The last 2 stages in HTS experiments: (1) *image analysis* and (2) *data analysis* receive as input a large volume of data, effectively it is physically impossible for a human examiner to go through every image and attempt to extract the required phenotypic measurements. Therefore, it is mandatory to use a computer-based environment for using image analysis techniques in order to ensure objective results.

In (YVDvdW09) and (YV12) are described two robust algorithms for image analysis customized for HTS studies. These algorithms were implemented using the Fiji software. This software is designed for the biologist and intended for a single user on a single machine using a sequential approach, interfacing through a GUI. For small tasks this is a proven setup, but it can prove impractical for high throughput experiments, with the large data sets often leading to long wait times and delays. A full analysis of one well plate can take two to three hours. Any possibility to speed up computation and decrease wait times is therefore highly desirable.

To support computation in the Life Sciences, at Leiden Institute of Advanced Computer Science (LIACS) has been built the Leiden Life Sciences Cluster (LLSC). This cluster consists of a fileserver, one user node and 24 worker nodes. This opens up possibilities to use a scalable and distributed environment for the Image Analysis stage. Additionally, the repository layer in the architecture of CytomicsDB relies on MonetDB database which provides a great platform for data analysis and visualization. The embedded MonetDB.R package serves as a powerful tool for data exploration and data mining.

In this chapter, we explore how CytomicsDB is integrated to the Cluster system, how the platform use the adapted image analysis algorithms on the LLSC and the performance of the resulting parallelized algorithms is also evaluated. Moreover, it is described the environment for data processing provided by MonetDB.

5.2 The Leiden Life Sciences Cluster

The Leiden Life Sciences Cluster (LLSC) is a computing cluster recently built at Leiden University. Its intended use is for research related to bioinformatics or other life sciences. All experiments conducted in this work have been performed on this cluster.

In order to run an application on the LLSC, first the user launches a job. The job contains information about the computer resources needed for its execution. e.g. amount of memory, cpus, etc. Moreover, it includes details about the process itself such as: name, input and output.

In order to guarantee efficient and effective use of the resources, a scheduler is in charge of the management of the jobs in the cluster. The main goals of the scheduler are: (1) allocation of computer resources, (2) job execution and (3) report to the user the output of the execution.

To analyze large datasets, it has become typical to use clusters of machines to execute jobs consisting of many tasks. Jobs of many applications coexist on these clusters and their tasks have diverse resource demands (GAK^{+}14).

The LLSC consists of:

- A single user node, or head node, running the scheduler, i.e. TORQUE (Sta06).
- 24 worker nodes with varying configurations:
  - 13 nodes with two dual-core Xeon 5150 CPUs and 16 GB main memory.
  - 9 nodes with two quad-core Xeon E5430 CPUs and 16 GB main memory.
  - 2 nodes with two dual-core Xeon 5150 CPUs and 8 GB main memory.
  - All nodes have 400 GB of local storage in a hardware RAID-0 configuration.
  - All nodes are interconnected with 100 MBit/s network interfaces.
- 2 file servers with 7.5 TB as temporary storage in hardware RAID-5 configuration, 32 GB main memory and connected to the network with a speed of 1 GBit/s.

The user node runs the TORQUE Resource Manager. All experiments are run through TORQUE as jobs. TORQUE is responsible for managing resources, the most important of which are the allocation of nodes and scheduling of jobs. TORQUE allows features such as requesting resources, tagging nodes, advanced job logging and statistics, job arrays and easy integration with third party parallel computing solutions such as MPI (Message Passing Interface).

5.3 Image Analysis

The experiments in this chapter utilize a standard dataset in cytomics and two Image Analysis algorithms: (1) Watershed Masked Clustering (YV12) and (2) Kernel Density Estimation (YVDvdW09), which are image segmentation and object tracking methods, respectively (CYW^{+}11).

The images obtained from the image acquisition phase are digital images, and as such they can be processed using digital image processing techniques. The purpose of the experiments is to measure phenotypic properties. Since phenotype is defined as the observable characteristics of some object, having some way to determine the boundaries of that object is critical. The success of both algorithms is highly dependent upon this. One digital image processing technique designed for this purpose is image segmentation.

Segmentation is the process of separating an image into its constituent parts or objects. Usually this means separating the background from the foreground. In our image samples this is also the case. Each pixel belonging to a cell in the image is considered part of the foreground, and all other pixels are background. Segmentation is considered one of the most difficult image processing tasks (GW06), and also one of the most important since a lot of subsequent processing techniques are dependent on the output of the segmentation phase. Separation of foreground and background is key to further decomposing the foreground into an accurate collection of object masks that represent individual cells.

It is important to note that the segmentation phase does not have to take the raw images directly from image acquisition. Segmentation is usually preceded by an Image enhancement phase. In this phase imperfections in the image such as too much noise or low contrast can be adjusted to make the image suitable for the segmentation tasks. Our algorithm makes use of enhancement techniques such as subtracting the background, gaussian blur, noise suppression and contrast enhancement. There are numerous popular segmentation techniques, and each comes with their own strengths and weaknesses. The choice of which technique to use largely depends on the composition of the target image.

Segmentation methods generally operate on one of two basic properties of intensity values: discontinuity and similarity. The former uses abrupt changes in an image to partition the image. The latter attempts to find regions of the image that are similar in pixel value to each other. Abrupt changes are usually edges of objects. For similar regions, the notion similar must first be defined. Put simply, for our input images obtained by fluorescence microscopy, similar pixels have close intensity values to neighbouring pixels above a certain threshold. The end result is a binary image. Each pixel was assigned intensity value 1 if it belongs to a cell, and 0 otherwise. The resulting image is referred to as *the mask*.

The segmentation step is followed by object tracking. Object tracking algorithms find links between objects over a time lapse series. This information can be used to study the movement of these objects over a period of time. In our case, a link must be found between two objects that appear in consecutive images. Both algorithms are described in detail in (Yan13).

5.4 Data Analysis and Pattern recognition

In (Ber03), data mining is defined as the process of identifying patterns and relationships in data that often are not obvious in large, complex data sets. As such, data mining involves pattern recognition and, by extension, pattern discovery.

Bergeron (Ber03) also identifies five major steps in the pattern recognition and discovery process: (1) *Feature selection*, (2) *Measurement*, (3) *Processing*, (4) *Feature extraction*, and (5) *Classification and discovery*.

Feature Selection. From the universe of available features, the selection of a set of features or attributes is considered the first step in pattern recognition.

Measurement. The measurement phase involves converting the original pattern into a representation that can be easily manipulated programmatically.

Processing. After the measurement process, the data are processed to remove noise and prepare for feature extraction. Processing typically involves executing a variety of error checking and correction routines, as well as specialized processes that depend on the nature of the data.

Feature Extraction. Feature extraction involves searching for global and local features in the data that are defined as relevant to pattern matching during feature selection. Clustering techniques, in which similar data are grouped together, often form the basis of feature extraction.

Classification and Discovery. In the classification phase, data are classified based on measurements of similarity with other patterns. These measurements of similarity are commonly based on either a statistical or a structural approach.

5.5 Implementation

Currently there are many open source software tools which assist researchers to perform complex image analysis processes in HTS experiments. For that reason it is relevant for systems that manage experiments data to facilitate the integratation of those tools to their architecture. CytomicsDB has been designed to assist the whole HTS workflow (YLL+11), thus the tools used during the analysis stage are easily integrated to our platform through web services. In this section, the architecture will be described as well as how the interaction between the components in the architecture are synchronized during the analysis stage.

5.5.1 The architecture

In Chapter 2 CytomicsDB's architecture was introduced, it has a four-layered style architecture: (1) *Presentation layer*, (2) *Service layer*, (3) *Persistence layer* and (4) *Repository*. This design allows scalability and flexibility thanks to the easy integration of external systems to the platform. In this particular case the integration of the cluster LLSC, built at Leiden University for assisting researchers in the execution of complex tasks during the image and data analysis where the large volume of data involved demands high computational resources. In this section the architecture is described from the image analysis point of view.

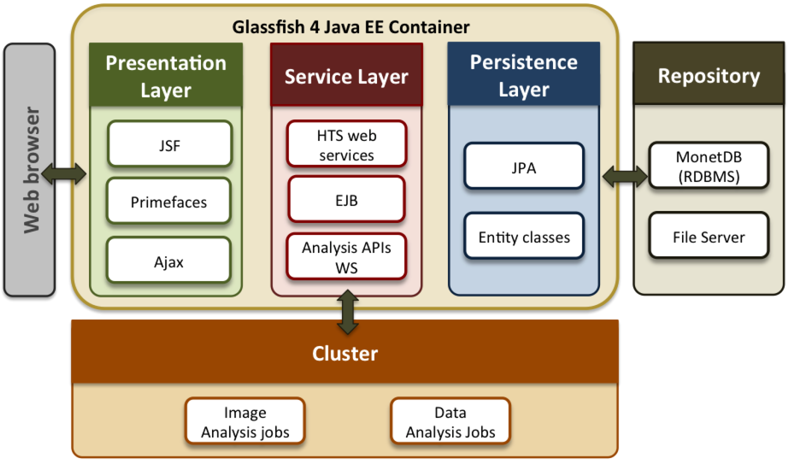
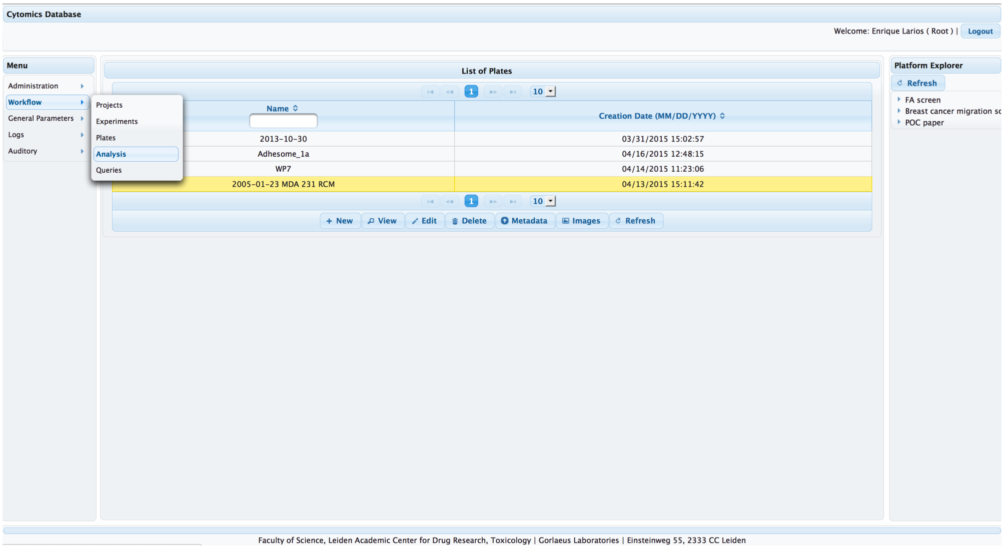


Figure 5.1: *CytomicsDB Architecture - Cluster Integration*



List of Plates

Name	Creation Date (MM/DD/YYYY)
2013-10-30	03/31/2015 15:02:57
Adhesome_1a	04/16/2015 12:48:15
WP7	04/14/2015 11:23:06
2005-01-23 MDA 231 RCM	04/13/2015 15:11:42

Faculty of Science, Leiden Academic Center for Drug Research, Toxicology | Gorlaeus Laboratories | Einsteinweg 55, 2333 CC Leiden

Figure 5.2: *Web Interface for Analysis*

The presentation layer

According to the HTS workflow, upon complexion of the image acquisition step and after uploading the images obtained to CytomicsDB, the analysis process can be triggered just by selecting a plate in the web interface and then selecting the option for analysis (Fig. 5.2). This layer also manages the web pages for visualizing the results, basically phenotype descriptors, binary masks and trajectories.

The service layer

This layer includes the web services in charge of the management of the analysis process. This style provides security and it is also a key factor in the integration with external systems. In this case the integration with the cluster LLSC is straightforward and independent of the programming language or the operating system running in each component of the architecture.

The persistence layer

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 used in CytomicsDB map a physical table from the database and are the most secure way to manipulate and query the data stored in MonetDB.

The repository layer

The image analysis results consists basically of: (1) phenotype data, which is parsed to the relational database management system and linked to the original image dataset. (2) Binary masks and Trajectories, these results are also images that are stored in the File Server, and their location stored in MonetDB (Bon02) (LZCV14).

The Cluster

In Section 5.2 it was introduced the hardware used in the LLSC, the following describes how it works for the image analysis stage, in particular case for the segmentation and object tracking.

The input stacks are obtained from the Database. There are three main components involved. Two scripts written in Python named PA.py and PA-Seg.py, to setup the tracking and segmentation experiments respectively. Two TORQUE/PBS jobscripts using Bash called parallel-tracking.jobscript and parallel-segmentation.jobscript. Finally, a jar file named PA.jar containing the modified ImageJ source and Java implementations for segmentation and tracking. The ImageJ source is modified so that GUI components of certain plugins are no longer instantiated. The call hierarchy is `PA.py` → `jobscript` → `PA.jar`. PA stands for Phenotype Analysis. PA.py takes a text file as its argument. This text file itself contains arguments for the rest of the process. PA.py creates a global output directory for the job. Each node has read/write privileges for this directory. It copies a file containing the locations of all input stacks to this directory. It also copies the jobscript, depending on whether it is a tracking or segmentation job. The script also passes along any relevant arguments to the jobscript. The last statement in both Python scripts executes qsub, which is a TORQUE command and instructs TORQUE to schedule the jobscript. The jobscript submitted to the cluster contains a header. In this header the output directory can be specified for all job output and logfiles. In our case this is the global output directory. These logfiles can later be used to gather job statistics. Job resources can also be requested. It is possible to request only nodes of a certain type or with a certain amount of memory. The main feature we are interested in is the amount of nodes, and the processors per node. Using these two attributes we can control the total number of resources available per job and measure how total job execution time reacts to different combinations of nodes and processors per node. The jobscript controls a selected number of nodes and is responsible for calling PA.jar on each node with the correct parameters. For segmentation, each node is given an equal number of stacks to process. If there is a remainder it is again split over the nodes. Once the node "knows" which stacks it must segment it can run concurrently with no synchronization until it is finished. For tracking, the jobscript first calculates the input arguments which it writes to a file on each node. Again, each node can process its stacks concurrently. When all nodes are finished the joining operation is started. PA.jar takes different arguments for segmentation and tracking. For segmentation it takes the input stack, the number of cores to use and the location of the output directory. Tracking takes the input stack, the masked input stack, the output directory, the number of cores, and the id's of the slices it must process.

MonetDB.R Package

R programming language has become for both academia and industry one of the most important tools for data analytics, statistics, visualization and data science. Scientist use R to solve their problems for data processing specially with respect to large volumes of scientific data. R is also becoming important because it is not only very flexible in reading, manipulating, and writing data but all its outcomes are directly available as objects for further programming (Kri09).

One of the challenges of data management in Cytomics mentioned in Chapter 1 is *the movement of data*, and the need for processing the data "in place" and transmit only the resulting information. Following this paradigm, MonetDB.R package was developed for providing a transparent connection to MonetDB from the R software suite. MonetDB.R was designed to reduce the overhead of shuffling data between different systems, resulting in the improvment of data processing time. This is achieved because the package decides which portions of the data analysis should be performed by either R or the MonetDB database. Thus, the combination of both applications in CytomicsDB platform speed up the process of data discovery (Mon).

5.5.2 The image and data analysis data flow

The diagram shown in Figure 5.3 illustrates the flows of the control in the HTS image and data analysis stage. In case of the image analysis, Figure 5.3 shows the steps for the execution as typical cases of the segmentation and object tracking algorithms. On

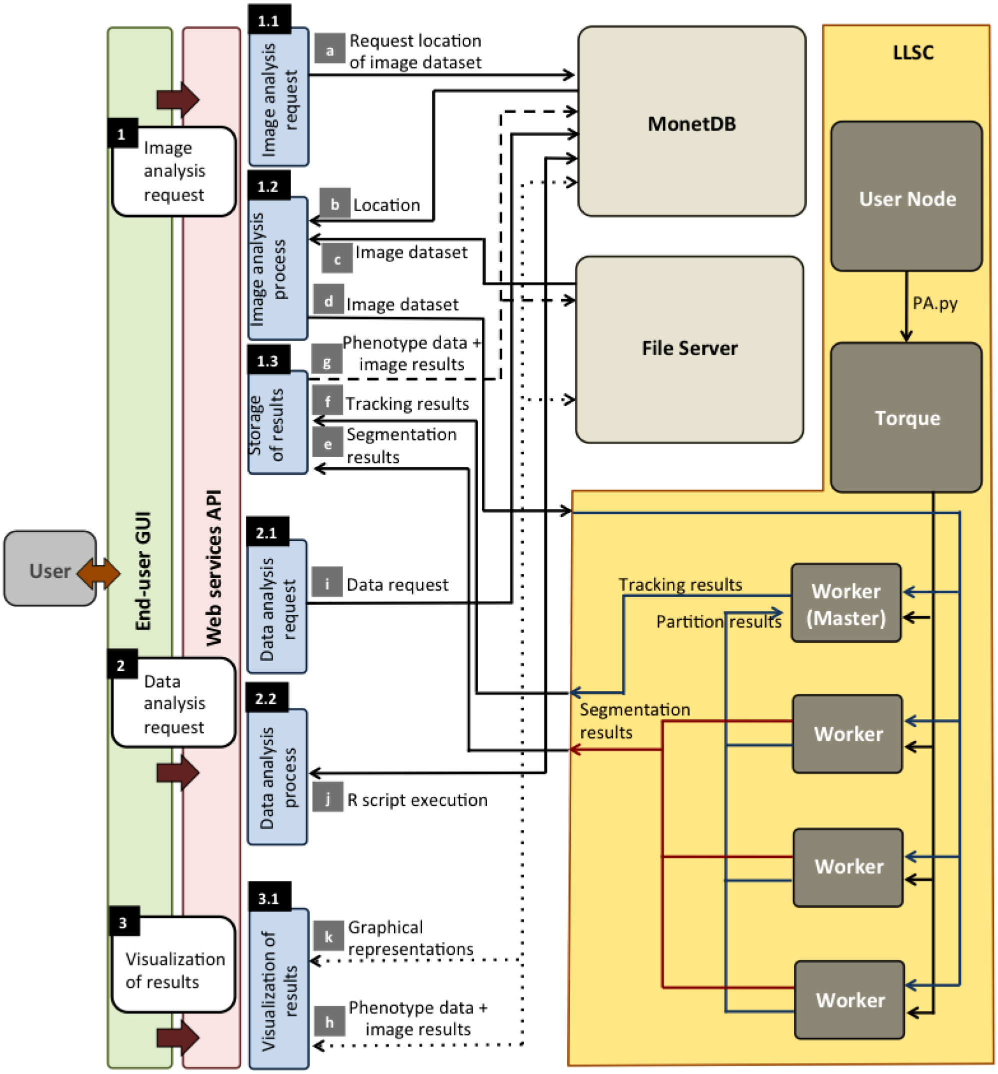


Figure 5.3: *Image and Data analysis data flow*

the other hand, in case of the data analysis, it is shown the steps for the execution of the R scripts in order to obtain graphical representations of the large datasets and identify patterns. The main three features of this stage are (1) *the image analysis request*, (2) *the data analysis request* and (3) *the visualization of the results*. How this main features are executed is shown by three sequences of annotated arrows starting from the end-user GUI. Arrows handling the same operation are grouped together by a major number, while the alphabetical characters correspond to the order of a particular step that is called in its containing sequence.

In Figure 5.3, we describe each of the sequences. Sequence 1 handles the image analysis request, this request is made from the web interface in CytomicsDB. First, the request is sent to MonetDB, the location of the image datasets under analysis is sent to the web service in charge of the management of the analysis process. Second, the image dataset is extracted from the File Server and then sent to the LLSC for the execution of the segmentation and then object tracking tasks. Finally, upon the complexion of the analysis process the phenotye data is stored in MonetDB and the new image datasets resulted are stored on the File Server, then their location is stored in MonetDB for further querying. Sequence 2 handles a data analysis request, which is first sent to MonetDB. Subsequently, the data analysis process is triggered. R scripts are executed using the package MonetDB.R, this package will manage which part of the data analysis will be performed in R or in MonetDB database. Sequence 3 handles the visualization of the results, since the results and experiment's metadata is stored in one single place, the visualization of the results can be handled by just requesting the data of the analysis results from MonetDB. In the web interface, the results are displayed with the corresponding plate layout.

5.5.3 Image Analysis setup and Results

Segmentation

This experiment measures job time for an input set of 512 stacks, processed over 1,2,4 or 8 nodes. The images are MTLn3 cancer cells, cultured in 4 different well plates. The 512 stacks are made up of:

- 144 images from well plate 2, not treated.
- 144 images from well plate 2, treated with EGF.
- 144 images from well plate 3, not treated.
- 88 images from well plate 3, treated with EGF.

This experiment utilizes all available nodes and was run 3 times. The results reflect the average of these runs. There are also an equal amount of jobs as nodes. Figure 5.4 shows how segmentation scales for up to 24 nodes, confirming a linear speedup.

Object tracking

The experiment input is 144 non treated image stacks from well plate 2, and their corresponding masked versions obtained through prior segmentation. Each stack has 31 slices. This experiment processes partitions of a stack concurrently but does not process multiple stacks concurrently. Each image file is read directly from the File Server and tracked. Once all stacks have been tracked and have written their

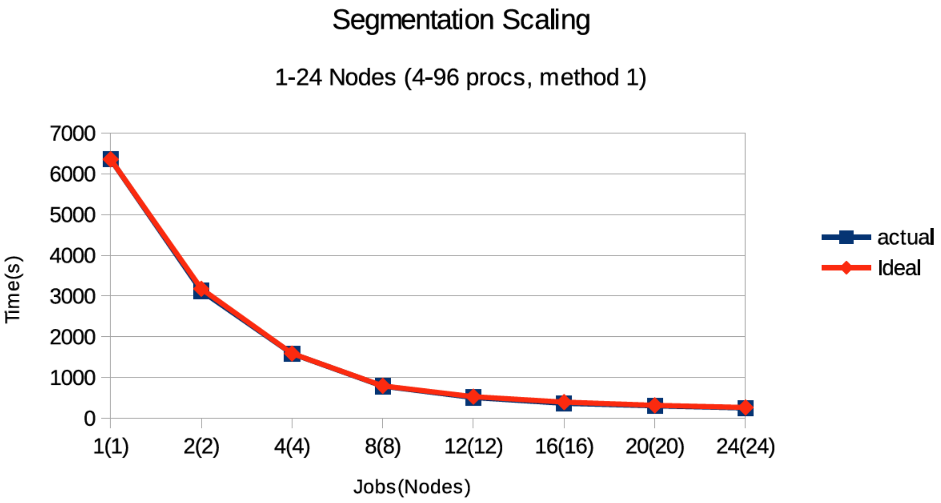


Figure 5.4: *Segmentation results using up to 24 nodes*

local results the joining starts. Each node sends its results to a master node where the joining algorithm is run for all stacks. When all algorithms are finished all results are copied back to the File Server. The tracking, joining and copy operations are all timed separately. The combinations listed in Table 5.1 were tested.

Table 5.1: *Algorithms tested*

Algorithm	Cores	Nodes
Sequential tracking	1	1
Partitioned tracking	1	2-10
Partitioned tracking	2	1-6
Partitioned tracking	4	1-3

Each experiment is run 3 times and the averaged results are given in figure 5.5. The line represents a linear (ideal) speedup. The chart shows that there is in fact a significant improvement even when more than 2 processors are used. In some cases a superlinear speedup is achieved. The use of 1 core per node consistently gives the best performance. A possible explanation for this is that 1 cpu is used for the algorithm, and in our implementation the entire node (4 processors) has been reserved by TORQUE. No other user intensive user processes were running on the remaining 3 cores. As a result, when 1 core per node is used the entire cpu cache and RAM memory can be used for tracking without interference. When 2 or 4 cores per node are used these resources must be shared, possibly causing increased cache miss rates. Using 4 cores per node is the slowest option (though it still achieves close to linear speedup). Even though it is outperformed, using 4 cores per node is still desirable. If one core per node is used and another user requests the remaining cores for another process performance may drop beyond that of using 4 cores per node. Even if no such request is made the remaining cores are essentially wasted. It is worth noting that using 7, 8 or 9 processors seems to be worse than using 6. This can be explained by the inefficiency of the slice allocator. Each node is allocated:

$$\frac{sizeofstack}{nrofnodes} + 1$$

slices, except the last which also processes the remainder. For 6 nodes, each node processes (31/6) + 1 = 6 slices. For 7 nodes, each node processes (31/7) + 1 = 5 slices, except the last which processes 8 slices. Similarly, 8 nodes each process 4 slices, except the last which processes 11. Each node must complete before the next stack can be processed. Therefore, when using 7,8 and 9 nodes since there is a node that processes more slices than when using 6 nodes, 6 nodes is faster. This method of slice allocation is the most basic easiest to implement method and should be improved in a future version.

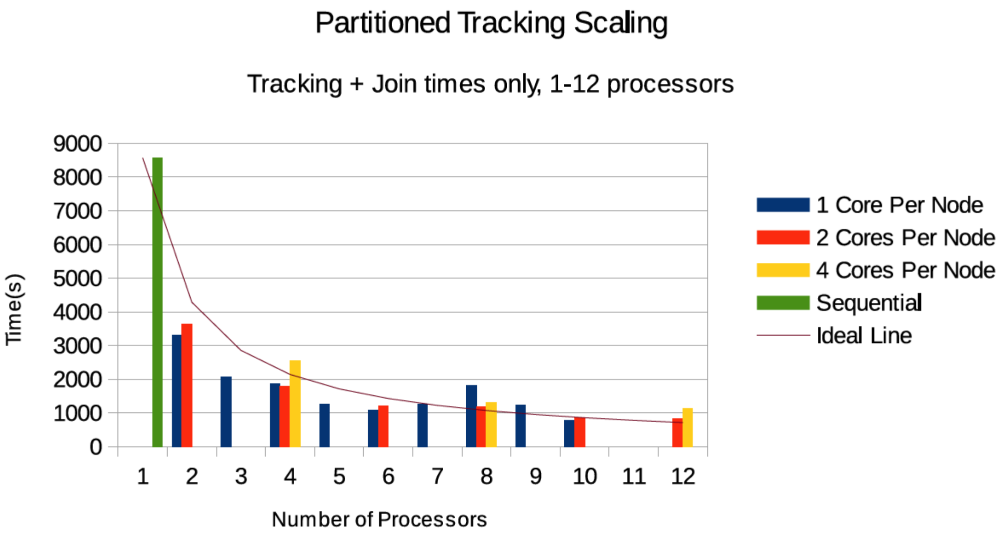


Figure 5.5: *Partitioned concurrent tracking and join times*

5.5.4 Data Analysis setup

In Section 3.3, it was described a case study which explains how the data model designed in CytomicsDB supports the image analysis stage. Tables *Feature* and *Measurement* are used to store the phenotype data resulted from the image analysis stage, and become the base for starting the data analysis process. In Figure 2.11, it was shown the the database schema for storing the measurements metadata. This section is based on the case study introduced in Section 3.3. The aim of the case study is to investigate the process of endocytosis and epidermial growth factor receptor (EGFR) signaling. EGFR signaling triggers breast cancer cells to escape from the primary tumor and spread to the lung, resulting in poor disease diagnosis. Moreover, it may result in resistance to anti-cancer therapy (CYW^{+}11).

Based on the data stored in the tables *Feature* and *Measurement*, the data analysis is triggered. First, it is requested from the database the feature names and the feature values used in this experiment.

The query executed for retrieving the feature names is:

```
SELECT f.feat_name
FROM HTS.Feature_plate p, HTS.Feature f
WHERE p.feat_id = f.feat_id and  p.plat_id = 17;
```

The *Features* are associated to a plate, and in this experiment is being used the plate identified by *plate_id* with value “17”. The result set obtained is:

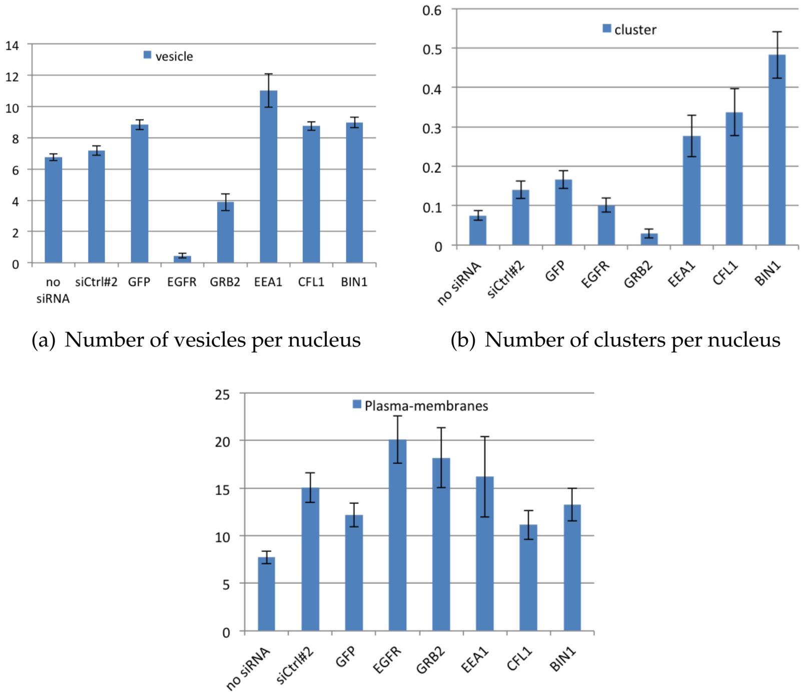
```
testNr, frame#, obj#, area, perimeter, massCenterX, massCenterY, extension, dispersion,
elongation, orientation, compactFactor, averageIntensity, Nucleolus Dist, Nuke X, Nuke Y,
Number of FAK, Number of Nucleolus, In Nucleolus, Closest FA Dist, long axis,short axis,
Border Distance, Int Std, Int Smoothness, Int Skewness, Int Uniformity, Int Entropy
```

Thanks to the use of the MonetDB.R package, there is no need to retrieve the feature values a.k.a. measurements from MonetDB to the web service layer in order to perform the data analysis. For this particular experiment, the table *measurement* contains 279 036 rows and 28 features.

A view of the features value stored in the table *Measurement* can be retrieved executing the following query:

```
SELECT f.feat_name, m.meas_value
FROM HTS.Feature_plate p, HTS.Feature f, HTS.Measurement m
WHERE p.feat_id = f.feat_id and p.feat_id = m.feat_id and
      p.plat_id = m.plat_id and p.plat_id = 17;
```

The data analysis process begins with the execution of the R script. This script is in charge of the following steps: (1) *Classification* and (2) *Comparison of the treatments per well*. In the *Classification*, for this particular case study (Section 3.3), the measurements were classified in three subsets: Cluster, junction and vesicle. In order to have this classification completed, first a ground truth data preparation is elaborated, this will be used for training the classifier algorithm, then it is necessary to perform feature selection and feature extraction and finally the feature values will be classified in one of these subsets. Upon completion of the classification task it is possible to do the comparison of the treatments per well and identify: (1) *Number of vesicles per nucleus* (c.f. Figure 5.6(a)), (2) *Number of clusters per nucleus* (c.f. Figure 5.6(b)), and (3) *Plasma-membranes (pixel) per nucleus* (c.f. Figure 5.6(c)) (CYW^{+}11).



(a) Number of vesicles per nucleus

(b) Number of clusters per nucleus

(c) Plasma-membranes (pixel) per nucleus

Figure 5.6: *Comparison of results with three phenotypic groups*

5.6 Conclusions and Future work

The use of cluster computing in Bioinformatics research, especially in cytomics, has been proved to be highly necessary due to the large datasets generated in HTS workflows. Due to the advantage of using parallelism in cluster architectures, traditional algorithms used for instance in image analysis need to be adapted to such environments. In this chapter, we have presented the cluster integration to CytomicsDB architecture and explored the effect of parallel computing on the performance of these algorithms in HTS experiments. We have shown that both segmentation and tracking algorithms can be parallelized efficiently with segmentation scaling linearly up to at least 96 processors using a combination of stack and slice level concurrency. Additionally, it has been described for the case study presented in Section 3.3 how the steps for data analysis are accomplished. Finally, it is still possible to reduce the volume of data generated in the image analysis stage e.g. auxiliary images such as binary masks and trajectories. Instead of storing the image files as tiff files, a matrix can be used to store the location of the objects of interest.

Chapter 6

Conclusions and Future work

This thesis addresses our research in the design and development of CytomicsDB, a comprehensive data management system for cytomics. This chapter presents the conclusions of the research. First, it addresses the research questions and the problem statements identified in this work. Second, the contibutions of this thesis are highlighted and finally the conclusions are presented, including an outline of future work.

6.1 Research questions and Problem statement

In this section, the evidence collected throughout this dissertation is summarized and used to address the research questions developed in this thesis. The Problem statement (PS) and the four Research questions (RQ) developed in this work are listed as follows:

- **PS**: *How to optimally/flexible organize the data managed for cytomics so as to be able to deal with the data deluge?*
- **RQ1**: *Which components and processes are required to build in a data management platform for cytomics?*
- **RQ2**: *How can be addressed the needs of metadata organization handled in Cytomics?*
- **RQ3**: *How can we prove the consistency of the metadata managed in Cytomics?*
- **RQ4**: *How can we speed up the data processing in Cytomics?*

6.1.1 Designing a software architecture for cytomics

- RQ1: Which components and processes are required to build in a data management platform for cytomics?

In Chapter 2 it was highlighted that given the large amount of different conditions and the readout of the conditions through images, it is clear that the HTS approach requires a proper data management system to reduce the time needed for experiments and the chance of man-made errors. Additionally, it was emphasized that due to the fact that there is different types of data, the experimental conditions need to be linked to the images produced by the HTS experiments with their metadata and the results of further analysis. Moreover, HTS experiments never stand by themselves, as more experiments are lined up, the amount of data and computations needed to analyze these increases rapidly. To that end it was necessary first to design an automated workflow for HTS experiments. Subsequently, it was proposed a software architecture which supports the demands for data management in typical HTS workflows. Finally, it was presented the database design capable to store experiments metadata and image and data analysis results.

6.1.2 Adressing the metadata organization

- RQ2: How can be addressed the needs of metadata organization handled in Cytomics?

In Cytomics, the study of cellular systems at the single cell level, High-Throughput Screening (HTS) techniques have been developed to implement the testing of hundreds to thousands of conditions applied to several or up to millions of cells in a single experiment. Recent technological developments of imaging systems and robotics have lead to an exponential increase in data volumes generated in HTS experiments. This is pushing forward the need for a semantically oriented bioinformatics approach capable of storing large volume of linked metadata, handling a diversity of data formats, and querying data in order to extract meaning from the experiments performed.

For addressing a solution, in Chapter 3 it was necessary to analyze and categorize the metadata managed in HTS experiments. The metadata was organized in five levels according to their role in the automated HTS workflow presented in Chapter 2. These levels are: (1) *Project*, (2) *Experiment*, (3) *Plate-wells*, (4) *Raw images*, and (5) *Measurements*. Finally, a case study is introduced which describes how the metadata is managed in CytomicsDB platform. The proper organization of the metadata facilitates scientist to work with a common data model thus improves the interoperability and the understandability of the metadata managed by the system.

6.1.3 Metadata validation

- RQ3: How can we prove the consistency of the metadata managed in Cytomics?

High-Throughput Screening (HTS) techniques are typically used to identify potential drug candidates. These type of experiments require invest in large amount of resources. The appropiate data management of HTS experiments has become a key challenge in order to succeed in the target validation. Current developments in imaging systems has to cope with computational requirements due to the significant increment of volumes of data. However, no special care has been taken to ensure the consistency, integrity and reliability of the data managed in HTS experiments. The appropiate validation of the data used in an HTS experiment has turned to be a key success factor in the target validation, thus a mandatory process to be included in the HTS workflow.

For tackling with these issues, in Chapter 4 firstly, it was analyzed which are the validation needs in the automated HTS workflow, then it was described the strategies available for metadata validation and how is the validation workflow developed by CytomicsDB, the workflow is explained using two case studies based on the *Compounds validation* and *siRNA validation*. Furthermore, it was described the implementation of the validation strategies in the system architecture. Finally, it was analyzed and evaluated the validation strategies in order to select the most appropiate ones according to the type of metadata managed by the platform. The strategy "Trust your friends" implemented with the "Most similar" function was chosen for the metadata validation stage.

6.1.4 Speeding up data processing

- RQ4: How can we speed up the data processing in Cytomics?

According to the automated workflow for HTS experiments presented in Chapter 2, the last two stages corresponds to the image and data analysis. These stages require high performance computing resources. Moreover due to the large volume of data involved in the analysis it is mandatory to take into account the location of the data for processing and how to adapt the algorithms so as can be used in a cluster environment.

It was demostrated in Chapter 5 that based on the two case studies for segmentation and object tracking algorithms, the performance of the parallelized version of both algorithms has been improved using a cluster environment. The processing duration has been reduced from aprox. 1 day to the range between 2 and 4 hours. Finally, it has been highlighted that in the data analysis stage is relevant to consider where to process the data. Thus, it has been also described how the MonetDB.R package has been used in CytomicsDB in order to avoid the overhead of shuffling data between different systems.

6.1.5 Adressing the Problem Statement

- PS: How to optimally/flexible organize the data managed for cytomics so as to be able to deal with the data deluge?

With reference to the answers of the four questions I may conclude that it is needed to design and develop a comprehensive data management platform for *Cytomics*. A platform capable to adapt to the continuos changes in Cytomics environment. Our platform called CytomicsDB relies on an architecture which is based on components which facilitate the integration with other systems in the cytomics domain, the experiments metadata has been properly organized so scientist can have a common data model which promote data collaboration and sharing, and special care has been taken for ensuring the consistency of the metadata managed by the platform. Finally, data processing in cytomics is becoming a challenge due to the large volume of information generated, thus CytomicsDB architecture has been designed to easily integrate the advantages of clustering computing for the image analysis and to speed up the data analysis avoiding the movement of huge amounts of data for processing.

6.2 Contributions

To address my problem statement it was necessary to collect data from the Toxicology Group at Leiden University. The data was obtained by analyzing the protocols, workflows, tools, software and harware environment where High-Throughput Screening experiments are developed. Based on the research findings across this work, it is possible to highlight the following main contributions of this study:

1. The design of an automated workflow system for HTS experiments.

2. The organization of experiments metadata for providing a common data model for data discovery, understandability, interoperability and data sharing.

3. Ensure metadata consistency by including a strict validation procedure in the workflow system of HTS experiments.

4. An integrated platform to automate the data management, capable to speed up the image and data analysis stage in the HTS experiments workflow by integrating a computational cluster to CytomicsDB architecture and including MonetDB.R package for managing the execution of the data analysis requests.

6.3 Future work

During the development of this thesis it was possible to identify other directions for further research. These other directions are summarized as follows:

1. **Reduction of the volume of data generated in HTS experiments**. One of the main challenges for HTS workflow systems is the management of huge amounts of data, each stage of the HTS workflow generate continuously more data. However, during the image analysis this amount of data is increased even more due to the generation of auxiliary image files e.g. binary masks and trajectories. These

image files are relevant for further analysis of the behavior of the cells exposed to different types of treatments during the experiment. Avoiding the use of binary data as an output of the image analysis reduces significantly the resources needed for its storage and further consultation.

2. **Data persistence** Other criteria to take into account is related to the data persistence. Relational Databases have been the dominant technology for data persistence for many years. Most of the databases in life sciences are based on a relational approach, even with the proliferation of other technologies such as NoSQL databases (Pok11). There is the possibility that relational databases may become an issue with the constantly demanding requirements due to the growing data size and computational resources needed for its processing.

3. **Efficient storage of data** In the universe of big data there is a tendency to use specialized distributed file systems e.g. Hadoop (Whi09) instead of typical Relational Database Management Systems (RDBMS). Unlike RDBMS it is not necessary to pre design an schema or prepare a sophisticated structure for the data before using it. The balance between scalability, standards for querying, availability, interoperability should be analyzed for structured and unstructured data. This will open the possibility to design hybrid solutions or to evaluate the impact and costs of turning data from unmanaged to managed, from disconnected to connected, from invisible to findable and from single use to reusable.

In Cytomics environment, scientist has to continuosly deal with a large volume of structured and unstructured data, this condition in particular, makes a challenge the interoperability for any platform developed for cytomics. *CytomicsDB* approach is an effort for developing a framework which takes care of the standardization of the unstructured data, providing a common data model layer for HTS experiments. This model as well is suitable for the integration with other systems in Cytomics, in special other repositories, which allow the validation of key metadata used in the experiments, thus ensure reliability of the data stored. Other possible solutions for cytomics data management, should take special care in the use of data model standards for enhancing the collaboration and data sharing in the scientific community.

Bibliography

[Alv] *96 well plate format*, `http://reinnervate.com/why-alvetex/alvetex-product-catalogue/`, Accessed: 2015-09-25. (cited on page 11).

[AMF00] George Avery, Charles McGee, and Stan Falk, *Product review: Implementing lims: A "how-to" guide.*, Analytical Chemistry **72** (2000), no. 1, 57 A–62 A. (cited on page 20).

[Ber03] Bryan Bergeron, *Bioinformatics computing*, Prentice Hall/Professional Technical Reference, Upper Saddle River, NJ, 2003. (cited on page 82).

[BKMC$^+$12] D.A. Benson, I. Karsch-Mizrachi, K. Clark, D. J. Lipman, J. Ostell, and E. W. Sayers, *Genbank*, Nucleic Acids Research **40** (2012), no. Database issue, D48–D53. (cited on page 63).

[BLW$^+$08] E. Bruford, M. Lush, M. Wright, T. Sneddon, S. Povey, and E. Birney, *The hgnc database in 2008: a resource for the human genome.*, Nucleic acids research **36** (2008), no. Database issue. (cited on page 60).

[BMK09] Peter A. Boncz, Stefan Manegold, and Martin L. Kersten, *Database architecture evolution: Mammals flourished long before dinosaurs became extinct*, PVLDB **2** (2009), no. 2, 1648–1653. (cited on pages 34 and 71).

[BN06] J. Bleiholder and F. Naumann, *Conflict handling strategies in an integrated information system*. (cited on pages 57, 58 and 61).

[BN09] ———, *Data fusion*, ACM Comput. Surv. **41** (2009), no. 1, 1:1–1:41. (cited on pages 57, 61 and 72).

[Bon02] Peter A. Boncz, *Monet: A next-generation dbms kernel for query-intensive applications*, Ph.d. thesis, Universiteit van Amsterdam, Amsterdam, The Netherlands, May 2002. (cited on pages 33, 34, 71 and 85).

[BSV11] L. M. F. Bertens, J. Slob, and F.J. Verbeek, *A generic organ based ontology system, applied to vertebrate heart anatomy, development and physiology.*, J. Integrative Bioinformatics **8** (2011), no. 2. (cited on page 52).

[BV08] M. Belmamoune and F. J. Verbeek, *Data integration for spatio-temporal patterns of gene expression of zebrafish development: the gems database*, Journal of Integrative BioInformatics **5(2):92** (2008). (cited on page 39).

[BWTB08] E. Bolton, Y. Wang, P. Thiessen, and S. Bryant, *Chapter 12 - pubchem: Integrated platform of small molecules and biological activities*, Annual Reports in Computational Chemistry, vol. 4, Elsevier, 2008, pp. 217 – 241. (cited on page 58).

[CDV08] David Chen, Guy Doumeingts, and François Vernadat, *Architectures for enterprise integration and interoperability: Past, present and future*, Comput. Ind. **59** (2008), no. 7, 647–659. (cited on page 17).

[CFK+11] C. Colmsee, Steffen Flemming, M. Klapperstuck, M. Lange, and U. Scholz, *A case study for efficient management of high throughput primary lab data*, BMC Research Notes **4** (2011), no. 1, 413. (cited on page 50).

[CJL+06] A. Carpenter, T. Jones, M. Lamprecht, C. Clarke, I. Kang, O. Friman, D. Gertin, J. Chang, R. Lindquist, J. Moffat, P. Golland, and D. Sabatini, *Cellprofiler: image analysis software for identifying and quantifying cell phenotypes*, Genome Biology 7(10) **10** (2006), no. 7. (cited on page 40).

[CMS06] T.P.S. Chan, P. Malik, and R. Singh, *An interactive visualization-based approach for high throughput screening information management in drug discovery*, Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE, Aug 2006, pp. 5794–5797. (cited on page 51).

[Cor88] F. Corpet, *Multiple sequence alignment with hierarchical clustering.*, Nucleic Acids Research **16** (1988), no. 22, 10881–10890. (cited on page 74).

[CYW+11] L. Cao, K. Yan, L. Winkel, M. de Graauw, and F.J. Verbeek, *Pattern recogntion in high-content cytomics screens for target discovery - case studies in endocytosis*, PRIB 2011, 2011. (cited on pages 28, 29, 30, 47, 48, 50, 81 and 91).

[DAJ12] S. K. Dubey, A. Anand, and H. Jangala, *Laboratory information and management system: A tool to increase laboratory productivity*, Clinical Research and Regulatory Affairs **29** (2012), no. 2, 46 – 56. (cited on pages 20, 21 and 22).

[DC08] Ewa Deelman and Ann Chervenak, *Data management challenges of data-intensive scientific workflows*, Proceedings of the 2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (Washington, DC, USA), CCGRID '08, IEEE Computer Society, 2008, pp. 687–692. (cited on pages 16 and 17).

[dGCW^{+}13] M. de Graauw, L. Cao, L. Winkel, M. H. A. M. van Miltenburg, S. LeDévédec, M. Klop, K. Yan, C. Pont, V-M. Rogkoti, A. Tijsma, A. Chaudhuri, R. Lalai, L. Price, F. Verbeek, and B. van de Water, *Annexin a2 depletion delays egfr endocytic trafficking via cofilin activation and enhances egfr signaling and metastasis formation*, Oncogene (2013). (cited on page 47).

[DGDLM13] Y. Demchenko, P. Grosso, C. De Laat, and P. Membrey, *Addressing big data issues in scientific data infrastructure*, Collaboration Technologies and Systems (CTS), 2013 International Conference on, May 2013, pp. 48–55. (cited on page 18).

[Di13] Z. Di, *Development of automatic image analysis methods for high-throughput and high-content screening*, Ph.d. thesis, Leiden University, Leiden, The Netherlands, December 2013. (cited on page 14).

[EBG^{+}12] Kevin W. Eliceiri, Michael R. Berthold, Ilya G. Goldberg, Luis Ibanez, B. S. Manjunath, Maryann E. Martone, Robert F. Murphy, Hanchuan Peng, Anne L. Plant, Badrinath Roysam, Nico Stuurman, Jason R. Swedlow, Pavel Tomancak, and Anne E. Carpenter, *Biological imaging software tools*, Nat Meth **9** (2012), no. 7, 697–710. (cited on pages 11, 12 and 13).

[FFM05] A. Fuxman, E. Fazli, and R. J. Miller, *Conquer: Efficient management of inconsistent databases*, Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (New York, NY, USA), SIGMOD '05, ACM, 2005, pp. 155–166. (cited on page 57).

[GAK^{+}14] Robert Grandl, Ganesh Ananthanarayanan, Srikanth Kandula, Sriram Rao, and Aditya Akella, *Multi-resource packing for cluster schedulers*, Proceedings of the 2014 ACM Conference on SIGCOMM (New York, NY, USA), SIGCOMM '14, ACM, 2014, pp. 455–466. (cited on page 81).

[GM07] Estelle Glory and Robert F. Murphy, *Automated subcellular location determination and high-throughput microscopy*, Developmental Cell **12** (2007), no. 1, 7 – 16. (cited on page 12).

[GMNR05] M. Gudgin, N. Mendelsohn, M. Nottingham, and H. Ruellan, *Soap message transmission optimization mechanism*, 2005. (cited on page 33).

[Gro10] The HDF Group, *Hierarchical data format version 5*, 2000-2010. (cited on page 46).

[GW06] R. C. Gonzalez and R. E. Woods, *Digital image processing (3rd edition)*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006. (cited on page 82).

[Han02] G. Hannon, *Rna interference*, Nature **418** (2002), no. 6894, 244–251. (cited on page 59).

[Hey02] Stephan Heyse, *Comprehensive analysis of high-throughput screening data*, 2002, pp. 535–547. (cited on page 18).

[HT03] T. Hey and A. Trefethen, *The data deluge: An e-science perspective*, pp. 809–824, John Wiley Sons, Ltd, 2003. (cited on page 17).

[JZR⁺08] M. Johnson, I. Zaretskaya, Y. Raytselis, Y. Merezhuk, S. McGinnis, and T. Madden, *Ncbi blast: a better web interface*, Nucleic Acids Research **36** (2008), no. suppl 2, W5–W9. (cited on page 60).

[KAEM13] Stephen Kaisler, Frank Armour, J. Alberto Espinosa, and William Money, *Big data: Issues and challenges moving forward*, 2014 47th Hawaii International Conference on System Sciences **0** (2013), 995–1004. (cited on page 19).

[KBK$^+$09] A. Kallergi, Y. Bei, P. Kok, J. Dijkstra, J.P. Abrahams, and F.J. Verbeek, *Cyttron: A virtualized microscope supporting image integration and knowledge discovery*, Cell Death and Disease Series: Proteins Killing Tumour Cells (2009), 291 – 315. (cited on page 39).

[KBL$^+$08] K. Kohl, G. Basler, A. Ludemann, J. Selbig, and D. Walther, *A plant resource and experiment management system based on the golm plant database as a basic tool for omics research*, Plant Methods **4** (2008), no. 1, 11. (cited on page 51).

[KGG10] S. Kalani, D. Gupta, and R. Gupta, *Good laboratory practices : compliance using lims*, no. 1, 41–43. (cited on page 22).

[Kos06] Kurt Kosanke, *Iso standards for interoperability: a comparison*, Interoperability of Enterprise Software and Applications (Dimitri Konstantas, Jean-Paul Bourrières, Michel Léonard, and Nacer Boudjlida, eds.), Springer London, 2006, pp. 55–64. (cited on page 17).

[Kri09] Wim P. Krijnen, *Applied statistics for bioinformatics using r*, 2009. (cited on page 86).

[KS06] M. Keith and M. Schincariol, *Pro ejb 3: Java persistence api (pro)*, Apress, Berkely, CA, USA, 2006. (cited on pages 33 and 85).

[Lak99] Joseph R. Lakowicz, *Principles of fluorescence spectroscopy*, second ed., Kluwer Academic/Plenum Publishers, 1999. (cited on page 13).

[LC09] Vebjorn Ljosa and Anne E Carpenter, *Introduction to the quantitative analysis of two-dimensional fluorescence microscopy images for cell-based screening*, PLoS Computational Biology **5** (2009), no. 12, e1000603. (cited on page 12).

[Lev66] V.I. Levenshtein, *Binary codes capable of correcting deletions, insertions and reversals*, Soviet Physics Doklady **10** (1966), 707. (cited on pages 67 and 74).

[LYdB$^+$10] S. LeDévédec, K. Yan, H. de Bont, V. Ghotra, H. Truong, E. Danen, F.J. Verbeek, and B. van de Water, *A systems microscopy approach to understand cancer cell migration and metastasis*, Journal of Cellular and Molecular in Life Science (2010). (cited on pages 28, 30 and 31).

[LZCV14] E. Larios, Y. Zhang, L. Cao, and F.J. Verbeek, *Cytomicsdb: A metadata-based storage and retrieval approach for high-throughput screening experiments*, Pattern Recognition in Bioinformatics (Matteo Comin, Lukas Käll, Elena Marchiori, Alioune Ngom, and Jagath Rajapakse, eds.), Lecture Notes in Computer Science, vol. 8626, Springer International Publishing, 2014, pp. 72–84. (cited on pages 54 and 85).

[LZY$^+$12] 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*, Proceedings of the 1st Int. Conf. on Health Information Science, vol. 7231, 2012, pp. 76–87. (cited on pages 42, 46, 54, 69, 71 and 80).

[MA04] R. T. Marler and J. S. Arora, *Survey of multi-objective optimization methods for engineering*, Structural and Multidisciplinary Optimization **26** (2004), no. 6, 369–395. (cited on pages 67 and 75).

[Mah91] R. R. Mahaffey, *Information technology in the laboratory*, Chemometrics and Intelligent Laboratory Systems **13** (1991), no. 1, 69 – 74. (cited on page 20).

[MAT10] MATLAB, *version 7.10.0 (r2010a)*, The MathWorks Inc., Natick, Massachusetts, 2010. (cited on page 31).

[MCV$^+$06] P. Malik, T. Chan, J. Vandergriff, J. Weisman, J. DeRisi, and R. Singh, *Information management and interaction in high-throughput screening for drug discovery*, Database Modeling in Biology: Practices and Challenges (Z. Ma and J. Chen, eds.), Springer Verlag, 2006. (cited on page 53).

[MF08] L. Mayr and P. Fuerst, *The future of high-throughput screening*, Journal of Biomolecular Screening (2008). (cited on page 41).

[Miz02] I. Mizrachi, *Chapter 1 genbank: The nucleotide sequence database*, The NCBI Handbook [Internet] (J. McEntyre and J. Ostell, eds.), Bethesda (MD): National Center for Biotechnology Information (US), 2002. (cited on page 60).

[MKSP00] D. R. Maglott, K. S. Katz, H. Sicotte, and K. D. Pruitt, *Ncbi's locuslink and refseq*, Nucleic Acids Research **28** (2000), no. 1, 126–128. (cited on page 63).

[Mon] *Boost your data analytics*, `https://www.monetdbsolutions.com/solutions/analytics`, Accessed: 2015-02-23. (cited on page 86).

[Mot99] A. Motro, *Multiplex: A formal model for multidatabases and its implementation*, Next Generation Information Technologies and Systems (RonY. Pinter and Shalom Tsur, eds.), Lecture Notes in Computer Science, vol. 1649, Springer Berlin Heidelberg, 1999, pp. 138–158 (English). (cited on page 56).

[Mot01] ______, *Data integration: Inconsistency detection and resolution based on source properties*, In Workshop on Foundations of Models for Information Integration (FMII), 2001, pp. 429–444. (cited on page 57).

[NSD+10] D. Nix, T. Di Sera, B. Dalley, B. Milash, R. Cundick, K. Quinn, and S. Courdy, *Next generation tools for genomic data generation, distribution, and visualization*, BMC Bioinformatics **11** (2010), no. 1, 455. (cited on page 51).

[NWH+10] B. Neumann, T. Walter, J.K. Hériché, J. Bulkescher, H. Erfle, C. Conrad, P. Rogers, I. Poser, M. Held, U. Liebel, C. Cetin, F. Sieckmann, G. Pau, R. Kabbe, A. W′unsche, V. Satagopam, M.H. Schmitz, C. Chapuis, D.W. Gerlich, R. Schneider, R. Eils, W. Huber, J.M. Peters, A.A. Hyman, R. Durbin, R. Pepperkok, and J. Ellenberg, *Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes*, Nature **Apr 1** (2010), no. 464(7289), 721–727. (cited on page 31).

[O'N] Elizabeth J. O'Neil, *Object/relational mapping 2008: Hibernate and the entity data model (edm)*, Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08. (cited on page 33).

[PCC13] Pasquale Pagano, Leonardo Candela, and Donatella Castelli, *Data interoperability*, Data Science Journal **12** (2013), GRDI19–GRDI25. (cited on page 18).

[Pok11] Jaroslav Pokorny, *Nosql databases: A step to database scalability in web environment*, Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services (New York, NY, USA), iiWAS '11, ACM, 2011, pp. 278–283. (cited on page 97).

[QSY+11] Y. Qin, G. Stokman, K. Yan, S. Ramaiahgari, F.J. Verbeek, B. van de Water, and L. Price, *Activation of epac-rap signaling protects against cisplatin-induced apoptosis of mouse renal proximal tubular cells*, Journal of Biological Chemistry (2011), In Press. (cited on page 28).

[Say08] E. Sayers, *E-utilities quick start. entrez programming utilities help [internet]*, `http://www.ncbi.nlm.nih.gov/books/NBK25500/`, 2008, Accessed: 2014 Oct 21. (cited on page 66).

[SKE07] Paul J. Smith, I.A. Khan, and R.J. Errington, *Cytomics and drug development*, Cytometry Part A **71A** (2007), no. 6, 349–351. (cited on pages 14, 16 and 17).

[SMS+02] P.T. Spellman, M. Miller, J. Stewart, C. Troup, U. Sarkans, S. Chervitz, D. Bernhart, G. Sherlock, C. Ball, M. Lepage, M. Swiatek, W.L. Marks, J. Goncalves, S. Markel, D. Iordan, M. Shojatalab, A. Pizarro, J. White, R. Hubley, E. Deutsch, M. Senger, B.J. Aronow, A. Robinson, D. Bassett, C.J. Stoeckert, and A. Brazma, *Design and implementation of microarray gene expression markup language (mage-ml)*, Genome Biology (2002), no. 3(9):RESEARCH0046. (cited on page 39).

[Sta06] Garrick Staples, *Torque resource manager*, Proceedings of the 2006 ACM/IEEE Conference on Supercomputing (New York, NY, USA), SC '06, ACM, 2006. (cited on page 81).

[SZK+11] D.O. Skobelev, T.M. Zaytseva, A.D. Kozlov, V.L. Perepelitsa, and A.S. Makarova, *Laboratory information management systems in the work of the analytic laboratory*, Measurement Techniques **53** (2011), no. 10, 1182–1189. (cited on page 20).

[TGDS04] Kerstin Thurow, Bernd Göde, Uwe Dingerdissen, and Norbert Stoll, *Laboratory information management systems for life science applications*, Organic Process Research & Development **8** (2004), no. 6, 970–982. (cited on page 20).

[Tox] *Visualisation core facilities*, `http://toxicology.leidenuniv.nl/research/facilities`, Accessed: 2015-03-25. (cited on page 12).

[Whi09] Tom White, *Hadoop: The definitive guide*, 1st ed., O'Reilly Media, Inc., 2009. (cited on page 97).

[WSP+07] M. Wendl, S. Smith, C. Pohl, D. Dooling, A. Chinwalla, K. Crouse, T. Hepler, S. Leong, L. Carmichael, M. Nhan, B. Oberkfell, E. Mardis, L. Hillier, and R. Wilson, *Design and implementation of a generalized laboratory data model*, BMC Bioinformatics **8** (2007), no. 1, 362. (cited on pages 42 and 51).

[Yan13] K. Yan, *Image analysis and platform development for automated phenotyping in cytomics*, Ph.d. thesis, Leiden University, Leiden, The Netherlands, November 2013. (cited on pages 31 and 82).

[YLL⁺11] 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*, Proceedings IEEE Conf. Engineering and Technology (CET 2011), vol. 7, 2011, pp. 195–198. (cited on pages 10, 11, 13, 14, 27, 32 and 83).

[YV12] K. Yan and F.J. Verbeek, *Segmentation for high-throughput image analysis: Watershed masked clustering*, Proceedings of the 5th International Conference on Leveraging Applications of Formal Methods, Verification and Validation: Applications and Case Studies - Volume Part II (Berlin, Heidelberg), ISoLA'12, Springer-Verlag, 2012, pp. 25–41. (cited on pages 48, 49, 80 and 81).

[YVDvdW09] K. Yan, F. J. Verbeek, S. Le Dévédec, and B. van de Water, *Cell tracking and data analysis of in vitro tumour cells from time-lapse image sequences.*, VISAPP (1) (Alpesh Ranchordas and Helder Araújo, eds.), INSTICC Press, 2009, pp. 281–286. (cited on pages 28, 30, 80 and 81).

[YXQZZH12] Z. Yong-Xin, L. Qing-Zhong, and P. Zhao-Hui, *Two-stage data conflict resolution based on markov logic networks*, Chinese Journal of Computers (2012). (cited on page 56).

Summary

In biomedical research, the study of cellular systems on large scale is called Cytomics. High-Throughput screening (HTS) is one of the most common techniques for target validation; it allows testing a large number of chemicals against disease targets for identifying hits. Moreover, an HTS environment is composed by a diversity of components that have to face challenges in terms of managing large volume of data, data heterogeneity, no integrated complex hardware architecture, diversity of software tools, managing protocols, etc. Therefore, this current scenario is pushing forward the need to develop a comprehensive data management platform for experiments in Cytomics. This new approach should be capable to enhance: (1) exploration of large volumes of data, (2) interoperability, (3) understandability, and (4) sharing and reusability of the data.

In this thesis, in four chapters our research on the development of CytomicsDB is described. This constitutes a platform which integrates the whole HTS workflow into a single system. An automated HTS workflow has been proposed to speed up the processes from plate design to image and data analysis. Chapter 2 focuses on the architecture of CytomicsDB, which relies on a modern relational database system, capable to integrate to other legacy systems used in HTS environments or other external repositories for data validation. The architecture also promotes data sharing and collaboration using several security roles, which allow scientist to publish their data or grant access to other researchers to their own experiments.

In Chapter 3, we further elaborate our approach in the metadata management performed by CytomicsDB. A semantic layer has been built in order to facilitate the understandability and exploration of the large volume of images and metadata involved in HTS experiments and at the same time allowing scientists to integrate new tools and APIs taking care of the image and data analysis. These results will become part of the experiments metadata and will be available for semantic post analysis.

In Chapter 4, we introduce the validation process as performed in CytomicsDB. Management of the HTS information is one of the key challenges for drug discovery

and in order to ensure consistency, integrity and reliability of the data stored in the platform it is compulsory to perform a strict validation process in every stage of the HTS workflow. CytomicsDB facilitates this validation process using web services, which will prove each critical entry with an internal or external repository. The metadata that we store become a key parameter for performing further image/data analysis and drill down the results of different experiments datasets.

Chapter 5 describes the integration of CytomicsDB architecture to the Leiden Life Sciences Cluster (LLSC) and the environment for data processing provided by MonetDB. In this new environment the image analysis algorithms need to be adapted, and the performance of the resulting parallelized algorithms is evaluated. The steps involved in this stage are also further elaborated in a image and data analysis data flow.

The current challenges in HTS experiments are pushing forward the need to design and develop more complex platforms for *Cytomics*, which are capable to adapt to the continuous changes presented in its environment. CytomicsDB's architecture facilitates the integration with other systems involved in HTS experiments. Moreover, the metadata has been organized in order to have a common data model, which enhances understandability, collaboration and data sharing. In CytomicsDB, the validation of the metadata stored in the repository is considered a key process. The consistency of metadata is guaranteed by accessing external biological databases that are supported and maintained by the scientific community. Finally, our platform integrates the advantages of clustering computing for the image analysis stage and also the data processing functionality provided my MonetDB.R package to speed up the data analysis avoiding the movement of huge amounts of data for processing.

Samenvatting

In biomedisch onderzoek wordt veelvuldig gebruik gemaakt van celculturen in het onderzoek. Het onderzoeksveld waarin op grote schaal gebruik gemaakt wordt van cel-systemen wordt aangeduid met Cytomics. High-Throughput Screening (HTS) is een van de meest gebruikte technieken in Cytomics voor het valideren van potentiële kandidaat "targets" voor de ontwikkeling van medicijnen. Hierbij kunnen binnen een experiment grote hoeveelheden verschillende chemische verbindingen worden getest die een cruciale rol spelen in een ziektebeeld en daarmee kandidaat zijn voor het ontwikkelen van medicijnen tegen deze ziekte. Een omgeving voor High-Throughput Screening is samengesteld uit een aantal componenten die onderling samenhang moeten vertonen. Voor een succesvolle toepassing van een dergelijke omgeving zijn er een aantal complicaties die moeten worden overwonnen: (1) het beheren van grote hoeveelheden data, (2) interoperabiliteit, (3) begrijpbaarheid en (4) het delen het hergebruik van data.

In dit proefschrift wordt in 4 hoofdstukken het onderzoek beschreven voor de ontwikkeling van een platform voor onderzoek en beheer van data, genaamd CytomicsDB, waarbij de gehele keten van werkzaamheden, het werkproces, in een systeem wordt ondergebracht. Een geautomatiseerde keten van werkzaamheden wordt voorgesteld waarmee een versnelling van de werkzaamheden teweeg kan worden gebracht; van het begin van het traject bij ontwerp van het experiment tot de data analyse aan het einde van het traject.

In Hoofdstuk 1 wordt een algemene inleiding geven van het werkveld en de concepten die van belang zijn voor het onderzoek. In Hoofdstuk 2 wordt een architectuur beschreven die voor CytomicsDB wordt gebruikt en dit gebaseerd is op een modern relationeel database systeem. Met een dergelijk systeem kunnen bestaande, maar deels verouderde systemen, zogenaamde legacy systemen, worden geïntegreerd; dit zijn de systemen die momenteel in HTS worden gebruikt, maar ook externe opslagsystemen die belangrijk zijn voor de validatie van de data. De voorgestelde architectuur is zeer geschikt voor het delen van data en samenwerking, waarbij verschillende niveaus van beveiliging van de data zijn ingevoerd die de wetenschapper in staat stellen data te publiceren voor de gehele gemeenschap of juist alleen een selecte groep mede-onderzoekers toegang te geven tot deze data.

Naast data zijn er gegevens over de data, bijvoorbeeld, hoe deze data tot stand zijn gekomen. Deze gegevens worden ook Metadata genoemd, in feite zijn dit eigenlijk data over data. In Hoofdstuk 3 wordt een strategie uitgewerkt voor het beheer van metadata zoals dat in CytomicsDB is ingebed. Er is een semantische schil geconstrueerd waarmee begrijpbaarheid en exploratie van grote volumina data en metadata de betrokken zijn bij een HTS experiment te faciliteren. Tegelijkertijd stelt deze semantische schil de wetenschapper in staat om nieuwe methoden en programma's te integreren waarmee de beeld- en data-analyse kan worden afgehandeld. De resultaten uit deze analyses zullen deel worden van de experiment metadata en komen beschikbaar voor een post data-analyse op een semantisch niveau.

In Hoofdstuk 4 introduceren we het proces van data validatie zoals dat in CytomicsDB wordt uitgevoerd. Het beheer van de HTS informatie is een van de belangrijkste uitdagingen bij het ontdekken van nieuwe medicijnen. Teneinde de consistentie, integriteit en betrouwbaarheid van de data opgeslagen in CytomicsDB te garanderen, is het verplicht in iedere stap van het HTS werkproces een nauwgezette validatie uit te voeren. In CytomicsDB wordt deze validatiestap gefaciliteerd door middel van web-services waarmee iedere nieuwe kritische invoer van data wordt getest aan de hand van een interne of externe opslagsystemen.

In Hoofdstuk 5 wordt de integratie van de CytomicsDB architectuur met een rekencluster, de Leiden Life Sciences Cluster (LLSC), beschreven in samenhang met de omgeving voor data processing die aanwezig is binnen het MonetDB database management system. Voor het efficiënt gebruik van het rekencluster moeten de algoritmen voor beeldanalyse worden aangepast. Deze parallellisatieslag is geëvalueerd en de prestaties van de herschreven algoritmen worden gepresenteerd. De stappen de betrokken zijn bij de analyses worden verder uitgewerkt in een werkproces voor beelden data-analyse.

De huidige stand van zaken in HTS experimenten is de uitdaging aan te gaan in de behoefte te kunnen voorzien om complexere platforms voor *Cytomics* te ontwikkelen waarmee ingewikkelde experimenten kunnen worden geanalyseerd en die zich kunnen aanpassen aan de continue veranderingen in het Cytomics veld. De architectuur van CytomicsDB faciliteert de integratie van met andere systemen die binnen HTS experimenten gebruikt worden. De metadata zijn, bovendien, zodanig georganiseerd dat ze voldoen aan een algemeen data-model waardoor de begrijpbaarheid, samenwerking en het delen van data sterk verbeterd is. De validatie van de metadata die in CytomicsDB worden opgeslagen is wordt een sleutel proces, de consistentie van deze metadata wordt verzekerd door een dubbele controle met data uit andere databanken die door de gemeenschap worden beheerd. Tot slot, het platform dat in dit proefschrift wordt gepresenteerd, integreert de voordelen van het toepassen van de beeldanalyse op rekenclusters en daarnaast het gebruik van de functionaliteit voor dataverwerking die in MonetDB is gebouwd met het softwarepakket MonetDB.R. Hiermee kan een aanzienlijke winst worden behaald met het verwerken van de data waarbij voorkomen wordt dat er heel veel data moeten worden verplaatst voor de verwerking.

List of 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.
- E. Larios, Y. Zhang, L. Cao, and F.J. Verbeek. **Cytomicsdb: A metadata-based storage and retrieval approach for high-throughput screening experiments**. Pattern Recognition in Bioinformatics (PRIB 2014). Lecture Notes in Computer Science, vol. 8626, Springer International Publishing, pages 72–84. 2014.
- E. Larios, Z. Xia, J. Slob and F. J. Verbeek. **A semantic-based metadata validation for an automated High-Throughput Screening workflow: Case study in CytomicsDB**. NETTAB 2015 - Integrative Bioinformatics 2015. (Submitted).
- E. Larios, D. van Es, K. Rietveld and F. J. Verbeek. **Cluster integration in CytomicsDB**. (Manuscript in preparation).
- B. Karasneh, D. Stikkolorum, E. Larios, M.R.V. Chaudron. **Assessing The Quality of UML Class Diagrams: A Comparative Study Between Experts and Students**. Educators Symposium, ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems 2015.

About the Author

Enrique Larios Vargas was born in 1977 in Lima, Peru. He studied at the Pontificia Universidad Catolica del Peru (PUCP) receiving a BSc degree in Informatic Engineering in 2000. Directly after his bachellor studies he started working as a project engineer in a telecommunications consulting firm. After working in this sector for 6 years, he started working as a part time lecturer in the Telecommunications Engineering department at PUCP. In 2006, he started his Master education at Universidad Nacional Mayor de San Marcos in Lima, Peru. During his master education he specialized in ICT Management and received the MSc degree in the end of 2008. In the same year, he was promoted to Assistant Professor and responsible of the Software Engineering specialization at the Telecommunications Engineering department at PUCP.

In 2010, Enrique applied to the BAPE Erasmus Mundus program for pursuing PhD studies, he was granted with a scholarship and admitted to Leiden University as a PhD student in the Section Imaging and BioInformatics at Leiden Institute of Advanced Computer Science (LIACS). Under the supervision of Dr. Fons J. Verbeek his research focused on the development of a comprehensive data management platform for Cytomics. His research project was developed in close collaboration between the Division of Toxicology of the Leiden Academic Centre for Drug Research (LACDR) and the Database Architecture Group at Centrum Wiskunde and Informatica (CWI) in Amsterdam. During the last years of his PhD studies he also worked as a lecturer in the Software Engineering course in the Master ICT in Business at Leiden University. His professional experience as an ICT engineer and scientist provide him with a broad perspective useful in the academy and business sector.