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

Larios Vargas, E.

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.

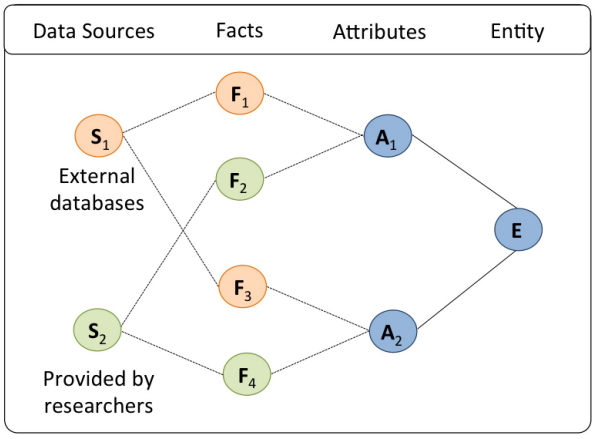



**Figure 4.1:** *Relationship between the entity, its attributes and fact values from data sources*

### 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.

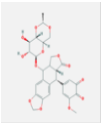
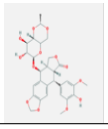
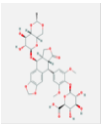
#### Compounds

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.

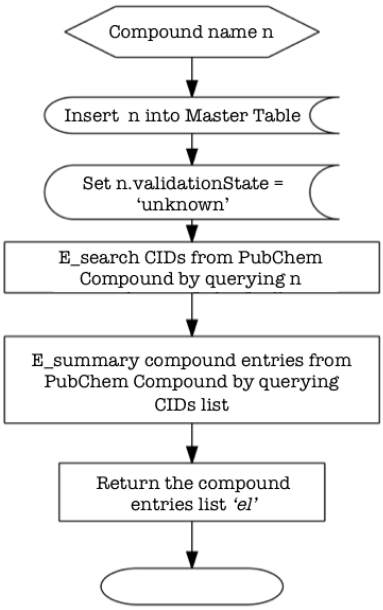



**Figure 4.2:** *Get candidates*

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

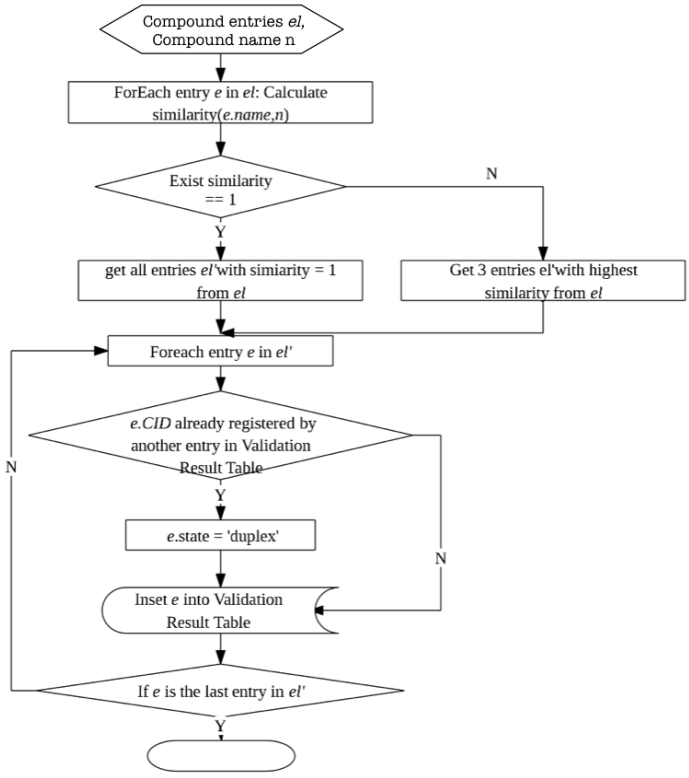



**Figure 4.3:** *Screening and Marking*

**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

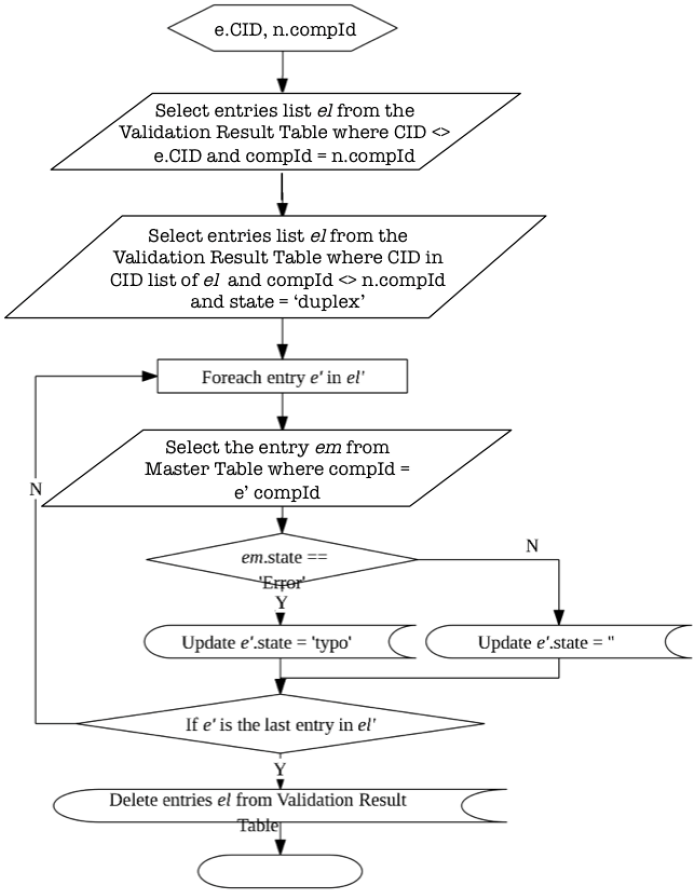



**Figure 4.4:** *Clean up*

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

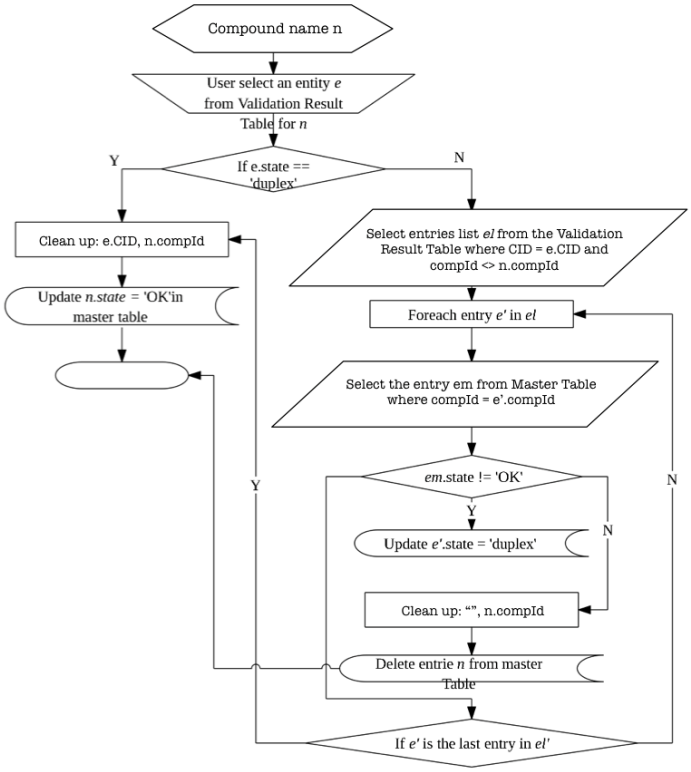



**Figure 4.5:** *Update duplex marks*

**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 | BLAST+ |
| | NM_003318 | |
| | NM_001166691 | |

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 | 4 |
| NM_003318.2 | |
| NM_003318.3 | |
| NM_003318.4 | |
| 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.

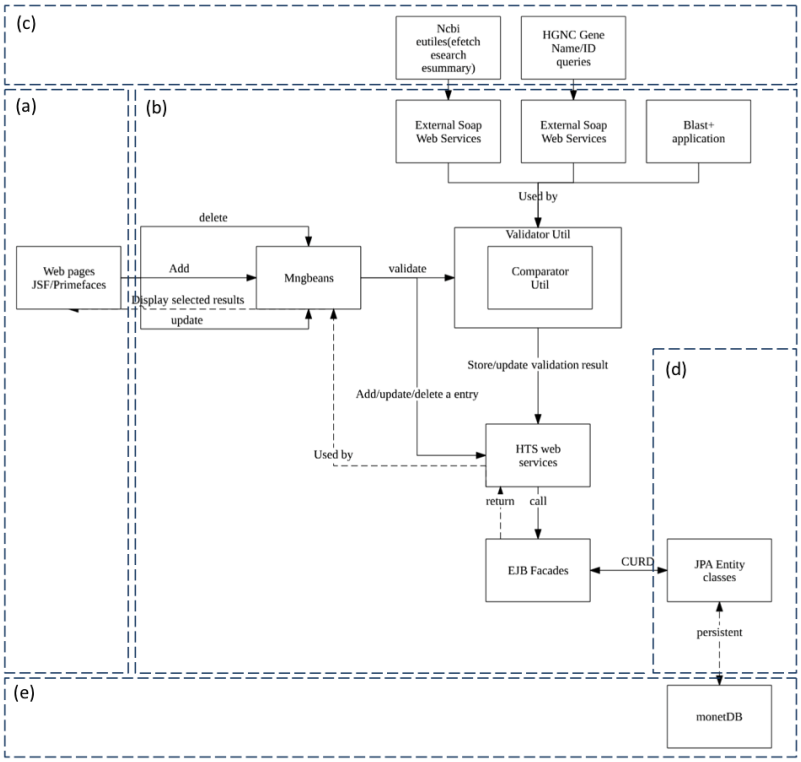
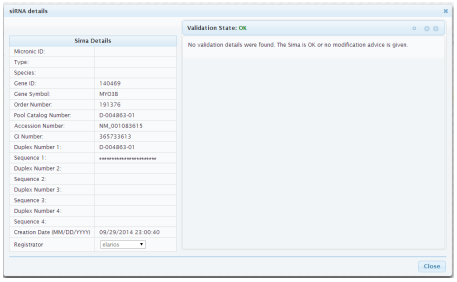



**Figure 4.6:** *Components diagram of the validation workflow*

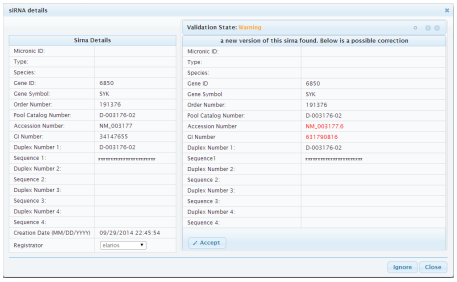
### The presentation layer

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

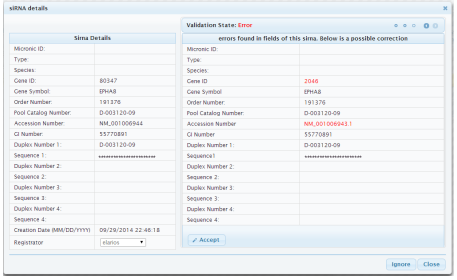
from the master table, the validation process will be called to update the duplicated information in the validation result table if it is necessary. Besides that, the presentation layer is the first stage of validation in the platform by considering mandatory fields for uploading experiment metadata. 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.



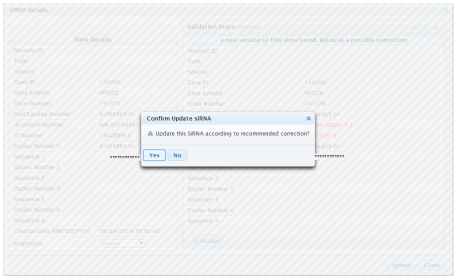
(a) Entry passed validation



(b) New version found in an external database



(c) Typo detected in the GeneID and Accession Number



(d) Confirm update

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

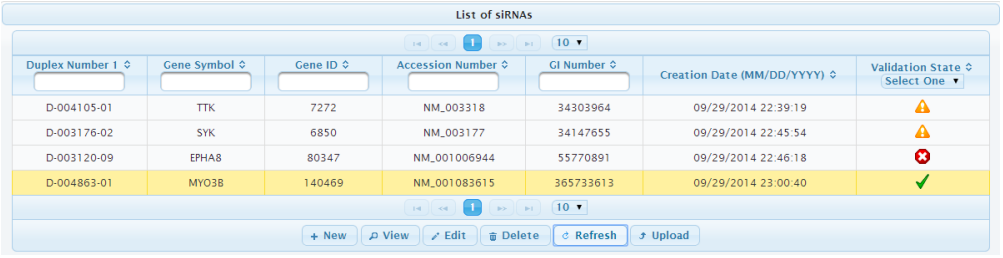



**Figure 4.8:** *Validation result interface*

### 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<br>LD: 0.983<br>RKR-GST: 0.925 | LCS: 0.567<br>LD: 0.573<br>RKR-GST: 0.527 | LCS: 0.974<br>LD: 0.974<br>RKR-GST: 0.949 | LCS: 0.93<br>LD: 0.93<br>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.