Chapter II DEVELOPING A BIOCURATION ASSISTING SYSTEM AND MAMMALIAN GENE EXPRESSION DATABASES AND DERIVING A NEW META-ANALYSIS METHOD.

**CONTENTS AND STRUCTURE**

INTRODUCTION.

SECTION A. DEVELOPMENT OF A WEB-BASED TOOL TO ASSIST BIOCURATION.

SECTION B. DERIVATION AND APPLICATION OF NOVEL META-ANALYSIS METHOD.

SECTION C. DEVELOPMENT OF NEW MAMMALIAN GENE EXPRESSION DATABASES.

DISCUSSION.

CONCLUSION.

REFERENCES.
INTRODUCTION

Establishing gene expression profiles is an important step towards understanding the mechanism of gene expression regulation and cellular/tissue functions and also for identification of biomarkers.

Understanding the gene expression behavior in different biological conditions has been important to address a wide variety of molecular biology questions. Some of the molecular biology techniques, such as the northern blotting, reverse transcription polymerase chain reaction (RT-PCR) and quantitative reverse transcription polymerase chain reaction (qRT-PCR), were used to analyze the expression profiles of a few genes of interest with high specificity and sensitivity. In the last few decades, many techniques, such as microarray, Expressed Sequence Tag (EST) profiling and Serial Analysis of Gene Expression (SAGE), allowed a screening expression pattern of thousands of genes at a time. Such high throughput methods aid molecular biologists in, a) identifying genes in differential expression pattern both in normal and abnormal conditions, b) finding the genes that indicate similar or differential behavior in different stages of disease conditions, c) classifying specific disease into subtypes based on differential expression profiles, d) analyzing the gene expression patterns in developmental stages of a tissue and e) identifying genes that are tissue-specific or constitutive expression.

Microarray data in the public repositories have been extensively used by researchers around the world.

The immense potential of high throughput techniques has led to the extensive studying of gene expression in a variety of biological systems at different stages. Microarray technique is one among the high throughput techniques used for gene expression profiling. Thus, through this technique enormous amount of gene expression data were generated. Besides being part of published literature, the data obtained from these high throughput technologies are deposited in public repositories such as Gene Expression Omnibus (GEO) [Edgar et al., 2003], Center for Information Biology gene EXpression (CIBEX) [Ikeo et al., 2003] and Stanford Microarray Database (SMD) [Sherlock et al., 2001].

Meta-analysis methods could address the issues associated with microarrays.

However, the widespread usage of microarray has highlighted some of its major limitations such as reliability and robustness. One of the popular approaches for addressing this issue was to
combine results from multiple independent of related studies using statistical methods, which is generally referred as meta-analysis [Ramasamy et al., 2008]. Although there are many microarray meta-analysis methods, they differ in terms of statistical parameters considered such as rank, p-value, and effect size as well as the purpose of usages like differentially expressed gene or pathway detection, co-expression analysis, prediction analysis, reproducibility analysis) [Tseng et al., 2012]  

**Using a novel meta-analysis method on biocurated gene expression data, a gene expression database for testicular conditions was developed.**

Our study on available meta-analysis methods revealed their inability to make maximum use of available gene expression data, and hence, development of a new meta-analysis method was undertaken. The novel microarray meta-analysis was applied to gene expression data corresponding to various testicular conditions compiled via biocuration. This attempt led to develop new gene expression database

The entire process is described in the three major sections, namely

1. Development of web-based workbench (tools), assisting biocuration.
2. Application of novel meta-analysis method on biocurated data.

**SECTION A. DEVELOPMENT OF WEB-BASED TOOLS TO ASSIST BIOCURATION.**

*A user-friendly computational system can enhance the efficiency of biocuration process.*

Biocuration is a process that involves collection of heterogeneous but related data from biomedical literature and public repositories. Thus, collected data are stored in the computer in readable format, leading to their Bioinformatics processing [Wiegers et al., 2014]. Data compilation can be approached in two ways, a) Manual curation: this involves thorough scanning of public resources for relevant data by pre-trained researchers called biocurators, and b) automated text mining: involves the use of computational tool for finding relevant information through natural language processing of public resources. Even though manual curation is time consuming, it is significantly more reliable than automated processes, in many cases. While automated text mining improves the speed of biocuration, it is error prone [Winnenburg et al., 2008]. Studies have shown that even
the best automated text mining method cannot compete with the accuracy of manual curation [Kwon et al., 2014]. Thus, it may be rewarding to address the problems associated with the biocuration. For example, it may be possible to assist biocurators in the process of biocuration. Hence, we have developed a web-based system to assist biocuration of the gene expression data related to testis tissue.


II-A.1.1. Third party tools used for developing the biocuration assisting system.

Apache 2.2 [http://httpd.apache.org/] was used as a web server on a 64 bit Red Hat Linux machine with 32 GB RAM and 3 TB HDD space. MySQL RDBMS [MySQL 5.1.69, http://www.mysql.com/] was used for data storage. The graphical user interface (GUI) was developed using PERL-CGI scripts [where hypertext markup language (HTML) tags are embedded within Practical extraction and reporting language (PERL) scripts]. Web technologies such as cascade style sheets (CSS) and JavaScript (JSS) were extensively used for implementing user-friendly features.

R scripts were written to process microarray rawdata downloaded from GEO, ArrayExpress, SMD using bioconductor packages such as ‘affy’, ‘marray’, ‘lumi’ and ‘codelink’ based on the microarray platform [Appendix A]. Complete scripts are given in Appendix C.

II-A.2. Results.

II-A.2.1. Usage of the web-based system to upload, edit and store biocurated data.

Systematic annotation and storage of biocurated data is as important as identifying relevant data among thousands of unrelated ones. Computational tools can assist largely in facilitating and speeding up the process. Thus, we developed a computational system that assist biocuration, figure II-1 illustrates its design. The system can be used as mentioned below:

The registration process is an important step towards avoiding unauthorized access and tracking the activities of users. This requires biocurator to use a PERL-CGI form to fill information such as name, institute, designation, email id, user name and password of choice [Figure II-2]. On form submission, the background PERL script will email the biocurator information to the system administrator. The system administrator will also receive link to a form, where he can either accept
or reject the registration proposal based on the authenticity of information provided. In both the circumstances, a notification about their registration status is sent to the user via email. If the registration is accepted, then the user details will be stored permanently in the in-house database and he/she will be able to login to the biocuration system.

*Figure 0-1. Major components of the biocuration system and the connections between them.*
Once the user’s registration is accepted, he/she can login to the system using login page [Figure II-3].
Figure 0-3. A snapshot of the web page to login into the form to upload biocurated data.

After logging into the system, the user is provided with a form using which manually curated data can be uploaded into the system.

Upload form: The PERL-CGI form was designed to accept two major types of information, such as

a. Gene-list: This refers to the list of genes whose expression profile is reported by the study in specific biological condition at specific tissue/cell type of a particular organism. If the gene-list is available as part of the main manuscript or supplementary notes in the form of table or running text or image, then biocurators prepare a gene-list in specific format [Table II-1] using open spreadsheet or Microsoft excel software. However, if a gene-list has to be made from the gene expression study, which is deposited in public repositories such as GEO, Array Express and SMD, then an in house web-based tool aids them in gene-list preparation [Figure II-4 to II-6]. The steps involved are:

Step 1: Enter the repository accession number of study of choice, to obtain a list of related hybridizations and brief description about them [Figure II-4].
Step 2: Select hybridizations corresponding to same/similar biological sample for merging and creation of gene-list [Figure II-4].

[Image of login page for data uploading form of MGEx-p]
Step 3: Gene expression details of first 100 probes related to selected hybridizations will be displayed in tabular format. By viewing, the table biocurator needs to enter column numbers of table having similar gene expression details to merge [Figure II-5].

Step 4: Download gene-list prepared after merging gene expression details from similar hybridizations [Figure II-6].
Table 0-1. Illustration of the format for gene-lists to be prepared from published literature.

<table>
<thead>
<tr>
<th>Gene name</th>
<th>Ensembl gene</th>
<th>UniGene gene</th>
<th>Entrez gene</th>
<th>Probe ID</th>
<th>Genbank accession</th>
<th>Clone ID</th>
<th>SwissProt name</th>
<th>Swissprot accession</th>
<th>Description</th>
<th>SAGE tag</th>
<th>MPSS tag</th>
</tr>
</thead>
</table>

Gene name: this column should include only official gene symbols

Ensembl gene: this column should include only gene identifiers from Ensembl

UniGene: this column should include only gene cluster id from UniGene

Entrez gene: this column should include only identifiers from Entrez gene database.

ProbeID: this column should include only probe identifiers from the microarray platform used.

Genbank accession: this should include only transcript identifiers from Genbank database.

Clone ID: this should include only clone identifiers corresponding to EST

SwissProt name: this should include only protein abbreviations from SwissProt/UniProt

SwissProt accession: this should include only protein accession from SwissProt/UniProt

Description: this should include only full name or description of the gene

SAGE tag: this should include only tag sequence from SAGE study

MPSS tag: this should include only tag sequence from MPSS study
Figure 0-4. A snapshot of the web interface to access processed data (either obtained from GEO/ArrayExpress or generated through in-house processing). Specific experiment identifier from GEO/ArrayExpress can be used to search data, and hybridizations corresponding to same tissue and condition can be merged for gene-list preparation.

---

Figure 0-5. A snapshot of the web page to visualize the details of multiple processed data before
merging and generating a gene-list.
First thirteen columns meant to store the gene identifiers from the study. ‘P-value’ stores the mean detection P-value across hybridizations; ‘Signal intensity’ stores mean signal intensity across selected hybridizations, ‘Type’ refers to type of processed data available (‘PAM’: if absolute call is available, ‘updown’: if differential expression is available), CallRatio summarizes expression behavior of a probe across hybridizations (For example, 2#0#0#0#0 indicates that the probe is having ‘Present’ status in 2 hybridizations, ‘Absent’ status in 0 hybridizations, ‘Marginal’ status in 0 hybridizations, ‘up-regulation’ status in 0 hybridizations and ‘down-regulation’ status in 0 hybridizations), ‘Score’ column shows overall expression status of the probe across hybridizations.

b. Gene-list associated information: When a relevant study is selected, biocurator will gather additional information as described below, along with gene-list.

I. Sample parameters: This includes details about biological sample used for gene expression study, which are as follows

1. Species: name of the organism from which sample was obtained.

2. Tissue: name of the tissue from which sample was extracted.

3. Sub tissue: name of the sub tissue to which the sample corresponds.

4. Main cell type: name of the cell type to which the sample corresponds.
5. Cell line: name of cell line, if the sample is obtained from a cell line.

6. Main condition: physiological or experimental or pathological condition of the sample considered.

7. Age: if the sample was obtained from an individual, then age of the individual is considered.

II. Experimental parameters:

1. Main experiment: name of the technology used for gene expression profiling.

2. Number of individuals: total number of individuals considered for sample extraction.

3. Number of samples: total number of samples used for the study.

4. Number of RNA isolations: total number of RNA isolations performed from samples.

5. Type of main experiment: this information is included only if microarray is used. Biocurator will mention that whether the conducted experiment is single or double channel experiment.

6. Validation experiment: information about an experiment conducted to verify the result of main experiment.

III. Expression parameters:

1. Expression status: expression status of the genes within uploaded gene-list.

2. Expression level: level of expression of genes within uploaded gene-list.

3. Type of raw data: this corresponds to the type of information obtained after processing microarray raw data.
IV. Statistics and other parameters:

1. Statistics: type of statistical procedures applied in the study for the identification of genes of interest.

2. The statistical significance of gene-list: statistical cut off applied to identify genes of interest.

3. Processing methods: name of the software/package used to normalize and summarize microarray raw data.

4. Selection criteria for gene-list: criteria chosen to report the gene-list in the relevant study (based on statistics or author bias)

5. The data source of the gene-list: name of the source from which gene-list is obtained.

6. Title of list: Title for the obtained gene-list, if provided by the author.

7. Extra information about the sample: for example, if sample extracted from smoking or diabetic individuals.

8. Availability of complete data: Yes/No, because the reported gene-list may or may not include all the genes considered within the study.

The upload form includes a series of drop down menus related to each gene-list associated features [Figure II-7]. The majority of them was hidden at first and will be displayed based on the selection of options in the connected drop down menu [Figure II-8]. The embedded javascript functions make sure that minimum required information is provided for each uploaded gene-list by the biocurator. On submission, the interface displays uploaded information in a tabular format and entails final confirmation. The confirmation leads to the storage of uploaded information in the server in comma separated value (CSV) format and uploader will be provided with the copy of it.
Figure 0-7. A snapshot of the form to upload the biocurated information.
Figure 0-8. Sequence of display of hierarchically connected drop down menus corresponding to
sample condition.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main condition1:</td>
<td>Select options</td>
<td>Main condition1:</td>
<td>Developmental stages</td>
<td>Select options</td>
</tr>
<tr>
<td>Main condition2:</td>
<td>Select options</td>
<td>Subcondition14:</td>
<td>Human</td>
<td>Select options</td>
</tr>
<tr>
<td>Main condition3:</td>
<td>Select options</td>
<td>Subcondition14a0:</td>
<td>Select options</td>
<td>Select options</td>
</tr>
<tr>
<td>Main condition4:</td>
<td>Select options</td>
<td>Subcondition14a0b0:</td>
<td>Select options</td>
<td>Select options</td>
</tr>
<tr>
<td>Main condition5:</td>
<td>Select options</td>
<td>Subcondition14a0b0c2:</td>
<td>Select options</td>
<td>Select options</td>
</tr>
</tbody>
</table>

A: Drop down menus for uploading sample condition information, B: selection of “Developmental stages” as main condition leading to the display of related sub conditions, C: selection of “Human” as sub condition (level1) leading to the display of next level of sub conditions, D: selection of “post embryonic” as sub condition (level2) leading to
the display of next level of sub conditions, E: selection of “Infant/Newborn/Neonate” as sub condition (level3) leading to the display of next level of sub conditions.

**Edit form:** Even though, manual curation is more accurate than any of exiting text mining approaches, it is not devoid of any error. Due to misinterpretation of curated data or fatigue that creeps during the curation process, one may a) choose and upload irrelevant study and b) upload wrong information from the relevant study. Because of these reasons, independent validation of uploaded information is imperative in biocuration. To facilitate such process, we developed an edit/ validation form [Figure II-9 and II-10]. Link to the form is embedded in the upload form.

*Figure 0-9. A snapshot of the web interface for editing/validating uploaded information.*
Figure 0-10. A snapshot of the edit form showing uploaded information in a tabular format.

The edit form displays list of all uploaded gene-lists from a relevant study on receiving inputs such as species, tissue and identifier of the study from public repository or literature database. By clicking on the gene-list name, biocurator can view its content in a tabular format. Each row represents different fields and biocurator will be provided with the option of either retaining or modifying the uploaded information. After the validation process is complete, new version of the uploaded information will be created. The uploaded information can be edited/validated for a maximum of 10 times.

II-A.2.2. Data storage: Submitted biocurated data was stored in computer friendly
format.

One of the principle objectives of biocuration is to structure publicly available heterogeneous data. The data uploaded through the web interface are stated in CSV files. Each line of the file is dedicated to carrying information about specific fields. This specific design will improve the ease of computational processing of the information.

**II-A.2.3. Validation:** The system was extensively used by multiple researchers to upload, edit manually curated data.

The system was used by about 40 researchers over the period of three years to upload 2077 gene-lists related to various testicular conditions from 350 studies, 2265 gene-lists related to various uterine conditions from 336 studies and 832 gene-lists related to 25 other healthy human tissues from 11 studies [Table II-2, II-3 and II-4].

*Table 0-2. Quantity of gene expression data, corresponding to testicular conditions, uploaded via the biocuration system.*

<table>
<thead>
<tr>
<th>Resource</th>
<th>Human</th>
<th>Mouse</th>
<th>Rat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Studies</td>
<td>Gene-lists</td>
<td>Studies</td>
</tr>
<tr>
<td>ArrayExpress</td>
<td>8</td>
<td>79</td>
<td>12</td>
</tr>
<tr>
<td>GEO</td>
<td>42</td>
<td>403</td>
<td>67</td>
</tr>
<tr>
<td>Stanford microarray database</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Main manuscript</td>
<td>62</td>
<td>203</td>
<td>88</td>
</tr>
<tr>
<td>Supplementary materials</td>
<td>15</td>
<td>30</td>
<td>27</td>
</tr>
</tbody>
</table>
Table 0-3. Quantity of gene expression data, related to uterine conditions, uploaded via the biocuration system.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Human</th>
<th>Mouse</th>
<th>Rat</th>
<th>Cow</th>
<th>Pig</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayExpress</td>
<td>6</td>
<td>25</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>GEO</td>
<td>92</td>
<td>824</td>
<td>19</td>
<td>85</td>
<td>4</td>
</tr>
<tr>
<td>caArray</td>
<td>2</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Main manuscript</td>
<td>186</td>
<td>980</td>
<td>14</td>
<td>64</td>
<td>2</td>
</tr>
<tr>
<td>Supplementary materials</td>
<td>31</td>
<td>145</td>
<td>5</td>
<td>18</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 0-4. Quantity of gene expression data, related to various healthy mammalian tissues, via biocuration system.

<table>
<thead>
<tr>
<th>Tissue</th>
<th>GEO</th>
<th>ArrayExpress</th>
<th>Row total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Studies</td>
<td>Gene-lists</td>
<td>Studies</td>
</tr>
<tr>
<td>Heart</td>
<td>5</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>Kidney</td>
<td>7</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>Liver</td>
<td>7</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>Brain</td>
<td>6</td>
<td>264</td>
<td>1</td>
</tr>
<tr>
<td>Lung</td>
<td>7</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>Ovary</td>
<td>7</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>Bone</td>
<td>5</td>
<td>21</td>
<td>1</td>
</tr>
<tr>
<td>Muscle</td>
<td>5</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>Spinal cord</td>
<td>6</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>Thyroid gland</td>
<td>6</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>Adrenal gland</td>
<td>6</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>Salivary gland</td>
<td>6</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>Prostate</td>
<td>6</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>GIT</td>
<td>5</td>
<td>47</td>
<td>0</td>
</tr>
<tr>
<td>Pituitary gland</td>
<td>5</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>Spleen</td>
<td>5</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Pancreas</td>
<td>5</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>Thymus</td>
<td>5</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>Placenta</td>
<td>5</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>Blood</td>
<td>3</td>
<td>29</td>
<td>1</td>
</tr>
</tbody>
</table>
II-A.2.4. **Efficiency increased**: An interface was created with upload and edit forms with an intention to minimize errors during the manual data entry process.

The interface was built in such a way that biocurators commit least errors during the data entry process. Attempts in this regard are as follows:

a. **Facilitating the use of controlled vocabulary for most of the features**: Controlled vocabulary is imperative for streamlined creation. Five senior curators with years of experience in biocuration were given the responsibility to design controlled vocabulary for required features based on current knowledge on molecular biology techniques and mammalian testis physiology. They are provided with an advanced upload form to add options to drop down menus related to different features [Figure II-11], while for other biocurators, this option will be disabled. For most of the features to be uploaded from relevant study, a biocurator is required to choose options in the series of connected drop-down menus, rather than typing them [Figure II-12]. Using combinations of javascript functions and CSS, display order of interconnected drop down menus is coordinated. Whenever, a biocurator comes across new information from relevant study, they can discuss with senior curators about the need for new option(s) in the specific drop down menu(s) and senior curators will take appropriate measures.
Figure 0-11. Snapshots illustrating the process of including new options into specific drop down menus in the form.
Figure 0-12. Blueprint of controlled vocabulary implemented through hierarchically connected drop down menus.
b. **Typing information through a virtual number pad:** Certain fields in the form require numerical information to be typed, which could be integer (E.g. 1, 2, 100, etc) or decimal number (E.g. 2.34, 34.56, 52.09, etc.) or other numerical representations (E.g. >=10, <8.5, etc). For those fields, a virtual number pad is provided for user to enter data, thereby minimizing the typing mistakes.

c. **Link to standard operating protocol (SOP) from the upload form:** A quick access to SOP document helps biocurator not only for clearing any doubt regarding the biocuration process, but also avoid them from committing mistakes.

**SECTION B. DERIVATION AND APPLICATION OF NOVEL META-ANALYSIS METHOD.**

*Meta-analysis methods provide reliable results by using the available microarray gene expression data.*

Meta-analysis provided hope to the researchers around the world for not only addressing common issues with microarray technique, but also promising more reliable and generalized results. This led to derivation of many microarray meta-analysis methods [Ramasamy *et al.*, 2008; Tseng *et al.*, 2012; Chang *et al.*, 2013], implementing various statistical approaches. Most of the existing microarray meta-analysis methods can be classified into four major categories [Ramasamy *et al.*, 2008], 1) voting method, 2) combining p-value, 3) combining ranks and 4) combining effect sizes.

*The failure of existing meta-analysis methods to utilize most of publicly available gene expression data prompted us to develop a new method.*

Each type has its own advantages and disadvantages over others [Table II-5]; however, usage of almost all of them is restricted to only a fraction of publicly available gene expression data. This serves as a major hurdle for researchers to make maximum use of publicly available gene expression data. With these observations, we decided to develop new meta-analysis method (a variant of the vote counting approach). The method focuses on deriving a consensus expression profile across most of the publicly available gene expression data and providing a quantitative measure for the reliability of consensus derived, known as “reliability score”.
Table 0-5. An account of advantages and disadvantages of existing microarray meta-analysis methods.

<table>
<thead>
<tr>
<th>Meta-analysis type</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| Vote counting      | 1) The method is simple and extendable.  
                      2) This method does not require the availability of complete gene expression data.  
                      3) This method can be used to combine gene expression data from different technologies such as RT-PCR, microarray, RNA-seq, Mass spec, etc. | 1) When significant genes are reported from a study, if p-value is derived from two tailed t-test, then the directionality of gene-regulation could be misinterpreted.  
                      2) In each gene expression study, threshold value to identify significant gene could be arbitrary. Thus, it completely ignores genes below threshold cutoff.  
                      3) This method does not account for the frequency of the genes across studies. Example: a gene which is expressed as per 10 lists and not expressed in 6 lists outweighs a gene which is expressed as per three studies and not available in the rest of 13 lists.  
                      4) The ranked list produced by this method will be granular (rank changes drastically along the list) |
| Combining p-value   | 1) The method is simple and extendable.  
                      2) It considers expression information of all genes from a study.  
                      3) It can account for the frequency of the genes across studies.  
                      4) The ranked list provided by this method will be in finer scale (gradual decrease in ranks along the list)  
                      5) Irrespective of the number of studies considered, the result obtained will be on a common scale (0 to 1)  
                      6) The method shows higher detection capability (detects more differentially expressed genes) and biological association (detects differentially expressed genes associated with the known condition related pathways) | 1) It requires availability complete data from a microarray study.  
                      2) It requires result of hypothesis testing for meta-analysis. (Some microarray one-channel experiments that provide detection information based on signal to noise ratio. These could not be utilized by this method).  
                      3) The result can be dominated by outliers |
<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combining effect sizes</td>
<td>1) It considers expression information of all genes from a study.</td>
<td>1) It requires availability complete data from a microarray study.</td>
</tr>
<tr>
<td></td>
<td>2) It can account for the frequency of the genes across studies.</td>
<td>2) The result can be dominated by outliers</td>
</tr>
<tr>
<td></td>
<td>3) The ranked list provided by this method will be in finer scale (gradual decrease in ranks along the list)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4) The use of effect size facilitates seamless integration of microarray data from different platforms</td>
<td></td>
</tr>
<tr>
<td>Combining ranks</td>
<td>1) This method considers order of significance of each gene across lists</td>
<td>1) This method does not account for the frequency of the genes across studies.</td>
</tr>
<tr>
<td></td>
<td>2) This method can be used in situations where complete gene expression data available or not available</td>
<td>2) Its implementation requires high computational complexity</td>
</tr>
<tr>
<td></td>
<td>3) This method produces stable meta-analysis results compared to other methods.</td>
<td></td>
</tr>
<tr>
<td>7) The method shows better robustness (effect when an irrelevant study is mistakenly added) compared to others.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

II-B.1.1. Biocurated data.

All curated gene expression data for various mammalian testicular conditions were uploaded as comma separated value files containing list of genes and its associated information. Within each file, list of genes and its associated information was stored in specific rows and columns. This leads to easy parsing and handling by computer programs.

II-B.1.2. A novel meta-analysis method was derived.

PERL scripts were written to handle data from multiple types of experiments and to quantify the repeated expression status across experiments, and thus assign a 'reliability score'. CPAN modules such as Spreadsheet::ParseExcel, DBI were used for smooth execution of the program. The implementation of the meta-analysis method can be divided into four parts:

1. **Information extraction:**
   As previously mentioned all uploaded biocurated data was stored in a computer friendly format in CSV file. Different features about the gene expression data uploaded are stored in different locations within the file; consequently, the program can directly fetch the required data instead of searching them in the file. For example, information about organism from which sample is collected and used for the experiment is always stored in the first line; similarly information about the type of experimental technique used will be stored in 12th line. While the table containing set of reported genes begins from 51st line.

2. **Gene identifier conversion:**
   Conversion of given diverse type of gene identifiers is one of the most important aspect of the meta-analysis method [Figure II-13]. When the gene-list is derived from published literature, it was observed that each study reports genes of interest using different type of gene identifiers ranging from the probe identifier of the microarray platform to the full name of the gene to official gene symbol to Refseq transcript identifier to Uniprot protein accession name. Adding to the complication, the list could include heterogeneous identifiers.
To address this challenge, we used GIDCon [http://resource.ibab.ac.in/GIDCON/geneid/home.html] for efficient gene identifier conversion. The database is created by downloading gene annotation information from various resources and linking them based on their association. GIDCon is constantly updated and maintained.

When the gene-list is derived from public repository, it was observed that the gene-list contains homogeneous type of identifiers (i.e., probe identifiers in case of microarray experiments, SAGE tags and MPSS tags in case of SAGE and MPSS experiments). However, the identifiers from microarray experiments are platform specific, thus, gene identifiers obtained from different microarray platform changes. In this case, platform annotation file (which is downloaded from the public repository) was utilized for gene identifier conversion. Based on the platform information provided in the gene-list, different annotation files are used for probe identifier conversion.
Figure 0-13. The process of gene identifier conversion before implementing meta-analysis method.
3. **Deriving the ‘reliability scores’:**

Expression status of each gene within a gene-list can be either ‘transcribed’ or ‘not detected’ or ‘not available’. The ‘not available’ status could be due to two different scenarios, 1) in case of Affymetrix microarray data, some of the gene’s expression is termed as marginal and 2) in other case the expression profiling for the gene is not tested in the experiment. The method assigns transcribed score of ‘2’ and not-detected score of ‘-2’ for each gene in each gene-list considered.

\[ RSc(G) = \sum_{j=1}^{K} Sc(G)j \]

RSc (G) is the final reliability score of a gene, obtained by summation of reliability score [Sc (G)] from K gene-lists corresponding to the same combination of condition and location. Sc (G) can be either 2 for transcribed status or -2 for not detected status or 0 for not available status [Figure II-14].

**Note:** Most of the microarray platforms represent a gene by more than one probe/probe sets. The method considers gene as transcribed, if at least one of them is detected, otherwise gene is considered to be ‘not detected’.

4. **Storing the meta-analysis result:**

The result of microarray meta-analysis is stored in the mysql tables. For each specific combination of condition and location, a table is created to include information about genes, condition, location, expression status and reliability scores (Section C).
Figure 0-14. Illustration of reliability scoring implementation.

II-B.2. Results.
The novel microarray meta-analysis was successfully applied on 4174 gene-lists from more than 500 independent studies.
II-B.2.1. Pseudocode for the PERL script implementing novel microarray meta-analysis.

**PROGRAM** ReliabilityScoringProgram

LOAD CPAN packages such as DBI, Spreadsheet::ParseExcel;

READ masterfile containing file names of all uploaded gene-lists related to specific condition;

Store the genelist filenames into an array;

FOR (taking each filename within the array)

    Initialize arrays such as spec, Condition, tissue, Celltype, Celltype2, expr_type, Condition2, subcon, array0, array01, array02, array03, array04, geneid, val1 to NULL;

    Instantiate DBI handler 'dbh' to connect gene database;

    Instantiate DBI handler 'dbh2' to connect Id_list database;

    Instantiate DBI handler 'dbh1' to connect CLUSTER database;

####Section1: parsing genelist to gather uploaded information####

Instantiate an object 'oExcel' for Spreadsheet::ParseExcel;

Parse the excel file through oExcel object;

FOR (isheet considering 1st sheet to last sheet of the file)

    FOR (iR taking values from 1st row to last row within the sheet)

    FOR (iC taking values from 1st column to last column within the sheet)

        Export the value from the cell of iR row and iC column to oWkC;

        IF (iC = 0) THEN

            IF (iR >= 57) THEN

                Store each gene name to array 'aray0';

            END IF

        END IF
ENDIF

ENDIF

IF iC = 1 THEN
    IF iR >= 57 THEN
        IF (the cell contains non empty value) THEN
            Store each value (ensembl gene id) to array 'aray01';
        ELSE
            store empty value to the array 'aray01';
        ENDIF;
    ENDIF;

ENDIF;

ENDIF;

IF iC = 2 THEN
    IF (iC = 2) THEN
        IF (iR >= 57) THEN
            IF (the cell contains non empty value) THEN
                Store each value (unigene id) to array 'aray03';
            ELSE
                store empty value to the array 'aray03';
            ENDIF;
        ENDIF;
    ENDIF;

ENDIF;

IF iC = 3 THEN
    IF iR >= 57 THEN

ENDIF

ENDIF

IF iR >= 57 THEN
Store each value (entrez gene id) to array 'aray04';

ENDIF;

ENDIF;

IF \( iC = 4 \) THEN
  IF \( iR \geq 57 \) THEN
    Store each value (probe id) to array 'aray02'
  ENDIF;
ENDIF;

IF \( (iR = 0) \) THEN
  Store each value into array 'spec';
ENDIF;

IF \( (iR = 1) \) THEN
  Store each value into array 'tissue';
ENDIF;

IF \( (iR = 2) \) THEN
  Store each value into array 'celltype';
ENDIF;

IF \( (iR = 4 \ OR \ iR = 5 \ OR \ iR = 6 \ OR \ iR = 7) \) THEN
  Store each value into array 'celltype2';
ENDIF;

IF \( (iR = 8) \) THEN
Store each value into array 'cond';

ENDIF;

IF (iR = 9) THEN
    Store each value into array 'subcon'
ENDIF;

IF iR = 51 OR iR = 52 OR iR = 53 THEN
    Store each value into array 'expr_type'
ENDIF;

ENDFOR;

ENDFOR;

ENDFOR;

ENDFOR;

Section2: Converting all gene identifiers to entrez gene id#

IF (spec[1] = 'Homo sapiens') THEN
    abr = 'Hs';
    tabl = 'human'
ENDIF;

IF (spec[1] = 'Mus musculus') THEN
    abr = 'Mm';
    tabl = 'mouse';
ENDIF;
IF (spec[1] = 'Rattusnorvegicus') THEN
    abr = 'Rn';
    tabl = 'rat';
ENDIF;

IF (sizeof(ary0) > sizeof(ary01)) THEN
    l1 = sizeof(ary0);
ELSE
    l1 = sizeof(ary01);
ENDIF;
IF (sizeof(ary02) > l1) THEN
    l2 = sizeof(ary02);
ELSE
    l2 = l1;
ENDIF;
IF (sizeof(ary03) > l2) THEN
    l3 = sizeof(ary03);
ELSE
    l3 = l2;
ENDIF;
IF (sizeof(ary04) > l3) THEN
l4 = sizeof(aray04);

ELSE
l4 = l3;
ENDIF;

FOR (e taking value from 1 to l4);

IF (aray04[e] exist AND not empty) THEN
Execute SQL query "select GeneID from tabl where GeneID = 'aray04[e]'" using DBI handler 'dbh2';

IF (value is fetched from GIDCon table) THEN
Store value in the array 'geneid';

ELSE
GOTO ensembl;
ENDIF;
ELSE
GOTO ensembl;
ENDIF;

ELSE
GOTO ensembl;
ENDIF;

ensemb1:

IF (aray01[e] exist AND not empty) THEN
Execute SQL query "select GeneID from tabl where EnsemblIDrlike'aray01[e]'" using DBI handler 'dbh2';

IF (value is fetched from GIDCon table) THEN
Store value in the array 'geneid';

ELSE


GOTO genename;

ENDIF;

ELSE

GOTO genename;

ENDIF;

genename:

IF (aray0[e] exist AND not empty) THEN

Execute SQL query "select GeneID from tabl where Symbol = 'aray0[e]'" using DBI handler 'dbh2';

IF (value is fetched from GIDCon table) THEN

Store value in the array 'geneid';

ELSE

Execute SQL query "select Symbol from tabl where Synonyms like '%aray0[e]%'" using DBI handler 'dbh2';

IF (no value is fetched from GIDCon table) THEN

GOTO affymetrix;

ELSE

IF (single gene symbol is fetched from gidcon table) THEN

Store fetched value to val;

Execute SQL query "select GeneID from $tabl where Symbol='val'" using DBI handler 'dbh2';

Store fetched value into the array 'geneid';
ELSE
  GOTO affymetrix;
ENDIF;
ENDIF;
ENDIF;
ELSE
  GOTO affymetrix;
ENDIF;
ENDIF;
ELSE
  GOTO affymetrix;
ENDIF;

affymetrix:
  IF (array02[e] exist and not empty) THEN
      Execute SQL query "select count(*) from tabl where AffymetrixIDrlike array02[e]" using DBI handler 'dbh2';
      IF (count is equal to 1) THEN
          Execute SQL query "select GeneID from tabl where AffymetrixIDrlike 'array02[e]'" using DBI handler 'dbh2';
          The value fetched from the query is stored into array 'geneid';
      ELSE
          GOTO unigene;
      ENDIF;
  ELSE
      GOTO unigene;
  ENDIF;
ELSE
  GOTO unigene;
ENDIF;
ELSE
  GOTO unigene;
ENDIF;
unigene:

    IF (array03[e] exist and not empty) THEN
        Execute SQL query "select count(*) from tabl where UniGeneID not regexp 'array03[e]\^\d*\]' and UniGeneIDrlike 'array03[e]'" using DBI handler 'dbh2';
        IF (the count is equal to 1) THEN
            Execute SQL query 'select GeneID from $tabl where UniGeneID not regexp $array03[e]\^\d*' and UniGeneIDrlike $array03[e]' using DBI handler 'dbh2';
            The value fetched from the query is stored into array 'geneid';
        ENDIF;
    ENDIF;
ENDFOR;

##Section3: Implementing reliability score and storing results in the mysql table##

FOR (each id in the array 'geneid')
    Execute SQL query 'select gene_abr from gene_alias_parent where db_id=id' using DBI handler 'dbh';
    Store fetched gene symbol into array a1;
ENDFOR;

tabl2 = Name_of_table_for_specific_condition;

Execute SQL query "select own_id,gene_abr from gene_alias_parent where own_id like 'abr\%'" using DBI handler 'dbh';
WHILE (pair of ownid and gene_symbol is fetched by execution of the query)
    DO
Execute SQL query "select own_id from tabl2 where own_id='ownid' and cond1='cond[0]' and asso_cond1='subcon[0]' and cond2='cond[1]' and asso_cond2='subcon[1]"" using DBI handler 'dbh1';

IF (ownid is not fetched from the execution of the query) THEN

Execute SQL query "insert into tabl2(own_id,gene_abr,species,tissue,cell_type,subcell_type,cond1,asso_cond1,cond2,asso_cond2,expr_type1,expr_type2,compiled_score) values('ownid','gene_symbol','spec[1]','tissue[1]','celltype[1]','celltype2[1]','cond[0]','subcon[0]','cond[1]','subcon[1]','expr_type[1]','expr_type[2]',0)" using DBI handler 'dbh1';

ENDIF;
ENDWHILE;

score = 0;

IF (expr_type[1] = 'Present') THEN
    score = 2;
ENDIF;

IF (expr_type[1] = 'Absent') THEN
    score = -2;
ENDIF;

FOR (each geneSymbol within the array a1)
    Execute SQL query "update tabl2 set compiled_score=compiled_score+score where gene_abr='geneSymbol' and species='spec[1]' and cell_type='Celltype[1]' and subcell_type='Celltype2[1]' and cond1='cond[0]' and asso_cond1='subcon[0]' and cond2='cond[1]' and asso_cond2='subcon[1]"" using DBI handler 'dbh1';
ENDFOR;
ENDFOR;
SECTION C. DEVELOPMENT OF NEW MAMMALIAN GENE EXPRESSION DATABASES.

*Gene expression databases are immensely useful for researchers as they provide access to existing gene expression data, leading to the new hypothesis generation and discoveries.*

Although a huge amount of gene expression data has been generated, expression information of only a fraction of experimental genes is available in the published literature and while only researchers with computational expertise can make use of complete expression data stored in a wide variety of formats in public repositories. Thus, gene expression database is one stop solution for molecular biologists, by not only providing access to publicly available expression information in a structured way, but also paving way for novel hypothesis generations and discoveries.

*A gene expression database for mammalian testicular conditions is developed, where the novel meta-analysis method can be implemented on existing gene expression data.*

There are two major reasons that prompted us to develop new mammalian gene expression database, 1) limitations in existing gene expression databases such as missing publicly available gene expression data, less user friendly query features and lack of consensus expression status across multiple studies and 2) With the exception of Rhodes *et al.*, [Rhodes *et al.*, 2004a], who developed oncomine database by integrating cancer microarray data sets using voting approach [Rhodes *et al.*, 2004], none of the meta-analysis approaches have been implemented on numerous data sets related to different conditions to make a database.

Among adult tissues, testis shows unique cell differentiation and substantially more widespread transcription of the genome [Soumillon *et al.*, 2013]. Thus, we decided to take it as first tissue for database creation.

**II-C.1. Methods.**

**II-C.1.1. Third party tools** used for database creation.

libwww-perl (LWP) CPAN module [http://search.cpan.org/dist/libwww-perl/lib/LWP.pm]. The web-based graphical user interface was developed using PERL-CGI (which involves embedding HTML tags in the PERL scripts). In order to make web pages user friendly, web technologies such as cascade style sheets (CSS) and Javascript were also utilized.

II-C.1.2. Overall contents and functioning of the database.

The primary objective of developing mammalian gene expression database for testis (MGEx-Tdb) was to provide a list of genes, which are reliably associated with different testicular conditions. Hence, gene expression data received from biocuration system were subjected to novel meta-analysis approach and the result was stored using relational database management system (RDBMS). The interactive, user-friendly web-based interface was developed to provide access to the list of associated genes for different biological states in testis [Figure II-15].

Figure 0-15. Overall flow of information in MGEx-Tdb.
II-C.1.3. MySQL database: content and connectivity.

The database design included three tables (Gene_info, Transcript_info and Protein_info) that store accessory information from NCBI Gene, Refseq and Uniprot respectively. While the database includes 79 tables storing expression information along with a reliability score for all the genes in different testicular conditions in different mammalian species (human, mouse and rat). The database schema is described in figure II-16. Appendix B includes the description of the tables such as Gene_info, Protein_info, Transcript_info and htnormal.

Figure 0-16. Representation of the MGEx-Tdb database schema.
II-C.2. Results.
The new user-friendly mammalian gene expression database for testis was developed and freely available at [http://resource.ibab.ac.in/MGEx-Tdb/](http://resource.ibab.ac.in/MGEx-Tdb/).

II-C.2.1. Usage of MGEx-Tdb.

Search features:

The website ([http://resource.ibab.ac.in/MGEx-Tdb/](http://resource.ibab.ac.in/MGEx-Tdb/)) allows users to search the backend database, in two distinct ways [Figure II-17]:

1. Searching though the specific combination of species, sub tissue/cell type and condition:
   This includes three drop-down menus including options for species, sub tissue/cell type and conditions [Figure II-18].

![Figure 0-17. A snapshot of the home page of MGEx-Tdb, showing two main search modes](image)

![Figure 0-18. A snapshot of the search mode to obtain expression profile of genes in specific testicular conditions.](image)

One of the issues of having three different drop-down menus was that often many combinations of options leads to no results. To address this issue, a javascript function was used to disable certain options within a drop down menu based on selections on adjacent drop down menus [Figure II-19 and II-20].
Figure 0-19. An illustration of inactivation of options under ‘Cell type’ from drop-down menu, in a case where Homo sapiens was selected.

Figure 0-20. An illustration of inactivation of options under ‘Condition’ from drop-down menu, in a case where Homo sapiens and Testis were selected.

2. Searching through gene name(s): This allows user to enter the name of one or more genes, separated by comma [Figure II-21]. The query terms are scanned in gene symbol, gene name and gene aliases field of gene_info table.

Figure 0-21. A snapshot of MGEx-Tdb, showing the option to search database using gene
Output features:

On querying with specific testicular condition MGEx-Tdb provides lists of genes that are [transcribed and dormant along with the reliability score [Figure II-22]. Each listed gene is hyperlinked to a page that provides not only accessory information about the gene [Figure II-23 to II-26], but also expression information in different testicular condition [Figure II-27].

*Figure 0-22. A snapshot of the result page of MGEx-Tdb listing genes associated with adult normal human testis. Gene related information can be obtained by clicking on the hyperlinked gene symbol within the table.*
Figure 0-23. A snapshot of the result page showing links to different gene associated information.

![Gene Information Table]

Figure 0-24. A snapshot of the result page showing gene information, obtained from NCBI gene.
Figure 0-25. A snapshot of the result page showing transcript information.
Figure 0-26. A snapshot of the result page showing protein information.

Figure 0-27. A snapshot of the result page showing the expression information of the gene in different testicular conditions.

On querying with a specific term, based on the match with existing gene names or gene symbol or aliases, the web-server shows hits in two categories [Figure II-28], 1) identical matches and 2) partial matches. User can choose among them to get related information.
Gene expression databases were created using compiled gene expression data related to uterine conditions
Similar to MGEx-Tdb, two new gene expression databases were developed, to test the robustness of the programs and their universal nature. Mammalian gene expression database for uterus tissue (MGEx-Udb, http://resource.ibab.ac.in/MGEx-Udb/) provides gene expression profiles for various uterine conditions and human gene expression database for endometrial receptivity (HGEx-ERdb, http://resouce.ibab.ac.in/HGEx-ERdb/) provides gene expression profiles for various phases of endometrial receptivity in humans. All three databases have been published and are being used by researchers worldwide.