CHAPTER 3 MATERIALS AND METHODOLOGY

3.1 Hardware and software

3.1.1 REannotate

REannotate [41] is a computational tool that performs post-processing of repeat annotation results generated by RepeatMasker, a program that computationally detects interspersed repeats and low complexity DNA sequences [44]. The post-processing is required to improve the biological interpretation of the RepeatMasker annotations because the annotated sequences often correspond to fragments of the repetitive elements resulting from accumulation of insertions and deletions [40]

REannotate can automatically perform main three tasks, including defragmentation of the dispersed repetitive elements, resolution of the temporal order of the elements' insertions in the nested clusters, and forecasting the age of the elements after the insertion time [41]. In this work, REannotate was employed to defragment the HERV annotations. Furthermore, there are some beneficial measurements additionally calculated by REannotate. These measurements would be kept as additional characteristics of HERVs.

3.1.2 Python programming language

Python is a high-level programming language that can be easily applied to many different problems and integrated to a system more efficiently [45]. It can run on a wide variety of operating systems: UNIX, Windows, Mac, and so on. Python is one programming language that is being used more and more in bioinformatics works. Also, there is an available tool that can facilitate the biological computation which is written in Python called Biopython.

3.1.3 MySQL database

MySQL database is a rational database management system or RDBMS that is the most popular today. It is freely downloadable and available as an open source software [46]. This software was used as the main database management system in this work.

3.1.4 HTML

HTML or Hypertext Markup Language is the basic building-blocks for webpage development and extensively used today. In this work, it is responsible for the development of the most user interfaces.

3.1.5 PHP script language

PHP is a general-purpose scripting language that is widely used at this time [47]. It is rather suitable for web development to produce dynamical web pages. Thus, PHP is usually used as an embedded code into the HTML documents. PHP can be used on many operating systems and platforms, and with several rational database management

systems, including MySQL database. In this work, PHP was used to develop most of user interfaces along with the HTML, and also connect with the MySQL database.

3.1.6 Apache web server

The Apache web server [48] is a free web server software, that helps to deliver content accessed through the Internet by clients. To develop the web application in this work, it required the Apache web server to be responsible for interacting with users and databases, and responding to the users.

3.1.7 R programming language

R is a programming language and software environment for statistical computing and graphics. It is an open source under the terms of the Free Software Foundation's GNU General Public License. There are a wide variety of statistical and graphical techniques provided in R, such as, classification, clustering, time-series analysis, linear and nonlinear modeling, and so on. It can compile and run on a various UNIX platforms, Windows, and MacOS [49].

3.2 Data resources

3.2.1 Human repeat annotation data

This is an annotation of all repeats, including short interspersed nuclear elements (SINEs), long interspersed nuclear elements (LINEs), long terminal repeat elements (LTRs, which contains HERVs), DNA repeats, simple repeats, low complexity repeats, and satellite repeats, of the human reference sequence version hg19/GRCh37 (Feb., 2009). The annotation was created by using RepeatMasker [44], along with the Repbase repeat library (Release 20090120). It is a UCSC data table, named rmsk, and freely downloadable on the UCSC table browser [50].

3.2.2 Human gene annotation data

This is also a UCSC track that shows the gene annotation resulting from the UCSCs' predictions [51]. This UCSC gene annotation table was named as knownGene table. Data from RepSeq, GenBank, CCDS and UniProt were used in the predictions. The track contains both protein-coding genes and putative non-protein coding genes. To be consistent with the repeat annotation data, the UCSC gene annotation of the hg19 human reference sequence would be used in this work.

3.2.3 Cross-reference gene ID data

This is the central cross-reference table for the UCSC known genes [51]. The data has been collected in the table named kgXref and can be downloaded from the UCSC table browser as well. In the table, there are several cross-reference IDs for a UCSC known gene, including UCSC known gene ID, GenBank accession number, SWISS-PROT protein accession number, SWISS-PROT display ID, gene symbol, NCBI RefSeq ID, and NCBI protein accession number.

3.2.4 SLE microarray data (GSE20864)

This data was obtained from GEO database. Primarily, this data was performed to be used in gene ontology and network pathway analysis of the SLE patients. It is the gene expression profiling of peripheral blood cells which were obtained from 21 SLE patients and 45 healthy individuals. All patients are women range from 26 to 72 years old in age [52]. The platform of the DNA microarray that was used for generating the data is Hitachisoft AceGene Human Oligo Chip 30K Chip Version (GPL1291). Traditionally, there are 30,336 probes in total.

3.3 System design and requirements

3.3.1 System flow design

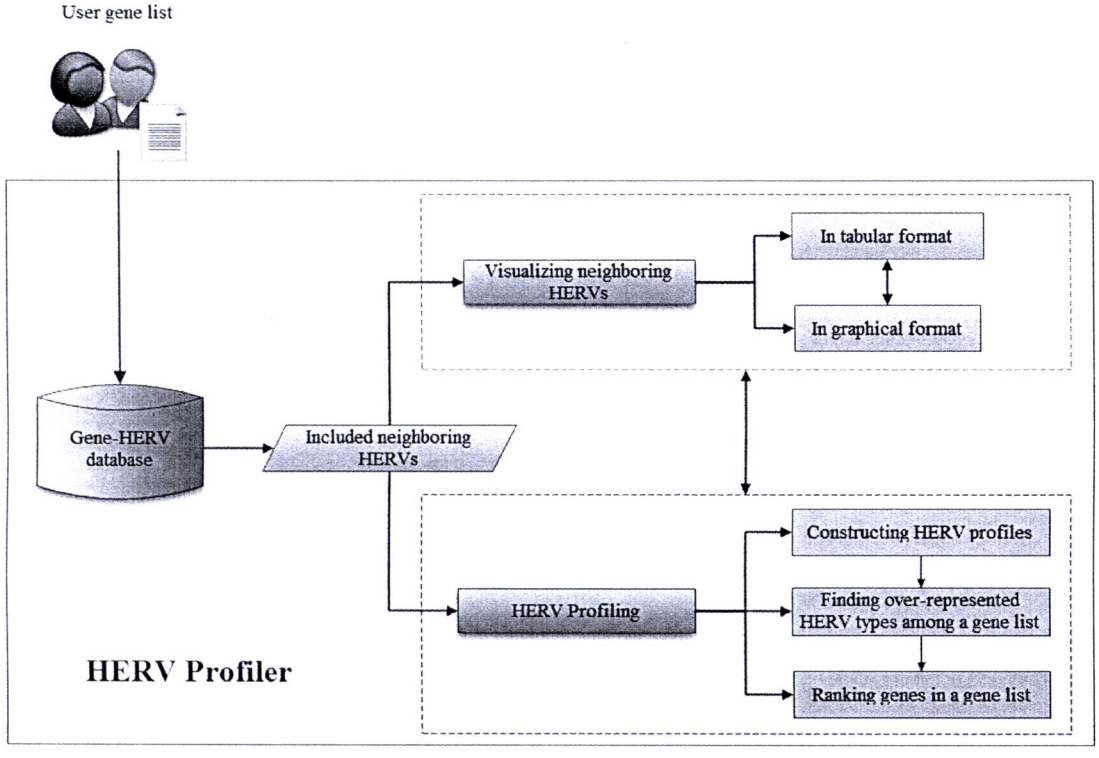


Figure 3.1 Diagram of system flow design

The main purpose of the development of this web-based tool or HERV Profiler is to facilitate investigations of neighboring HERVs of interested genes. There are two main features designed for this tool. The first one is visualizing neighboring HERVs of the input genes (Figure 3.1). This feature would facilitate users to manually investigate the neighboring HERVs of the genes. The second one is to provide the tools for gene list analysis, so-called HERV profiling (Figure 3.1). This feature would purpose potential HERV types which are over-represented among a gene list and also their corresponding genes by performing computational and statistical processes. Also, users can freely switch between the functions of visualizing and profiling.

An input of the tool is a list of genes that users would like to investigate their neighboring HERVs. Instead of including all HERVs to further processes of the tool, HERV Profiler has an option that allows users to manually omit HERVs which they prefer to exclude. This would be beneficial when users aim to study only certain types of the HERVs, such as only a particular superfamily. Only the selected HERVs will be used in both visualizing and profiling.

In the visualizing feature, users can display the neighboring HERVs in either tabular or graphical format. Also, users can freely switch the displaying between these two formats. In terms of the profiling, there are three tasks that need to be completed orderly, including constructing HERV profiles, finding over-represented HERV types, and ranking genes in a gene list. These all would be described in details later.

3.3.2 Functional requirements

Following to the system design, there are two main modules designed for the tool. The first module is to visualize neighboring HERVs of interested genes. The second module is to computationally and statistically analyze neighboring HERVs of a gene list input, so-called HERV profiling. This module would purpose interesting HERV types and genes based on the assumption of being over-represented of particular HERV types among the gene list input. The details of each module are described below.

Module 1: Visualizing neighboring HERVs of interested genes

The objective of this module is to provide visualization of all neighboring HERVs of input genes. There are two ways to visualize all of the information of the query results, either in forms of traditional tables or graphical displays. In the tabular displays, a table would show all neighboring HERVs of a particular gene along with the characteristics of each HERV element. One table is used for displaying only one gene isoform. The higher number of genes input, the higher number of tables displayed. To be responsible for this point, the graphical displays would be thus designed to serve as an alternative choice for the visualizing. In the graphical displays, both a gene and their neighboring HERVs would be shown in an image. This would allow users to easily explore the overview pictures of the neighboring HERVs of the interested genes.

In addition, ranking genes in a gene list is also required for the tool. This could be advantageous when users have to deal with a number of genes investigated. The number of neighboring HERVs is preferred in this situation, because higher number of neighboring HERVs, higher possibility that those neighboring HERVs could be related to those genes. In contrast, the genes having no neighboring HERVs at all should be neglect in the studies that are suspecting on the neighboring HERVs. There are seven options, based on locations relative to genes, in the ranking: rank genes based on the number of HERVs throughout the genes, the number of upstream HERVs, the number of downstream HERVs, the number of in-gene HERVs, the number of intron HERVs, the number of exon HERVs, and none for using ordinary ranking since gene list input.

Furthermore, each gene is required to be linked to an external database for additional information of a gene. In this case, UCSC database was chosen to serve as a cross-linking database of all genes in HERV Profiler, because the genes used in HERV Profiler are all annotated and downloaded from the UCSC resources.

Module 2: Profiling neighboring HERVs of genes in a gene list

This module provides the tools for gene list analysis. There are three sub-modules in the HERV profiling, including constructing of HERV profiles, finding over-represented types, and ranking genes based on their HERV types.

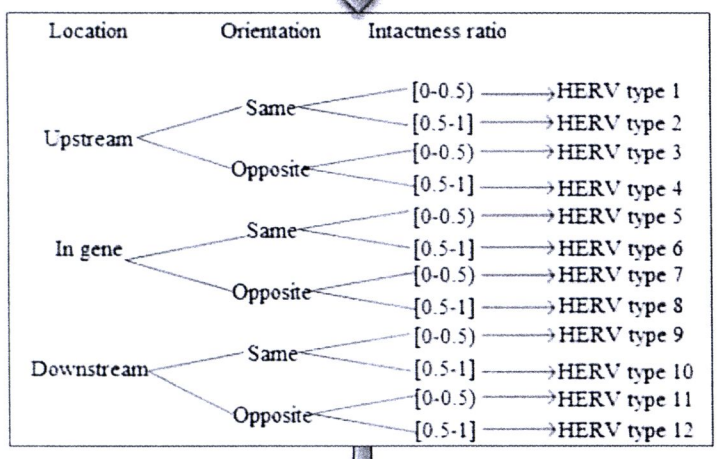
Sub-module 2.1: Constructing HERV profiles

HERV profiles are the primary input of the HERV profiling. This sub-module is thus responsible for constructing the HERV profiles, which would be used in the next sub-modules of the profiling.

The HERV profiles are information of neighboring HERVs that are summarized and manipulated into a particular format which is more suitable to be computationally and statistically analyzed. The input of this sub-module is raw information of neighboring HERVs of the genes input. These pieces of information are the tables containing neighboring HERVs and their characteristics. As mentioned earlier, one table is for one gene isoform. The output of this constructing is a table, so-called HERV profiles, where a row inside contains the summarized information of all neighboring HERVs of one gene.

To construct the HERV profiles, HERV types are required to be defined a priori. HERV types, in this work, are the combinations of independent HERV characteristics. For example, if the selected characteristics are HERV location relative to genes and orientation, there would be six HERV types, including sense-upstream, antisense-upstream, sense-in-gene, antisense-in-gene, sense-downstream, and antisense-downstream. The HERV types previously defined would then be numbered all. After that, we would be able to summarize the raw information of the neighboring HERVs of a gene by counting the number of each HERV types, and fill in the HERV profile table. Lastly, the table or HERV profiles which a row represents the neighboring HERVs of one gene isoform would be obtained, and the numbers in the table indicate the count numbers of the elements following to the HERV types. An example showing how to construct an HERV profile of one gene is illustrated in Figure 3.2.

Gene A			
No. of neighboring HERVs	HERV characteristics		
	Location relative to genes	Orientation	Intactness ratio
1	upstream	same	0.5
2	downstream	opposite	0.25
3	in gene	same	0.1
4	upstream	opposite	0.3



Gene A				
No. of neighboring HERVs	HERV characteristics			
	Location relative to genes	Orientation	Intactness ratio	
1	upstream	same	0.5	HERV type 2
2	downstream	opposite	0.25	HERV type 11
3	in gene	same	0.1	HERV type 5
4	upstream	opposite	0.3	HERV type 3

	HERV type 1	HERV type 2	...	HERV type 11	HERV type 12
Profile of **gene A**	0	1	0	1	0

Figure 3.2 An example to construct HERV profile

The first table in Figure 3.2 is the raw information of the neighboring HERVs of gene A. To construct the HERV profile of this gene, HERV types are defined based on HERV locations relative to genes, orientations, and intactness ratios. Particularly, for the quantitative characteristics, such as the intactness ratio, they need to be categorized instead of using the real value. In this example, the intactness ratios are separated into two classes. The first one is a class of being less than 0.5, and another one is a class of being greater than or equal to 0.5. Thus, there are totally twelve HERV types in this case, as enumerated in Figure 3.2. After defining the HERV types, each neighboring HERV of the gene A would then be able to be categorized following to the HERV types. Finally, we would be able to count the number of each HERV types located

nearby gene A, and summarize into one row instead of one ordinary table. The last obtained table in Figure 3.2 is called the HERV profile of gene A. If there are more than one genes input to this sub-module, the result table or HERV profiles would have more than one row, unlike that shown the example.

Sub-module 2.2: Finding over-represented HERV types

Because HERVs are somewhat different in several points depending on considered characteristics, it is feasible that only some HERVs with some certain characteristics could disturb their neighboring genes under a certain condition. HERV Profiler would thus provide a way to purpose candidates of the HERVs with the preference to the HERVs which are specially occurred in a gene list. This process is performed in sub-module 2.2, after constructing the HERV profiles.

This sub-module aims to identify over-represented HERV types given a list of input gene. Fisher's exact test was used to identify the over-represented types by comparing the present proportions in a gene list to the proportions of whole genome. This test was performed for each HERV types specified in the previous sub-module. The contingency of the test for an HERV type is illustrated in Table 3.1. If at least one gene isoform has a considering HERV type, the gene is considered to associate with the HERV type. The number of genes both related and not related to HERV types would be counted both from user gene list and whole genome. Then, Fisher's exact test would be performed on each contingency table. Each HERV type would obtain its p-value from Fisher's exact test indicating that how significantly they are over-represented among a gene list when comparing to the random chance.

Table 3.1 An example of a contingency table of sub-module 2.2

	User genes	Genome	**Total**
Related to an HERV type	x	n_1	$x+n_1$
Not related to an HERV type	y	n_2	$y+n_2$
Total	x+y	n_1+n_2	$x+y+n_1+n_2$

Sub-module 2.3: Ranking genes based on their HERV types

After identifying over-represented HERV types from the input genes as in the previous sub-module, this sub-module assigns score to each input gene based on the over-represented pattern, and then ranks all of them according to the score assigned. The score is primarily based on two factors, including the number of occurrences of the HERV types and significant level of those HERV types obtained from the sub-module 2.2. Only HERV types considered as significantly over-represented types are included into the score calculation. The equation to calculate the scores is shown below.

$$score\ of\ a\ gene = \sum_{i=1}^{n} x_i\, y_i$$

where

n = the number of HERV types that are considered as significantly over-represented types (the number of types having p-value lower than or equal to the specified maximum p-value)

x_i = the number of occurrences, or in average between isoforms, of the i^{th} HERV type

y_i = score of the i^{th} HERV type obtained from minus log-transformation of its p-value

3.3.3 Overview of system development processes

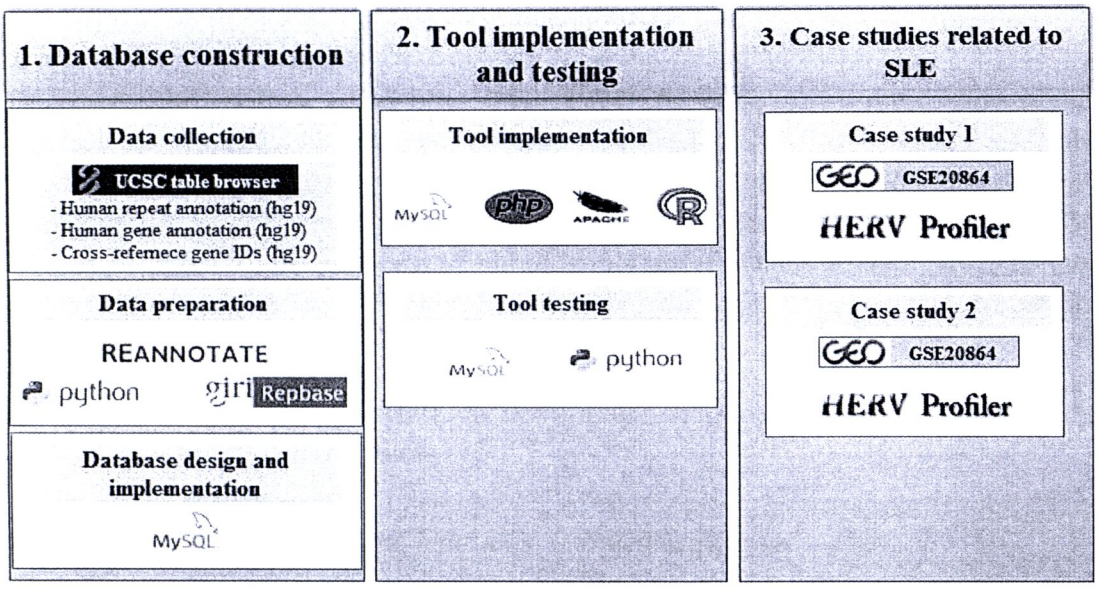


Figure 3.3 Overview processes of system development

There are three main processes in the development of HERV Profiler, including the construction of the database, the implementation and testing of the tool, and the case studies related to SLE (Figure 3.3).

Since the tool requires the database to be properly constructed, the database was implemented first. To construct the database, all of the resources, including the human repeat and gene annotation as well as the cross-reference gene IDs, should be downloaded and collected from the UCSC table browser before [50]. All of the data should be properly prepared and manipulated before storing in the database. REannotate, the python programs, and the resources downloaded from the Repbase were used in the preparation of the data. After the data already properly prepared, all data would then be stored in the database by using MySQL.

To implement the tool, MySQL was used to manage the database of HERV Profiler. Web interfaces were implemented by using PHP, HTML, and APACHE. In terms of the R programs, it was embedded into the system to perform Fisher's exact tests in the sub-module 2.2. After finishing the implementation, tool testing is then performed to check whether the tool functions accurately. MySQL and the python programs were used to generate the gene sets used for the testing.

Finally, to illustrate how efficient the tool is, two case studies related to SLE were demonstrated by using HERV Profiler. These two case studies used the SLE microarray data, with the series name GSE20864, which was downloaded from GEO database. The details of each step are described next.

3.4 Database construction

3.4.1 Data collection

The human repeat annotation, human gene annotation, and also the cross-reference IDs of the UCSC genes were all downloaded from the UCSC table browser [50]. In the UCSC table browser, these three data are in the UCSC table named rmsk, knownGene, and kgXref, respectively. They are all based on the February 2009 human reference sequence (hg19/GRCh37), which includes sequences of 24 main chromosomes (chromosome 1-22, X, and Y), 59 unplaced contigs, and 9 haplotype chromosomes. The haplotype chromosomes are a collection of haplotype segments, additionally generated during the genome assembly [53]. The number of records in all data collected is shown in Table 3.2.

Table 3.2 The number of records in the original downloaded data according to the categories of assembled sequences

Data	Categories of assembled sequences	The number of records (%)
Human repeat annotation	Main placed chromosomes	5,232,237 (98.76%)
	Haplotype chromosomes	55,980 (1.05%)
	Unplaced contigs	9,913 (0.19%)
	Total	**5,298,130**
Human gene annotation/ Cross-reference gene IDs	Main placed chromosomes	73,660 (94.91%)
	Haplotype chromosomes	3,835 (4.94%)
	Unplaced contigs	119 (0.15%)
	Total	**77,614**

3.4.2 Data preparation

Data preparation was performed in order to prepare and manipulate all of the data resources with the aim to be finally stored in the database of HERV Profiler. There are five processes in the data preparation, including data selection, HERV defragmentation, determination of HERV truncation patterns, HERV designation, and mapping HERVs onto the human genes.

3.4.2.1 Data selection

As mentioned before, the annotation results and the cross-reference gene IDs contain not only the records based on the main placed chromosomes, but also based on the unplaced contigs and the haplotype chromosomes, which were separately generated from the main placed contigs. In this work, only the data of the main placed chromosomes would be included in further utilization, due to easier and tidier in referring to the chromosome on which a gene or a repeat is located. This step was thus performed to remove all of the unwanted records in the data resources before performing any manipulation of the data. The details of this step are summarized in Figure 3.4.

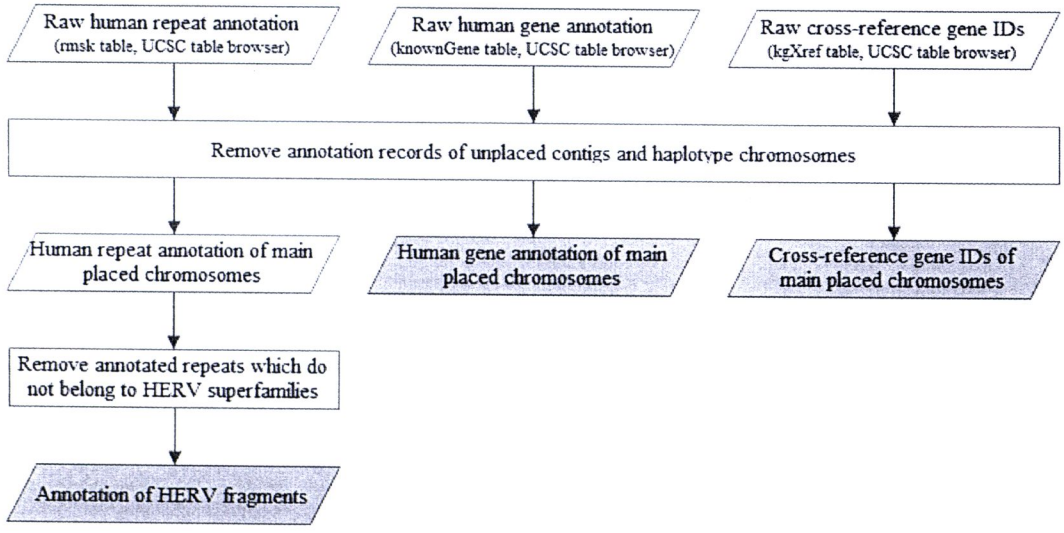


Figure 3.4 Flow chart showing all of the steps in the data selection

According to Figure 3.4, the first time of removing the unwanted records was performed on all of the downloaded data to remove the annotation records based on the unplaced contigs and the haplotype chromosomes. The second time is for only the selected human repeat annotation to remove all of the repeats which do not belong to HERV superfamilies. Finally, the annotation of HERV fragments, the gene annotation and the cross-reference IDs which are based on only 24 main placed chromosomes, including chromosome 1-22, X and Y, were obtained from the data selection. The numbers of records resulting from the data selection are shown in Table 3.3.

In addition, there are six HERV superfamilies found in the annotation of the HERV fragments, including ERV1, ERVK, ERVL, ERVL-MaLR, ERVL?, and ERV. For the

last two, they are used to indicate a fragment which cannot be certainly determined for its superfamily. Thus, all of the fragments annotated belonging to ERVL? and ERV would have been considered as the unclassified fragments in this work. The detailed proportions in the annotation data of the HERV fragments, obtained from the removing unwanted records from the raw human repeat annotation, is shown in Table 3.4.

Table 3.3 The number of selected records from the data resources

Data	The number of records
Annotation of HERV fragments	687,420
Selected human gene annotation	73,660
Selected cross-reference gene IDs	73,660

Table 3.4 Detailed proportions of each HERV superfamily in the HERV annotation data

Original proportions			New proportions		
Superfamily	**The number of records**	**Percentage**	**Superfamily**	**The number of records**	**Percentage**
ERVL-MaLR	343,666	49.99%	ERVL-MaLR	343,666	49.99%
ERV1	172,863	25.15%	ERV1	172,863	25.15%
ERVL	157,992	22.98%	ERVL	157,992	22.98%
ERVK	10,490	1.53%	ERVK	10,490	1.53%
ERVL?	1,800	0.26%	Unclassified ERVs	2,379	0.35%
ERV	579	0.09%			
Total	**687,420**	100%	**Total**	**687,420**	100%

3.4.2.2 HERV defragmentation using REannotate

Generally, a complete HERV element is composed of two flanking LTRs and a set of internal retroviral genes in the central. However, the computational annotation of the HERVs is usually incompletely performed unlike the picture of that complete element, because the LTRs and the internal sequences would be separately detected in the annotation. There are two reasons supporting why the separate detection is required. The first one is to avoid the missing discoveries of the solitary LTRs. Secondly, due to massive accumulation of insertions and deletions on the HERV sequences, the annotation results are often displayed as several fragments of an element instead of one with a long gap [40]. Therefore, to enable observing the HERVs in a view of being the elements, the HERV defragmentation is required to join fragments of the same element into a single element before.

The HERV defragmentation is the post-processing of the computational annotation of the HERVs. Only the annotation data of the HERV fragments, one that was obtained from the data selection, would be manipulated in this step. The numbers and the proportions of the HERV fragments in the annotation data are shown in Figure 3.5.

Furthermore, all of the steps in the defragmentation is summarized and shown in Figure 3.6.

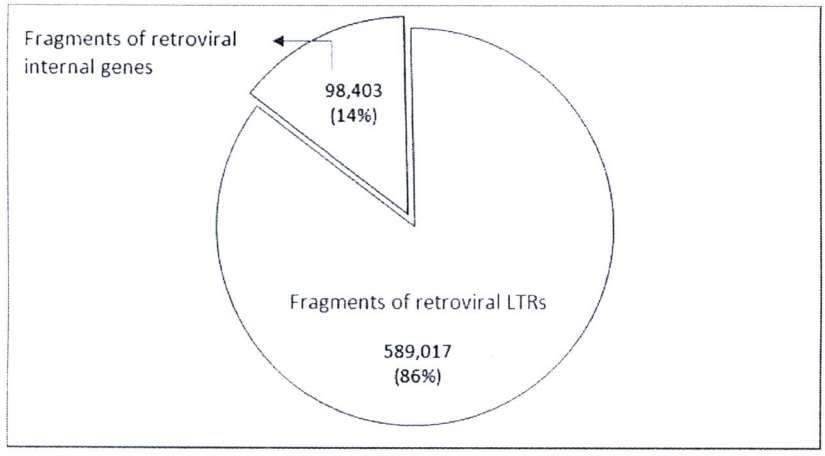


Figure 3.5 The number and proportions of each HERV fragment type

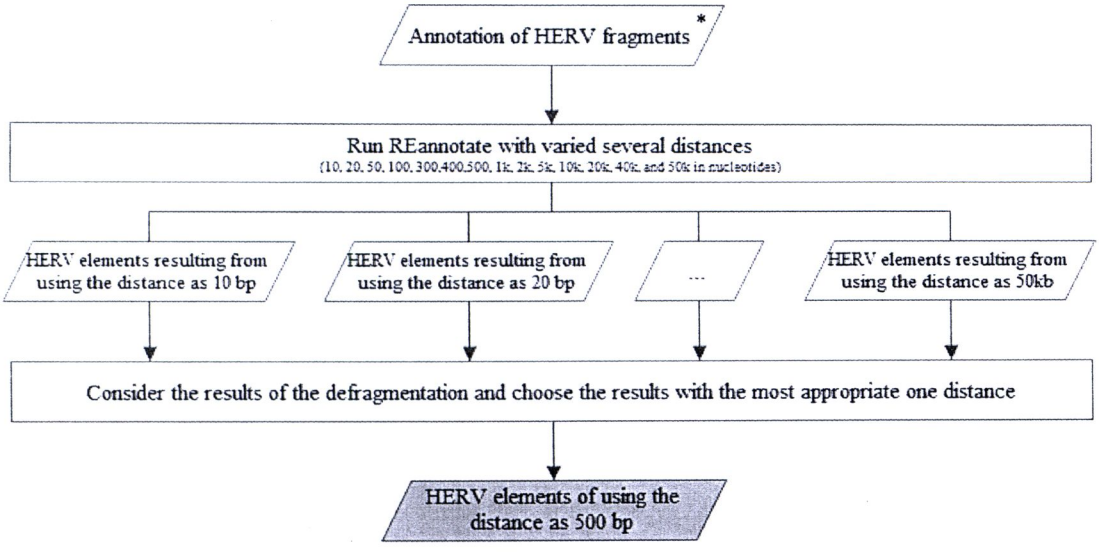


Figure 3.6 Flow chart showing the processes in the HERV defragmentation. A star indicates an output from the previous data selection.

REannotate [41] was used for defragmenting the HERV annotation, by the reason of its more flexibility and result details, additionally computed for each defragmented element. The defragmentation of this program is typically based on distance and orientation between fragments, and membership in the same HERV family. However, a program parameter that is the maximum allowed distance for joining any two fragments should be specified first.

To making the decision to that issue, REannotate would thus be run several times while the maximum distance was varied as 10, 20, 50, 100, 200, 300, 400, 500, 1k, 2k, 5k, 10k, 20k, 40k, and 50k in nucleotides. The number of elements resulting from the defragmentations would be compared to observe the sensitivity of the distance parameters to the defragmentation results. The detailed results of the consideration of the distance parameter can be found in the Section 4.1. It was found that the suitable distance is 500 bp. Therefore, the defragmentation results of using this number of the distance would be chosen and used in further data manipulation.

An example of the defragmentation result provided by REannotate is illustrated in Figure 3.7. Each row of the result represents a defragmented element resulting from the defragmentation. Components of the element are separated into three parts following the complete genomic structure of the HERVs. The complete structure is simply symbolized by LTR1-Int-LTR2, where LTR1, Int, LTR2 represent a left LTR, an internal sequence, and a right LTR, respectively. Furthermore, intactness ratios of each part of the element, indicating how much the part is complete by comparing to the reference sequence, were also provided by REannotate. This value is typically calculated from a fraction of the reference sequence matched in a query sequence. After finished the defragmentation, there are 537,061 elements received.

REannotate id	chromo-some	family	start position of the first hit	end position of LTR1	start position of LTR2	end position of the last hit	LTR1 intactness	Int intactness	LTR2 intactness	orient	superfamily
u1	chr1	THE1B	353590	353949	355514	355871	0.99	1	1	+	LTR/ERVL-MaLR
u2	chr1	THE1B	633822	634179	635744	636103	1	1	0.99	-	LTR/ERVL-MaLR
u3	chr1	THE1C	2871155	2871528	2873101	2873475	0.99	1	1	-	LTR/ERVL-MaLR
u4	chr1	MSTB	2922869	2923299	2924957	2925378	1	0.99	0.99	+	LTR/ERVL-MaLR
u5	chr1	MSTA	4373369	4373658	4375244	4375611	0.84	0.99	1	-	LTR/ERVL-MaLR
u6	chr1	THE1B	4898867	4899223	4900697	4901059	1	1	1	-	LTR/ERVL-MaLR
u7	chr1	THE1B	4917506	4917868	4919451	4919809	1	0.99	0.99	+	LTR/ERVL-MaLR
u8	chr1	THE1B	4960934	4961299	4962872	4963226	1	1	0.99	-	LTR/ERVL-MaLR
u9	chr1	THE1B	4978090	4978454	4980010	4980375	0.98	1	1	-	LTR/ERVL-MaLR
u10	chr1	THE1B	5497521	5497874	5499523	5499878	1	1	1	-	LTR/ERVL-MaLR
u11	chr1	THE1B	5578793	5579156	5580762	5581126	1	1	1	-	LTR/ERVL-MaLR
u12	chr1	MSTA	5588844	5589263	5590872	5591289	1	1	0.99	+	LTR/ERVL-MaLR
u13	chr1	THE1B	6979673	6980033	6981623	6981988	1	1	1	-	LTR/ERVL-MaLR
u14	chr1	THE1C	7850232	7850605	7852140	7852501	1	1	1	-	LTR/ERVL-MaLR
u15	chr1	MSTA	8502491	8502697	8504618	8504978	0.49	1	0.86	+	LTR/ERVL-MaLR
u16	chr1	THE1A	11570858	11571216	11572809	11573166	1	1	1	-	LTR/ERVL-MaLR
u17	chr1	LTR5B	13458305	13459295	13466829	13467826	1	1	1	+	LTR/ERVK
u18	chr1	LTR5B	13679141	13680131	13687665	13688662	1	1	1	+	LTR/ERVK
u19	chr1	THE1D	14173964	14174340	14175877	14175931	1	1	0.14	+	LTR/ERVL-MaLR
u20	chr1	THE1D	14181203	14181562	14183175	14183519	1	0.99	1	+	LTR/ERVL-MaLR

Figure 3.7 An example of the results obtained from the HERV defragmentation by using REannotate. '+' and '-' for the orientation indicate forward and reverse direction of the element on a query sequence, respectively.

3.4.2.3 Determination of types of truncation patterns

As mentioned before, the complete HERVs are usually composed of two flanking LTRs, 5'-LTRs and 3'-LTRs, as well as a set of internal sequences for the retroviral genes. However, due to the accumulations of insertions and deletions on their sequences, most of the HERVs currently present in the human are hardly found as the complete elements, but usually found as the truncated elements. Therefore, assigning

the types of truncation patterns to each HERV element would facilitate us more in categorizing and referring to HERVs according to their structural truncation.

In this work, the types of truncation patterns are defined based on truncated parts of an element, including a 5'-LTR, an internal sequence, and a 3'-LTR. Thus, there are five types for the classification of the truncation patterns: complete, 5'-truncated, 3'-truncated, both 5'- and 3'-truncated elements, as well as solitary or solo LTRs. To classify each HERV element, the result obtained from the step of the HERV defragmentation would be used for the purpose.

According to Figure 3.7, the number of fragments which hit to each part of an element was also provided in the result. Therefore, each HERV element would be easily considered and assigned the truncation type by looking at these numbers. If which part of an element has the number of fragment hits equal to zero, it can be implied that the part of the element is truncated. For example, if the number of the hits to Int of an element is zero, it means that the internal sequence of the element is truncated. However, for the cases of truncation on either left or right LTR, it is required to consider the orientation of the element additionally. The detail in determining the types of truncation patterns of each element is summarized in Figure 3.8. To achieve this step, the Python program was responsible for the determination and performed on the output from the HERV defragmentation. Lastly, the determined types of the elements would be added to the result of the defragmentation as an additional characteristic of the HERVs.

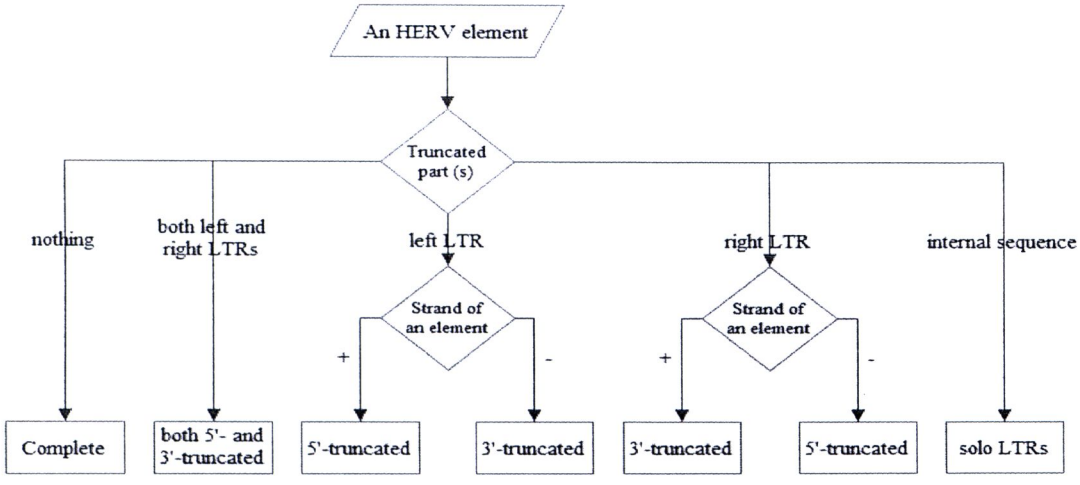


Figure 3.8 Flow chart illustrating the way to determine the type of truncation pattern for an HERV element

3.4.2.4 Manipulation of HERV family names

In running REannotate, one additional input needed, besides the annotation data of the HERVs, is the lists of equivalent family names, provided as an additional text file in [41]. This is required because names of reference LTRs and internal regions of HERVs in Repbase Update [17], a database provided repeat library for repeat annotation, that correspond to the same HERV family may be dissimilar. The first name in each line

would be selected by REannotate to assign a new family name to a defragmented element if it contains fragments of different family names within an equivalent class.

Nevertheless, the result of REannotate is still required additional manipulation afterwards, especially in the issue related to the family names. There are still many redundant family names, different names which correspond to the same family, present in the result. The first cause of it that could be observed was the mistake in text-processing of REannotate that assigned the names with a little bit change in a character to the elements of the same family. All of these mistaken names and their new edited names are listed in Table 3.5. Secondly, the defragmented elements are treated in different ways by REannotate in terms of the family name assigning. Only elements composed of fragments with different equivalent names would be assigned new names, while the others, including fragments no joining and elements originated from joining fragments with the same family names, still use old redundant names. This problem should be solved before storing the data in the database to facilitate us in categorizing and referring to a particular family. All of the steps in manipulating the HERV family names are summarized in Figure 3.9.

Table 3.5 List of the family names assigned by the mistakes in REannotate's text-processing, and their new edited names

Mistaken family names	New edited family names
ERV3-16A3	ERV3-16A3
ERV316A3	
ERV316A3_I	
HERVFc1_LTR1	HERVFc1_LTR1
HERV-Fc1_LTR1	
HERVFc1_LTR2	HERVFc1_LTR2
HERV-Fc1_LTR2	
HERVFc1_LTR3	HERVFc1_LTR3
HERV-Fc1_LTR3	
HERVFc2	HERVFc2
HERV-Fc2	

The aim of performing this step is to manipulate the redundant family names in the REannotate output before being stored in the database. According to Figure 3.9, there are 419 different family names found in total, at first. After editing the mistaken names, as listed in Table 3.5, the number of the remaining names in the list is 413. This name list would be grouped further based on supporting evidences found in the REannotate equivalent name list and the information retrieved from the Repbase Update [17]. The information of each family was extracted from the lines of comments and descriptions in the Repbase Update. On those lines, there are the previous names and the names of paired LTRs and internal regions that could be used to group the redundant names in the

list in addition to the REannotate list as well. If any family name in the list lacks of the evidence supporting its equivalence to other name, the family name would be retained the same in the new created list. This new list of the family names of HERVs found in the human genome is shown in Appendix A. In summary, there are 133 HERV families with 413 names or groups found in the human genome in total.

Lastly, this new list would be referred in editing the family names of each element in the REannotate output. Although the family name would be changed, the old one was not lost, but still kept in the additional field named HERV name/group instead. The edited REannotate output would be used then in further manipulation, and finally stored in the database of HERV Profiler. The summary of the numbers of the types of truncation patterns found is shown in Table 3.6.

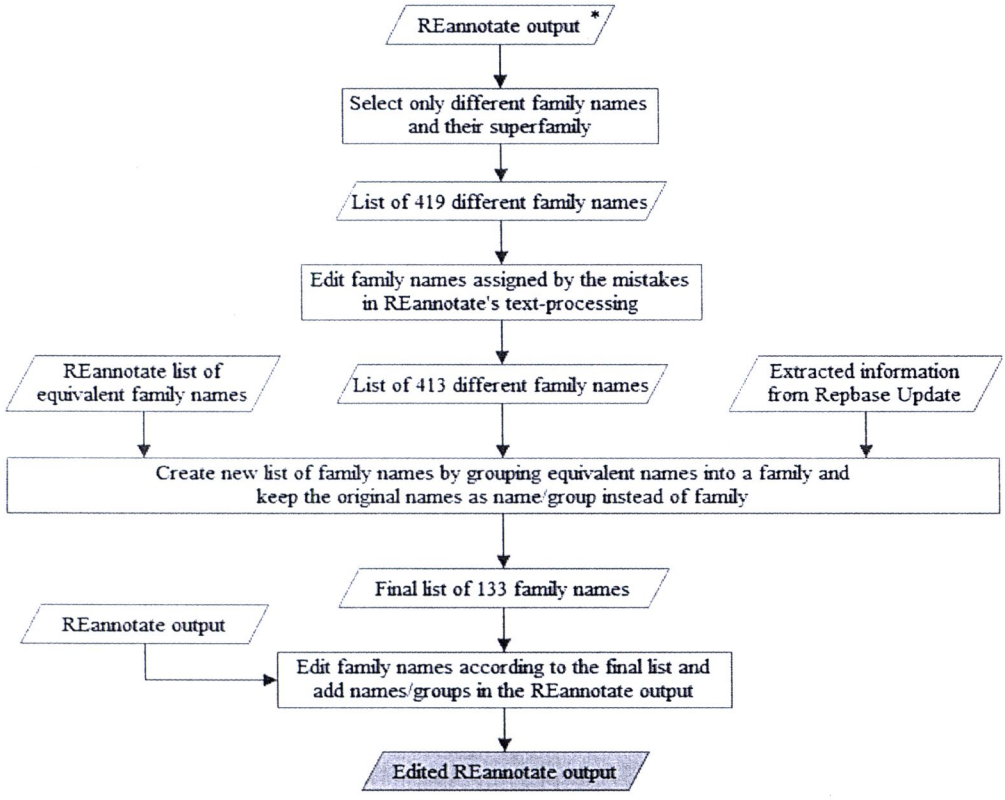


Figure 3.9 Flow chart showing all of the steps to manipulate HERV family names in the REannotate output. A star indicates an output from the HERV defragmentation.

Table 3.6 The numbers and percentages of HERV elements according to each type of truncation patterns

Type of truncation patterns	The number of elements (elements)	Percentage
complete element	7,975	1.48%
5'-truncated element	10,796	2.01%
3'-trunctated element	9,856	1.84%
both 5'- and 3'-truncated element	39,724	7.40%
solo LTRs	468,710	87.27%
Total	537,061	100.00%

3.4.2.5 Mapping HERVs on the human genes

The edited REannoate output and the selected gene annotation were used in this step. Each gene isoform were collected with their neighboring HERV elements which are located not far from the transcription start or transcription termination site more than 100,000 bp. This mapping was done by using pyhon program. The summary of this mapping result is shown in Table 3.7.

Table 3.7 Summary of both the numbers of genes and HERV elements resulting from mapping HERVs on the human genes

HERV elements			Gene isoforms		
gene-neighboring	non-gene-neighboring	Total	HERV-hosting	non-HERV-hosting	Total
382,622 (71.24%)	154,439 (28.76%)	537,061	73,645 (99.98%)	15 (0.02%)	73,660

3.4.3 Database design and implementation

In this step, all of the prepared and manipulated files would be collected in the HERV Profiler database. The entity relationship (ER) diagram of the database is shown in Figure 3.10.

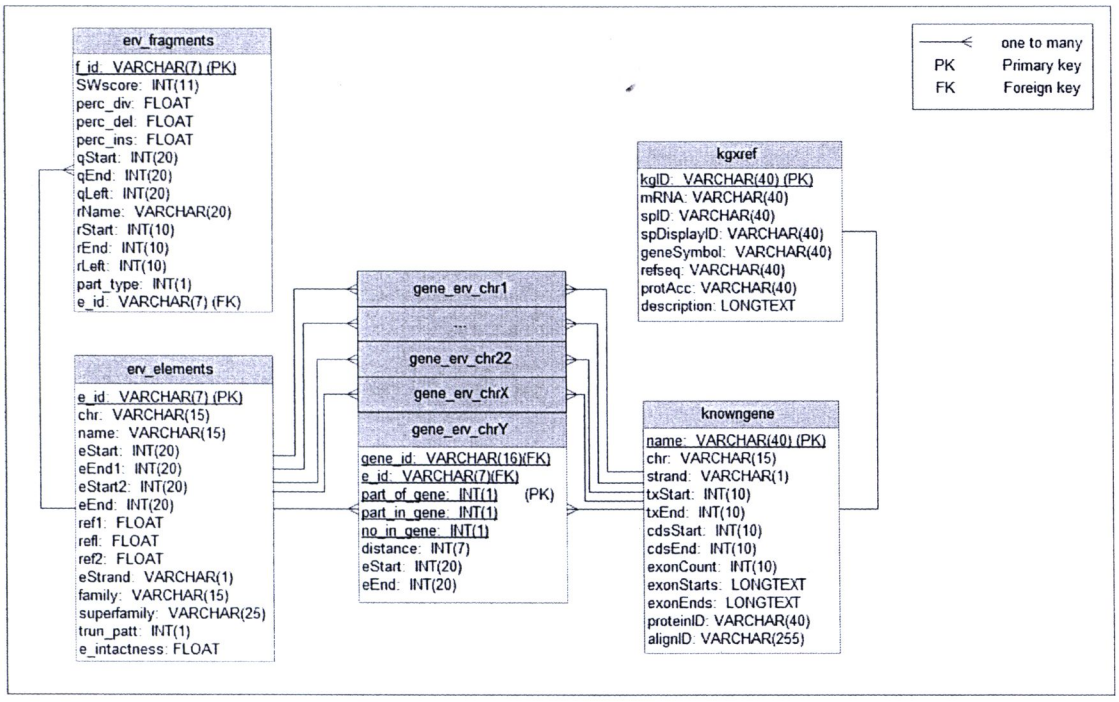


Figure 3.10 Summary of entity relationship diagram in HERV Profiler database

Each entity that is referred to in the diagram is described in detail in Appendix B. MySQL was used to implement the HERV Profiler database. After the implementation was finished, this database would be integrated to the system of HERV Profiler.

3.5 Tool implementation and tool testing

All of the web interface and calculating functions were implemented by using HTML and PHP scripts. After finishing the tool implementation, a process of testing is required to check whether it can function accurately.

For module 1 and sub-module 2.1, these two modules were always tested during the implementation, this is because these two modules are responsible for querying and showing the results obtained from the database, no complex calculating process. Also, for sub-modules 2.2 and 2.3 were tested similarly to those two sub-modules. However, sub-modules 2.2 and 2.3 are related to complex calculating, thus additional testing is required. The methods of testing these two sub-modules are described in detail below.

Testing method for sub-module 2.2

Sub-module 2.2 is related to finding and reporting the HERV types that are over-represented among a gene list input when comparing to the whole genome. Therefore, in the testing of this sub-module, the gene lists with known over-represented HERV types were generated and applied to the tool to observe if the tool would report those types. The work flow showing the methods in generating the gene lists is illustrated in Figure 3.11.

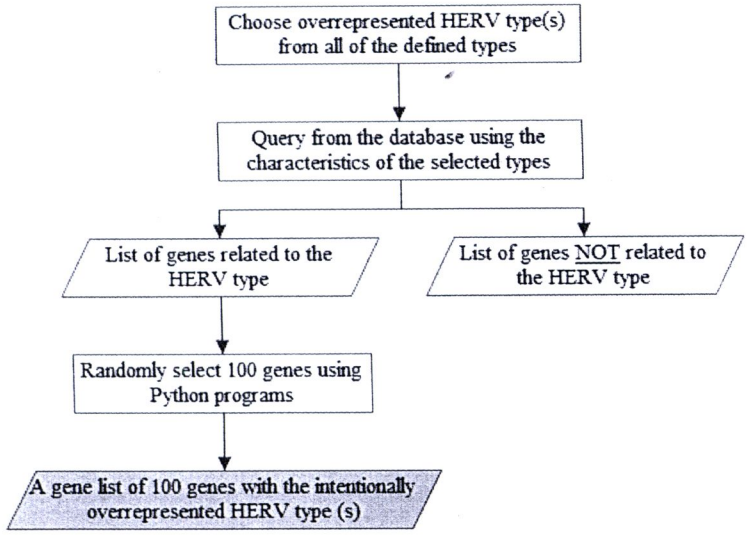


Figure 3.11 Work flow diagram illustrating the way to generate gene lists for testing the sub-module 2.2

In general, to create a gene list for testing sub-module 2.2, it begins with specifying over-represented HERV type (s) first. After the specifying, a set of desired characteristics would be retrieved, and it would be used to query in the database. Then, all of the genes in the genome can be separated into two groups, including the genes that are related to the selected type (s) and not related to the selected type (s). To finally retrieve the gene lists, the genes in a group related to the selected type (s) would be randomly selected 100 genes. It can be certain that the type which is selected before are certainly over-represented among this gene list, because the genes in this gene list are all related to that HERV type.

To define the HERV types for the tool testing, the characteristics of location relative to genes, and additional consideration on in gene location, superfamily, and HERV orientation were selected in the profile construction. Therefore, there are totally 40 HERV types in the case. Then, four gene lists with different over-represented types selected were generated. For the gene lists 1, 2, and 3, only a single HERV type, from 40 types, was arbitrarily chosen as an over-represented type among the gene lists. The HERV types that are chosen for the gene lists 1, 2, and 3 are upstream-ERV1-same, in gene-intron-ERVL-opposite, and downstream-ERVK-opposite, respectively. In terms of the gene list 4, the types chosen are the types that were selected in the generating of the gene lists 1, 2, and 3. It means that there are three types that are over-represented among the gene list 4, including those that are over-represented among the gene lists 1, 2, and 3. With an exception, there are only 54 genes found in overlapping with the three types mentioned above. Thus, the number of genes in the gene list is 54, not 100 like the other gene lists.

After four gene lists were retrieved, all of the gene lists would be applied to the tool with the same selected HERV characteristics in the profile construction. Then, the sub-

module 2.2 would be performed on the generated gene list. The evaluation of this sub-module was done by observing the results from the tool to see whether the known over-represented type (s) was reported by the tool or not. If all of the selected types are reported as found over-represented types by the tool for all gene lists, it can be concluded that this sub-module can work accurately.

Testing methods for sub-module 2.3

This sub-module is working for ranking genes in a gene list by using scores assigned according to being over-represented of their HERVs. Therefore, genes which are related to the over-represented types should have higher scores or being on the better rank than the genes not related to the over-represented types. To test it function accurately, gene lists with known status of being related to the over-represented types were generated and applied to the tool following the way as shown in Figure 3.12.

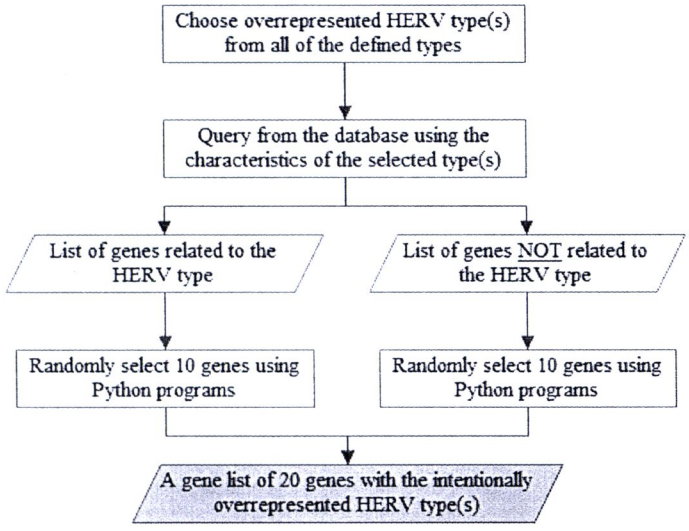


Figure 3.12 Work flow diagram illustrating the way to generate gene lists for testing the sub-module 2.3

Similar to the process of generating gene lists for testing sub-module 2.1, an over-represented type was pre-defined, and used to query in the database. After that, genes in whole genome can be separated into two groups according to their relation to that HERV type. To finally generate the gene lists, 10 from a group related to the type and 10 from a group not related to the type were randomly selected resulting in a gene list of 20 genes.

The characteristics used to preliminarily define the HERV types are similar to those used in generating the gene lists for testing sub-module 2.2. Also, there are totally 40 HERV types according to HERV locations and superfamilies. Four gene lists were also generated and used in this testing. For the gene lists 1, 2, and 3, the selected types are uptrseam-ERVK-opposite, exon-ERVL-same, and downstream-ERVK-same, respectively. In terms of the gene list 4, the known over-represented types were chosen for three types mutually, including upstream-ERVL MaLR-opposite, in gene-intron-

ERV1-opposite, and downstream-ERVL MaLR-same. Each gene list would then be applied to the tool and used to induce performing sub-module 2.3. The expected results, if this sub-module functions accurately, are that the genes that are related to the over-represented type should have higher score or better rank than the genes not related to the selected type. If the results show something like that mentioned for all gene lists, it could be suggested that this sub-module is work accurately now.

3.6 Case studies related to SLE

After the tool would be already implemented and tested, it is necessary to apply the tool to a real utilizing situation. There are two case studies that would be done in this step. Both of the studies would use the same data set, GSE20864 (see Section 3.2.4 for more details). The details of each study would be then described below.

3.6.1 Case study 1: Significance of the number of HERVs under the SLE condition

This case study aims to observe the significance of the number of HERVs under the SLE condition. The question of the study is that whether or not the number of HERVs is related to SLE-relevant genes (significant genes under SLE condition). The methods of microarray data preparation is shown in the below work flow (Figure 3.13)

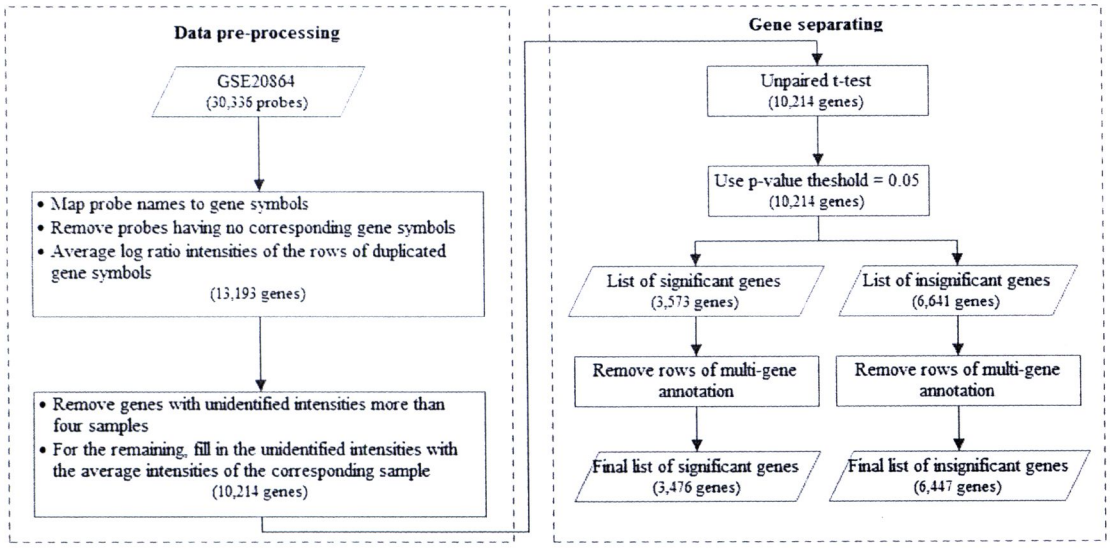


Figure 3.13 Work flow diagram illustrating methods to prepare the microarray data in the case study 1

To prepare the microarray data, there are two main steps inside, including data pre-processing and gene separating. The first step was done to finally retrieve the expression values identified each with gene symbols and without null values. The data pre-processing began with mapping probe names to their corresponding gene symbols. The probes no matched gene symbols would be removed. In terms of probes having the same gene symbols, the average values of them would be used instead. Then, the probes having unidentified expression level more than four samples would be removed from

the data. For the remaining data, the unidentified values would be filled in by an average value of the corresponding sample.

After finishing the data pre-processing, the unpaired t-test would be performed by using p-value equal to 0.05. Thus, the lists of significant genes and insignificant genes would be obtained. Nevertheless, there are some rows identified by multiple gene symbols. These rows would thus be removed from the lists. Finally, the lists of 3,476 significant genes and 6,447 insignificant genes would be retrieved and used.

The top 500 and bottom 500 genes following the p-values were selected and applied to the tool to construct HERV profiles. Then, those profiles were retrieved and applied to the statistical testing to test whether the number of neighboring HERVs is different between those two groups of genes.

In this case study, all of the neighboring HERVs located within 10,000 bp of the covering distance is included into the analysis. The HERV types were defined based on location relative to genes, with separating in-gene location into intron and exon.

3.6.2 Case study 2: Significance of HERV characteristics under the SLE condition

This case study has paid an attention to the characteristics of the neighboring HERVs of a significant gene list under the SLE condition. The hypothesis of this study is that over-represented among a significant gene list could be related to genes possessing them under the SLE condition. The methods of preparing the microarray data is shown in Figure 3.14.

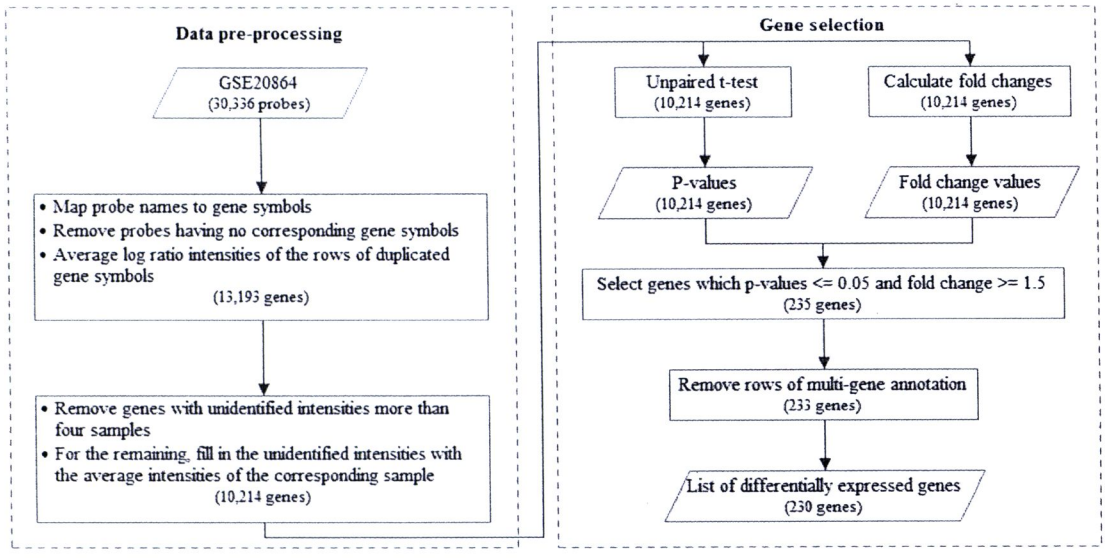


Figure 3.14 Work flow diagram illustrating methods to prepare the microarray data in the case study 2

According to Figure 3.14, the data pre-processing at the right-handed side is similar to the process in the case study 1. In addition to the case study 1, fold change values would

be additionally used to finally select the genes, because there are too many genes when using only p-value to select the genes. Therefore, in this case study, the genes with p-values lower than 0.05 and fold changes higher than 1.5 were selected to be used further in the study. After removing the rows with multi-gene annotation, the 230 significantly differentially expressed genes were retrieved. These genes would then be used as a input in the tool then. Furthermore, to certain the results of this case study, three lists of 230 genes is randomly selected among the whole genome and applied to the tool as well. The results of the random gene lists would be compared to the significant gene list then.

In this case study, all of the neighboring HERVs located within 10,000 bp of the covering distance is included into the analysis. The HERV types were defined based on location relative to genes, with separating in-gene location into intron and exon, family, superfamily, and HERV orientation.