LymAnalyzer User Guide

National University of Ireland,
Galway, Ireland
Contact Email: yuyaxuan0@gmail.com

PART I: What You Need to Have First

1.1 Running Environment

LymAnalyzer is implemented in JAVA. Therefore it's cross platform: You can use it on Mac, Windows or Linux. The only running library you should install in advance is:

JAVA version 1.7 or higher

It can be downloaded here:
http://www.oracle.com/technetwork/java/javase/downloads/index.html

1.2 Software Package Content

LymAnalyzer software package contains two files:

LymAnalyer.jar: This is the command line version of LymAnalyzer.
LymAnalyzer-gui.jar: This is the GUI version of LymAnalyzer.

1.2 Input Data

This tool is mainly designed for processing next generation sequencing data. The data source can come from T cell receptors or antibodies from human or mouse. The required data format of LymAnalyzer is FASTQ. The suffix of the input data should be fastq. There are specific memory usage requirements for datasets with different size. The detailed instructions of how to change the memory setting in Java program will be introduced in Part II.

Dataset size	<128mb	128mb-512mb	512mb-1g	1g – 2g	2g-4g	>4g
Recommended memory setting	512mb	2g	4g	8g	16g	32g

Table 1.1 Memory Configurations

PART II: Run LymAnalyzer

2.1 Run from Command Line

An example of running command is as below shows:

java -jar -Xmx32g Lymanalyzer_cmd.jar /Testfiles/tcrTest.fastq /Resultpath TCRB hs ctest_1 Yes No

The explanation of the parameters:

java -jar	memory	Lymanalyzer_cmd.jar	Inputfile	ResultFolder	ChainType	Species	Name	SNP
1	2	3	4	5	6	7	8	9

MutationTree	TagNumber	ReferenceFolder
10	11	12

1: Run java.

2: Define the maximum memory usage of the Java program. Example: Xmx4g.

3: Run Lymanalyzer.

4: The path of the input dataset, it should be a fastq file.

5: The path of the output folder.

6: The chain type: For T cell receptors, if the sequences are from the beta chain, type in “TCRB”, if it’s the alpha chain, type in “TCRA”. For IGs, if the sequences are from the heavy chain, type in “IGH”, and if it’s from the light chain, type in “IGL”.

7: The species. If the sequences are coming from human, type in “hs”, and if the sequences are from mouse, type in “ms”.

8: The name of the analysis.

9: Type in “Yes” or “No” to determine whether if LymAnalyzer does polymorphism analysis.

10: Type in “Yes” or “No” to determine whether if LymAnalyzer generates the mutation tree for Immunoglobulins.

11 (Optional): User can define the number of detection tags used for alignment ranging from 0 to 10. The default tag number is 5.

12 (Optional): User can define their own Reference gene dataset by simply typing in the path of the folder, which contains all the reference genes.

Different reference genes should be stored as FASTQ format with the filename "species_gene type" in seprate files. The valid filenames are human_TRBV, human_TRAV, human_IGHV, human_IGLV, human_TRBJ, human_TRAJ, human_TRBD, human_IGLJ, mouse_TRBV, mouse_TRAV, mouse_IGHV, mouse_IGLV, mouse_TRBJ, mouse_TRAJ, mouse_TRBD and mouse_IGLJ.

2.2 Run with GUI

Running from GUI version of LymAnalyzer is very straightforward. You can simply click on the LymAnalyzer-gui.jar file. And the console will pop out as below shows. Like the command line version, there are 7 parameters that you need to fill in. The "IG mutation Analysis" is editable only if you choose the chains are from immunoglobulins.

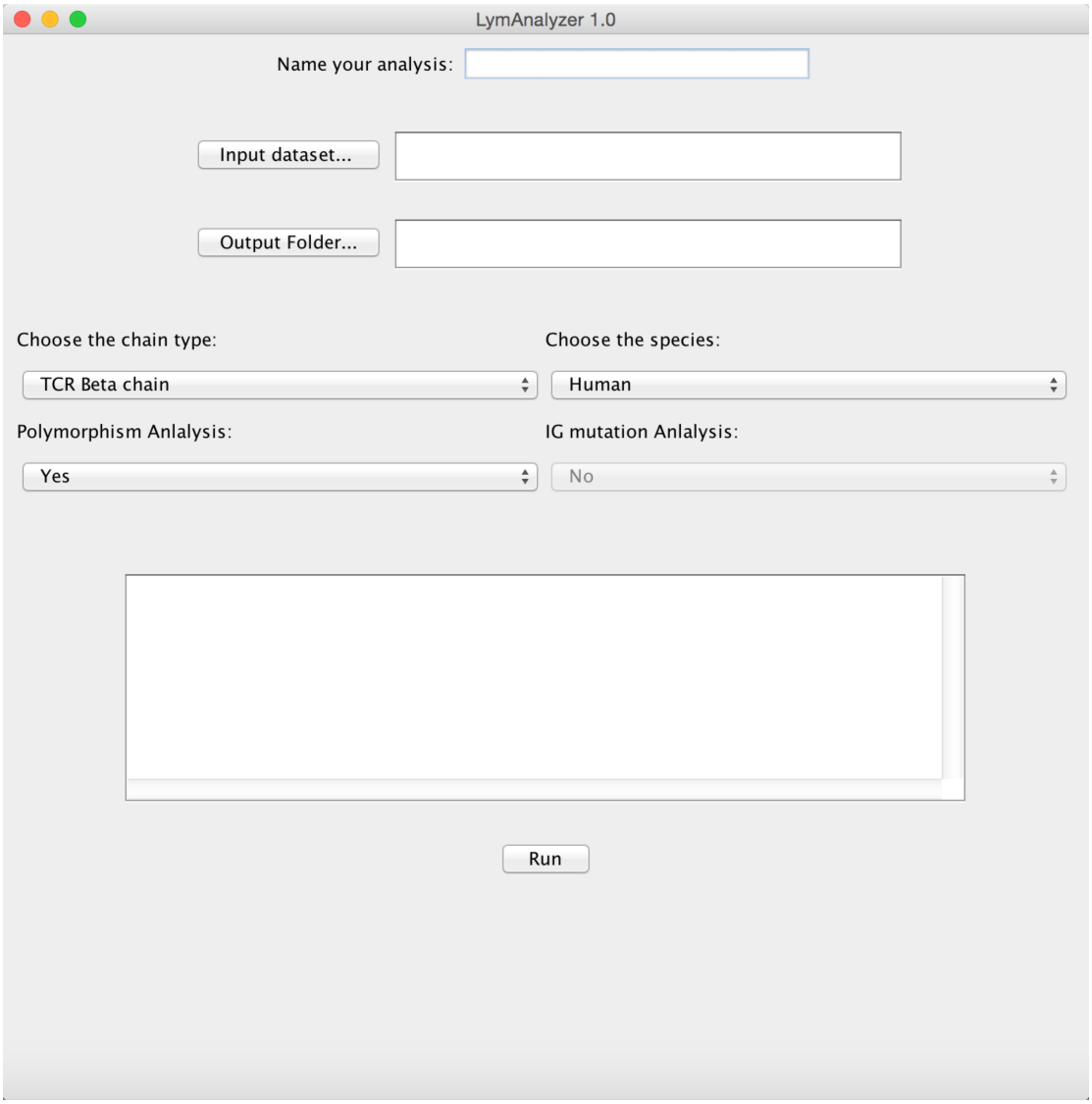

Figure 2.1 Main console of LymAnalyzer

PART III: Results Interpretation

LymAnalyzer generates three types of result files depending on users' setting.

3.1 CDR3 statistic file

Table 3.1 shows the structure of the result statistic file. The input sequences are classified according to their CDR3 sequences. The first column is the number of repeats for such CDR3 sequence. The second column is the amino acid sequence of the CDR3, the third column is the nucleotide sequences of the CDR3 and the last 3 columns are accordingly the reference V, D and J gene where this CDR is coming from.

CDR3 counts	CDR3 AA sequence	CDR3 NN sequence	V gene	D gene	J gene
2936	CATATSGEHTDTQYFG	TGTGCCACCGCGACTAGCGGGGAGCACACAGATACGCAGTATTTTGGC	TRBV27*01	TRBD1*01	TRBJ2-3*01
1393	CASSLAGLPSGRTEAFFG	TGTGCCAGCAGCTTAGCCGGGCTCCCCTCGGGAAGAACTGAAGCTTTCTTTGGA	TRBV13*02, TRBV13*01	TRBD1*01	TRBJ1-1*01

Table 3.1 An example of CDR3 statistic file

3.2 SNP calling results

SNP calling results are saved in Html file, user can view them using browsers. Figure 3.1 shows the representation of a SNP calling file for the J gene. It stores each reference gene, which contains putative SNPs labeled in red. The content labeled in green shows the alternative nucleotide of the reference nucleotide, the frequency and for how many repeats it can be found in the input files.

>TRBJ1-1*01

TGAACACTGAAGCTTTCTTTGGACAAGGCACCAG(G=>T,17.45%,1144)A(A=>G,29.08%,1906)C(C=>G,22.40%,1468)T(T=>A,26.14%,1713)CACAGTTGTAG

>TRBJ2-5*01

ACCAAGAGACCCAGTACTTCGGGCCA(A=>G,19.70%,3552)GGCACG(G=>C,23.14%,4172)C(C=>A,18.77%,3385)G(G=>T,31.30%,5642)G(G=>T,29.00%,5229)C(C=>T,31.08%,5603)T(T=>A,21.99%,3965)C

>TRBJ1-5*01

TAGCAATCAGCCCCAGCATTTTGGTGATGGGACTCG(G=>C,26.18%,1014)A(A=>C,20.26%,785)C(C=>G,28.78%,1115)T(T=>C,18.43%,714)CTCCATCCTAG

>TRBJ2-4*01

AGCCAAAAACATTCAGTACTTCGGCGCCGGGACCCG(G=>C,16.86%,1428)G(G=>T,33.73%,2856)C(C=>G,28.07%,2377)T(T=>G,23.45%,1986)CTCAGTGCTGG

>TRBJ2-1*01

CTCCTACAATGAGCAGTTCTTCGGGCCAGGGACACG(G=>A,23.40%,5125)G(G=>T,30.70%,6722)C(C=>T,38.63%,8459)T(T=>C,15.91%,3484)CACCGTGCTAG

>TRBJ1-2*01

CTAACTATGGCTACACCTTCGGTT(T=>G,18.95%,2234)CGGGGACCAG(G=>A,24.20%,2853)G(G=>C,19.64%,2316)T(T=>G,28.04%,3306)T(T=>A,25.75%,3036)AACCGTTGTAG

Figure 3.1 An example of SNP calling result file

3.3 Ig mutation tree file

Ig mutation tree result file are represented as newick format. For each CDR3 with the default germline configuration, LymAnalyzer creates an Ig mutation tree to display the hyerpmutation process. The result files can be viewed by multiple newick tree viewers like FigTree, Phyfi or NewickViewer. The generation of the mutation is highly depending on the size of the dataset; small dataset may not contain enough sequences to construct such tree.

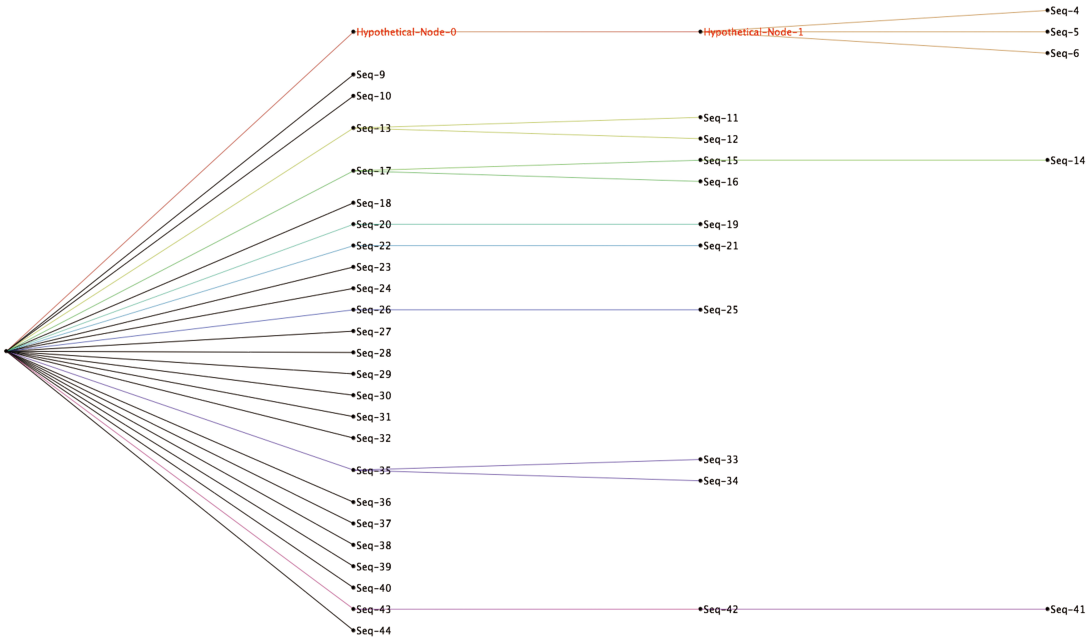

Figure 3.2 Mutation tree viewed by FigTree