

**User and Technical manual**

**for**

**GOssTo: Gene Ontology Semantic  
Similarity Tool**

**Version 1**

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	The idea of GOssTo . . . . .	3
1.2	Contents of the Package . . . . .	4
1.3	This Document . . . . .	4
1.4	Vocabulary . . . . .	4
1.5	References . . . . .	5
<b>2</b>	<b>Assumptions</b>	<b>6</b>
2.1	Technical Assumptions . . . . .	6
2.2	Input Assumptions . . . . .	6
2.3	Performance . . . . .	7
2.4	Output considerations . . . . .	8
2.4.1	GOssTo and memory usage . . . . .	9
2.5	Output assumptions . . . . .	10
<b>3</b>	<b>Using GOssTo</b>	<b>11</b>
3.1	Using Console Parameters in the standalone version of GOssTo . . . . .	11
3.2	GOssToWeb . . . . .	12
3.3	How to add a new semantic similarity measure . . . . .	13
3.4	Example . . . . .	14
3.4.1	Step 1. Class definition and constructor . . . . .	15
3.4.2	Step 2. Define the required methods. . . . .	16
3.5	How to add a new extension measure . . . . .	17
<b>4</b>	<b>Walkthrough</b>	<b>19</b>
4.1	Step by Step Guide to Using the Application . . . . .	19
4.2	User FAQ . . . . .	24
4.3	Using GOssTo as a Library . . . . .	25
<b>5</b>	<b>Error Messages</b>	<b>26</b>
5.1	Errors Relating to Console Parameters . . . . .	26
5.2	Errors Relating to Input Validation . . . . .	27
5.3	Errors Relating to Calculation and Parsing . . . . .	28
5.4	Other Errors . . . . .	29
5.4.1	Preamble . . . . .	31
5.4.2	Terms and conditions . . . . .	32

# 1 Introduction

Welcome to the manual for GOssTo; the Gene Ontology Semantic Similarity TOol.

The GOssTo application is a practical and fully functional implementation of the ideas set out in Yang *et al.*'s paper 'Improving GO semantic similarity measures by exploring the ontology beneath the terms and modelling uncertainty'. The purpose of this software is to compare biological data and produce a meaningful metric describing the likeness or similarity of the biological data in question.

The application will take files from the Gene Ontology Project as input and calculate semantic similarity values based off of six initial, and previously established, 'host' similarity measures (HSM). The 'integrated similarity measure' or ISM will then use the results of the HSMs and calculate a more meaningful value using the formulae set out in the aforementioned paper.

## 1.1 The idea of GOssTo

GOssTo was created as a standalone tool to provide a reference implementation for the Random Walk-based measure of Yang et al. Given the fact that Yang's approach requires a "host" similarity measure (HSM) as the starting point, we also implemented a few of the most popular semantic similarity measures, such as Resnik, Lin, Jiang, etc. This will allow the user two things:

1. To run one of HSMs for a particular annotation of an organism and a certain version of the Gene Ontology.
2. To run Yang's et al. ISM over the same data.

Our idea, by providing a standalone tool, was to give the user the opportunity to run their own experiments providing his/her GOA files and preferred version of the Gene Ontology, allowing the selection of large variety of different parameters. Given the way ISM was conceived, we were also able to compute HSMs without the need to obtain the improved similarity measure.

As stated, it is clear that GOssTo does not pretend to be an exhaustive implementation of all known semantic similarity measures, as it is conceived as a proof of concept implementation of Yang's et al. Random Walk Contribution measure.

We have designed GOssTo as an extensible tool. Users can add new similarity semantic similarity measures and add-ons, taking advantage of the large code base available. In order to do that, we have added a section in entitled "How to implement your own semantic similarity measure" which provides detailed instructions on how to extend

GOssTo. If you have developed a new similarity measure for GOssTo, please submit it to support@paccanarolab.org, we will test it and include it in the next release.

## 1.2 Contents of the Package

The GOssTo package contains an executable .Jar file encompassing all the source files for the program and from which the application will be launched. Also contained within the package are a selection of GOA annotation files and a copy of the gene ontology .OBO file from which similarity values can be readily calculated. These files may need to be updated from the Gene Ontology Project website (<http://www.geneontology.org/GO.downloads.shtml>) as they are regularly updated with new data. Using out of date files may lead to the generation of inaccurate results.

## 1.3 This Document

The purpose of this document is to aid the proper use of and provide a comprehensive understanding of the GOssTo application that will lead users to become professional users of the package. This document will offer detailed information on any and all assumptions made in the construction of the program, information regarding licensing, details of planned or developed extensions to the software and a full walkthrough and technical detail section to aid users getting to grips with the system.

## 1.4 Vocabulary

This section further details some of the more complex technical language or abbreviations used within this document.

- **GOssTo:** The abbreviated name of this software package, standing for: Gene Ontology Semantic Similarity TOol.
- **OBO:** OBO is the name of the flat file format used by the Gene Ontology project to store the central ontological data. The format uses a tag structure with the simultaneous goals of being easily parsed and readable by humans.
- **GOA:** GOA is the name of the file format or plain text structure used for the storage of Annotation files. Unlike OBO, the GOA format is not meant to be read by humans, and so all annotations data is represented by a single line per annotation, with individual data points separated within by tab spaces.
- **HSM:** The abbreviation HSM stands for 'Host Similarity Measure', and refers to previously established formulae for semantic similarity.
- **ISM:** The abbreviation ISM stands for 'Integrated Similarity Measure' in reference to the measure proposed by Yang et al.

- **IO:** An abbreviation for Input Output.
- **Genewise:** A term referring to genes as the base data for calculation.
- **Termwise:** A term referring to GO terms as the base data for calculation.

## 1.5 References

Detailed below are references to any papers or works mentioned or directly referenced within this document and also to the websites of any related or helpful material.

*Papers:*

- Yang, H. *et al* (2012). Improving GO semantic similarity measures by exploring the ontology beneath the terms and modelling uncertainty. *Bioinformatics*, vol. **28**, iss. 10, pp. 1383-1389

*Websites:*

- **Gene Ontology Project:** <http://geneontology.org/>
- **GNU Licenses:** <http://www.gnu.org/licenses/>
- **Paccanaro Lab:** <http://www.paccanarolab.org/>
- **Semantic Similarity Paper (Yang et al.):**  
<http://bioinformatics.oxfordjournals.org/content/28/10/1383>

## 2 Assumptions

### 2.1 Technical Assumptions

There are several technical assumptions that have been made in relation to the development of this application. As such, all details with respect to the programming language, version numbers and associated packages used within the application are listed below along with the specifications of the development environment.

If you have any problems running GOssTo, please first check the data listed below against your own software or hardware and mention in if you need to contact the developers.

*Language and Package Details:*

- **Programming Language:** Java, version 1.6.0\_24
- **IDE:** Eclipse Juno, version 4.2
- **External Packages:** JAMA, version 1.0.3

*Development Environment:*

- **OS:** Xfce 4.8 64bit
- **RAM:** 3.8GB
- **CPU:** Intel Core i5 CPU (4 x 2.67GHz)
- **CPU Cache:** 8162KB

### 2.2 Input Assumptions

All input for the application will be taken either by command line parameters upon initiating the application from the Jar file, or requested by the application once it has started. These inputs will take the form of file paths for the OBO and GOA files, text input of GO terms or gene IDs and numerals or booleans representing choices presented by the application.

The inputs that the application offers the user are as follows:

- File path location for OBO and GOA files
- Choice of evidence codes
- Choice of Gene Ontology relations to be used

- Choice of HSM measure
- Decision of whether or not to compute the ISM for the selected HSM
- Which ontology(s) to utilise
- Whether or not to do term based or gene based similarity
- Names for the output files
- Choice of using individual GO terms or Gene IDs

More information regarding the OBO and GOA file formats can be found here:  
<http://geneontology.org/GO.contents.doc.shtml#file>

## 2.3 Performance

The length of computation time naturally varies dependent on the choice of HSM, whether or not an ISM is to be computed, how many files are to be output and the size of the GOA annotation file being used for the calculation.

For the calculation of just an HSM, it should take in the region of 1.5-2 minutes dependent on the choice of annotation file. For an ISM it should take between 6.5-7.5 minutes. For gene based similarity calculations, the time taken is entirely dependent on the number of genes you wish to calculate the semantic similarity for, the less genes the quicker the computation.

When tested using the Yeast annotation file for term-wise semantic similarity calculation, the application took 83 seconds to calculate solely the HSMs and 6 minutes and 7 seconds to calculate the ISMs.

With respect to memory requirements, they are entirely dependent on the number of terms or genes considered in the calculation. Given that we store similarities using single-precision floating point matrices, every cell in the matrix would require 4 bytes. Memory usage for a few organisms is presented, in GigaBytes, in the following table:

Organism	Termwise	Genewise
Arabidopsis	0.24	1.85
Rat	0.48	0.52
Mouse	1.14	5.72
Fly	0.28	0.77
Yeast	0.26	0.60

To increase the memory allocated use the `-Xmx` parameter when launching the execution. From Java's manual (<http://docs.oracle.com/javase/6/docs/technotes/tools/windows/java.html>):

`-Xmxn` Specify the maximum size, in bytes, of the memory allocation pool. This value must a multiple of 1024 greater than 2MB. Append the letter k or

K to indicate kilobytes, or m or M to indicate megabytes. The default value is chosen at runtime based on system configuration. For more information, see HotSpot Ergonomics Examples: `-Xmx83886080 -Xmx81920k -Xmx80m`

The following example is based on the GOssTo and the example data available at: <http://www.paccanarolab.org/gossto/>. To run the example with 3 GigaBytes of memory, we modify the command as follows: `ava -Xmx3000 -jar Gossto.jar -calculationdata termwise -calculationtype ism -evidencecodes EXP,IDA,IPI,IMP,IGI,IEP,TAS,IC -goapath gene_association.goa_yeast -obopath gene_ontology_ext.obo -hsm Resnik -hsmoutput demo_-hsm_output -ismoutput demo_ism_output -ontology all -relations is_a,part_of -weightedJaccard true -terms all`

This would make 3 GB of memory available to GOssTo. Keep in mind that the total memory in the system is shared by all running applications, so increasing the available memory for GOssTo might have an impact on the overall performance of your system. For larger computations, you might want to consider using GOssToWeb: [www.paccanarolab.org/gosstoweb](http://www.paccanarolab.org/gosstoweb)

These figures are acknowledging that the application calculates the similarity for all three ontologies in each instance of its running.

## 2.4 Output considerations

It is important to clarify the nature of the results in light of the semantic similarity calculations performed by GOssTo. Particularly, given that there are combinations of parameters that might result in a matrix of zeroes.

To explain this phenomenon, we will resort to the well-known Resnik similarity measure.

Resnik computes the similarity of two nodes in an ontology by computing the information content of the lowest common ancestor of two given nodes. By lowest common ancestor we mean the first ancestor node shared by both nodes in question.

We define information content of the node as the minus base 10 logarithm of the fraction of genes annotated by that node and the total number of annotated genes, that is:  $-\log \frac{\#genes\_in\_node}{\#total\_genes}$ . The information content will be zero every time the fraction  $\frac{\#genes\_in\_node}{\#total\_genes}$  is 1. This is equivalent to say that there is, in fact, no informatin content whatsoever in a node that annotates *all* genes.

The true-path-rule states that every gene annotated by a particular node in the Gene Ontology is also annotated by all ancestors of that particular node. This means that the root of the GO DAG will annotate *all* genes in that particular ontology, which means that the number of genes annotated by the root of the GO DAG will be the same as the total number of annotated genes.

Therefore, everytime the user obtains a matrix composed of zeroes, the genes represented in said matrix did not share any ancestors other than the root of the ontology itself. We illustrate this case in the following figure:

Figure 2.1 represents a toy example, in which only four nodes of the Gene Ontology are annotated. Consider two distinct cases, the first one represented by the blue circles

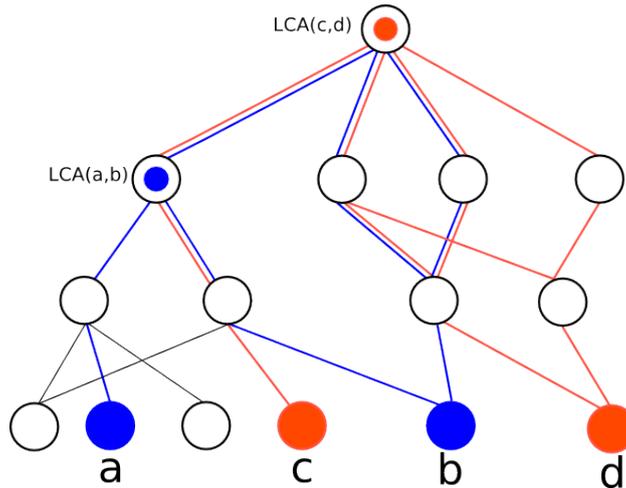


Figure 2.1: Lowest common ancestors and information content

(nodes  $a$  and  $b$ ) and the second one represented by the orange circles (nodes  $c$  and  $d$ ). The colour lines, blue and orange respectively, represent the true path from each node.

If we first analyse the blue case, by following the blue lines we can reach the lowest common ancestor, represented by the node  $LCA(a, b)$ . In this case, the number of genes annotated by the lower common ancestor of nodes  $a$  and  $b$  will definitely not annotate all the genes. In fact, the genes annotated by the orange node  $d$  are not annotated by  $LCA(a, b)$ . This means that the fraction,  $\frac{\#genes\_in\_node}{\#total\_genes} \neq 1$  will definitely be different than 1 for the node  $LCA(a, b)$ , leading to a non-zero information content and thus to a non-zero Resnik similarity.

The other case, represented by the orange circles, we can, by following the orange lines in this case, reach the lowest common ancestor of nodes  $c$  and  $d$ . This node, represented by node  $LCA(c, d)$ , happens to be the root node of this example graph. The fact that this node is the root means that, due to the true path rule, it is in fact annotating *all* genes. This results in  $\frac{\#genes\_in\_node}{\#total\_genes} = 1$ , which means that the information content of it is, effectively, zero.

The user must, therefore, not be surprised in the cases in which he obtains a matrix composed exclusively of zeroes, for it simply means that the nodes do not share relevant information.

### 2.4.1 GOsTo and memory usage

GOsTo can potentially use a lot of RAM memory. Most of the calculations are done in memory and the result is kept there until printed, for efficiency reasons. Therefore, it is likely that some of the “large” eukaryots (cow, human, etc.) will require a lot of memory, especially while doing genewise computations. We have, however, tried to reduce the memory footprint to a minimum, doing all the computations in single-precision floating point. Nevertheless, in some special cases, an error like this could be shown:

ERROR: GOssTo has run out of memory: the memory allocated is insufficient for this calculation. To increase the memory allocated use the `-Xmx` parameter when launching the execution. For example use: `java -Xmx3000M -jar gossto.jar` to increase the maximum available memory to 3 Gb. Alternatively, please use the web service at [www.paccanarolab.org/gosstoweb](http://www.paccanarolab.org/gosstoweb)

Please refer to section [2.3](#) for a detail explanation on how to address this issue.

## 2.5 Output assumptions

The output produced by the package is in the form of plain text files. Each file is named by the user and contains an N by N matrix of decimal similarity figures where N is the number of GO terms or Genes used in the calculation. The similarity values are separated by commas and there is an axis of terms, GO ids or Gene ids, along the top and down the left hand side. This allows the output to be both humanly readable and easily parsable by other applications.

However, for results files involving many terms its readability will suffer somewhat due to the quantity of numbers and the proximity.

## 3 Using GOssTo

### 3.1 Using Console Parameters in the standalone version of GOssTo

The command line application offers the ability to enter all parameters the application would normally prompt for at the point of execution. This increases the speed at which the program can be used by removing this prompt section and starting the calculations straight away.

To utilise this feature, in the line with which you execute the applications .Jar file, after typing the name of the .Jar simply enter the parameters by typing the parameter label in the form of '-parameter' followed by a space and the appropriate parameter value. The available parameters are listed below:

#### *Administrative Paramaters:*

These parameters takes no arguments; you don't follow them with any values.

- **-help:** The help parameter brings up a list of all the parameters with their descriptions.
- **-version:** Displays the version of the GOssTo release.
- **-getw:** Displays the section of the licence pertaining to the warranty, or lack thereof.
- **-logfile:** Creates a log file containing notes on the various stages of completion of the GOssTo execution. Used in addition to a full set of calculation parameters.

#### *Calculation Parameters:*

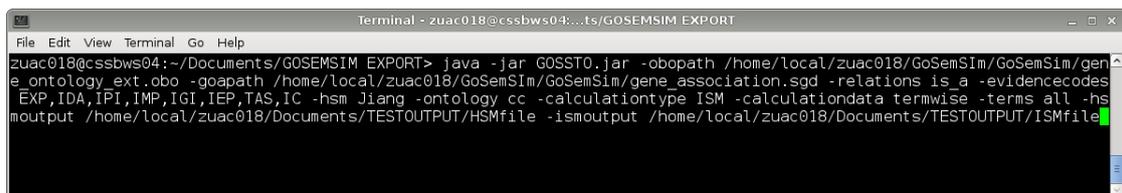
These parameters DO take arguments; you MUST follow them with the appropriate values described below.

- **-obopath:** Follow this label with the file path of the OBO file you wish to use, no spaces are allowed in this file path.
- **-goapath:** Follow this label with the file path of the GOA file you wish to use, no spaces are allowed in this file path.
- **-relations:** Follow this label with the Gene Ontology relations you wish to use in the calculation. If more than one, separate by a comma only. No Spaces.

- **-evidencecodes:** Follow this label with the evidence codes you wish to utilise. Again comma separated and no spaces.
- **-hsm:** Follow this label with the name of the HSM you wish to use. The name of the HSM will be case sensitive so please enter the name of the HSM exactly as its class name in its java file.
- **-ontology:** Follow this label with either 'bp' to use the Biological Process ontology, 'mf' for Molecular function, 'cc' for Cellular Component or 'all' for all three. Do not enter the inverted commas.
- **-calculationtype:** Follow this label with simply 'HSM' or 'ISM' depending upon which you would like to calculate.
- **-calculationdata:** Follow this label with 'termwise' to calculate semantic similarity by GO terms, or 'genewise' to calculate by gene.
- **-terms:** Follow this label with the ID's of any terms or genes you wish to focus the calculation on, comma separated. If you do not wish to use any specific terms, follow the label with 'all', without the inverted commas.
- **-hsmoutput:** Follow this label with the name you wish to entitle the HSM results file with. Please do not put a file extension in your name, result files will be '.txt' text files by default.
- **-ismoutput:** Follow this label with the name you wish to entitle the ISM results file with. Please do not put a file extension in your name, result files will be '.txt' text files by default. This field is OPTIONAL if calculating solely HSM values.

### Usage Example:

The following is an example of the correct entering of the console parameters:



```

Terminal - zuac018@cssbws04:...ts/GOSEMSIM EXPORT
File Edit View Terminal Go Help
zuac018@cssbws04:~/Documents/GOSEMSIM_EXPORT> java -jar GOSSTO.jar -obopath /home/local/zuac018/GoSemSim/GoSemSim/gen
e_ontology_ext.obo -goopath /home/local/zuac018/GoSemSim/GoSemSim/gene_association.sgd -relations is_a -evidencecodes
EXP,IDA,IPI,IMP,IGI,IEP,TAS,IC -hsm Jiang -ontology cc -calculationtype ISM -calculationdata termwise -terms all -hs
moutput /home/local/zuac018/Documents/TESTOUTPUT/HSMfile -ismoutput /home/local/zuac018/Documents/TESTOUTPUT/ISMfile

```

If you receive any error messages when using the console parameters, they can be referred to the Error Messages section at the end of the manual. Suggested action is listed for each error.

## 3.2 GOssToWeb

GOssToWeb provides an intuitive and powerful web interface for GOssTo. Available at <http://www.paccanarolab.org/gosstoweb>, it allows extensible configuration of the experiment and provides features such as:

- Email notification
- Full parameter validation
- Up to date organism annotation files and ontologies.

GOssToWeb provides an ideal tool for occasional calculations. We keep the version of GOssToWeb consistent to the provided Java version to ensure identical results.

### 3.3 How to add a new semantic similarity measure

We will use as a guide the implementation of the Resnik measure which can be found in the file Resnik.java of the HSM package.

1. Download the last version of the source code from: <https://github.com/pwac092/gossto>.
2. Add a new Java class in the package HSM. The name given here will be the name used to identify the measure from the command line tool (so, the "class Resnik" in "Resnik.java" is referred to as "Resnik", if you want to implement a "class ImprovedResnik", you will be able to invoke it using "ImprovedResnik").
3. Inherit from class "HSM", and complete the required methods. The constructor will probably remain empty.
4. Now, we need to implement two methods: the termwise semantic similarity (i.e., the similarity between two terms) and the genewise semantic similarity (given two genes, which is the semantic similarity between them).
5. To implement the termwise semantic similarity, we are only given a parameter, an integer showing the ontology (0=BP, 1=MF, 2=CC). Note that we assume that the three ontologies are independent, and we will ignore inter-ontology relationships. The structure matrixAxis is a double array of GOTerm objects, which is first indexed by ontology, and secondly by an integer in the interval 0 to numGOtermsPerOntology[ontology] (without including this last one). Once retrieved, every GOTerm object can provide access to the GO term identifier, a pointer to the ontology, the parents and the descendants for each one of the relationships, and other miscellaneous data. In order to store the computed semantic similarity values, we will have to create an NxN Matrix, where N is the number of terms of this ontology. Therefore, we will iterate over every pair of GOTerms, using the two nested for loops, and computing the semantic similarity using this measure. We have provided a function lowestCommonAncestor, implemented in the HSM class, which returns the number of annotations given to the lowest common ancestor in the specified ontology and annotation file, divided by the maximum number of annotations of that ontology.
6. Most of the genewise implementations only require, for every two genes, the computation of a function of all the pairwise values of the termwise Gene Ontology of

the terms associated to both genes. Given the fact that this is independent of the proper implementation of the measure, we have implemented a method "geneWiseSimilarityByMaximum" which computes the maximum of all values given by all possible pair of GO terms where the first one is associated to the first gene and the second one to the second. If you prefer to compute the average or other aggregate function, you can implement your own method using the mentioned one as a base. Please note that this method is optimized and makes use of some optimizations, finding which rows dominate others, hence saving computation time. If you use another measure be careful of not using this optimization, as it may not suit your needs.

Once GOssTo has been recompiled, the new HSM will be available for use. To use this new HSM; when prompted by the program for your choice of HSM, input the name of the new HSM and the application will look for it within its classes and if found, will use it as the HSM for the upcoming calculation.

Note that if your HSM does genewise semantic similarity calculations, it will be told to use the specialised gene ISM implementation strategy. If you do not wish to use this, follow the instructions to change the source code above and in the source code itself.

If you have developed a new similarity measure, you can submit the GOssTo-friendly version of the implementation, and we will do our best to evaluate it quickly and include it in the next release of GOssTo.

There is one last consideration. While the code we provide is licenced under GPLv3, licensing of individual component packages have to be respected.

### 3.4 Example

In order to clarify the previous explanation, we will present an example based on Resnik's Semantic similarity measure.

```
public class Resnik extends HSM {

    public Resnik(GOTerm[] allTerms, String[] genes, String[][] goIds,
        GOTerm[][] axis, String[] targets, Assignment annotations,
        String[] relations, TinyLogger logw) {

        super(allTerms, genes, goIds, axis, targets, annotations,
            relations, logw);
    }

    @Override
    public Matrix calculateTermWiseSemanticSimilarity(int ontology)
        throws IOException, OutOfMemoryError {

        assert (ontology >= 0 && ontology < 3);
    }
}
```

```

    final int N = numGOtermsPerOntology[ontology];

    Matrix result = new Matrix(N, N);

    for (int i = 0; i < N; i++) {
        for (int j = i; j < N; j++) {
            float res = (float) -Math.log(
                lowestCommonAncestor(
                    matrixAxis[ontology][i].getAncestors(),
                    matrixAxis[ontology][j].getAncestors(),
                    ontology)
            );
            result.set(i, j, res);
            result.set(j, i, res);
        }
    }

    return result;
}

@Override
public Matrix calculateGeneWiseSemanticSimilarity(int ontology)
    throws IOException, OutOfMemoryError {

    return super.geneWiseSimilarityByMaximum(ontology);
}
}

```

This code excerpt presents the *complete* implementation of Resnik's semantic similarity measure. We will explore it step by step, to illustrate the simplicity of GOssTo's extension procedures. There are no specific input assumptions other than the required structures, and the functions you implement have to return the results in matrix form, as shown in the example. You can implement the measures as you see fit, as long as the end product conforms to this standard.

### 3.4.1 Step 1. Class definition and constructor

Every new measure has to extend the HSM class. That is, each measure will implement their own specific methods and functions to produce the results, while using existing utilities to do so.

```

public class Resnik extends HSM {

    public Resnik(GOTerm[] allTerms, String[] genes, String[][] goIds,

```

```

    GOTerm[] [] axis, String[] targets, Assignment annotations,
    String[] relations, TinyLogger logw) {

    super(allTerms, genes, goIds, axis, targets, annotations,
        relations, logw);
}

```

In this particular case, we had no need to initialise any of the parent's class properties, but if the measure you design requires you to, you are free to modify them.

### 3.4.2 Step 2. Define the required methods.

Since the parent class does not actually implement any measure but defines nevertheless placeholders for the methods, you need to override the following methods:

```

public Matrix calculateTermWiseSemanticSimilarity(int ontology)
public Matrix calculateGeneWiseSemanticSimilarity(int ontology)

```

These methods will implement the actual measures.

In this particular case (that is for Resnik's similarity) we need to determine the Lowest common ancestor of every two nodes, and compute its information content.

Given that several measures require the Lowest Common Ancestor to be computed, we provide a method implemented in the parent class.

```

@Override
public Matrix calculateTermWiseSemanticSimilarity(int ontology)
    throws IOException, OutOfMemoryError {

    assert (ontology >= 0 && ontology < 3);
    final int N = numGOTermsPerOntology[ontology];

    Matrix result = new Matrix(N, N);

    for (int i = 0; i < N; i++) {
        for (int j = i; j < N; j++) {
            float res = (float) -Math.log(
                lowestCommonAncestor(
                    matrixAxis[ontology][i].getAncestors(),
                    matrixAxis[ontology][j].getAncestors(),
                    ontology)
            );
            result.set(i, j, res);
            result.set(j, i, res);
        }
    }
}

```

```

    return result;
}

```

The last method that needs to be implemented is the one that computes the genewise similarity. For Resnik it is simply the maximum of all similarities between any two nodes in the Gene Ontology. Again, since several measures follow a similar strategy, we have implemented this method in the parent class.

```

@Override
public Matrix calculateGeneWiseSemanticSimilarity(int ontology)
    throws IOException, OutOfMemoryError {

    return super.geneWiseSimilarityByMaximum(ontology);
}

```

Once the new class is compiled and properly located, it will become available in the HSM menu in the Interactive interface of GOssTo. This is done through the Java Reflection engine, and the path for the compiled classes is fixed to the same place where the existing measures are located.

### 3.5 How to add a new extension measure

We have designed GOssTo as an open system, in which the different components can be disassembled. This allows the user to modify components in order to take advantage of the provided utilities.

As an example, we will show how to replace the Improved Semantic Similarity for another add-on measure that can extend Host Similarity Measures.

The class

```
public class ISM_validImplementation
```

implements the add-on measure.

This class does not extend any other class in the project, with which we obtained a decoupled module that can easily be replaced.

The class constructor has to be appropriately initialised as follows.

```
public ISM_validImplementation(GOTerm[] ISM_currentGoTerms,
                               Matrix HSM,
                               String[] ISM_currentRelations,
                               Assignment ISM_Annotations,
                               boolean termwise,
                               boolean wJaccard,
                               TinyLogger logger)

```

This example is based on ISM, so your measure might not require all these parameters. The input considerations for the different parameters is as follows.

- `GOTerm[] ISM_currentGoTerms`
- `Matrix HSM`: Contains the Host Similarity values, e.g. in case Resnik was selected, this matrix will contain all the Resnik similarity values.
- `String[] ISM_currentRelations`: A string array containing the relations to consider in the ontology.
- `Assignment ISM.Annotations`: An object that contains a mapping between `GOterm` and the annotated genes. The class `Assignment` defines the object.
- `boolean termwise`: Boolean value that determines whether we compute genewise or termwise similarities.
- `boolean wJaccard`: Boolean value that selects a specific weighting we require for ISM.
- `TinyLogger logger`: A logger object. This is clearly defined in the `TinyLogger` class.

While we cannot determine whether you will require more values for the function, the only requirement is that the calling function should be extended to include all the parameters.

All output values are expected to be stored in a `Matrix` object.

In our case, we have implemented the function

```
private Matrix getISM()
```

which returns the end result of the add-on measure. These methods could be replaced or fully renamed to incorporate the new measure.

## 4 Walkthrough

### 4.1 Step by Step Guide to Using the Application

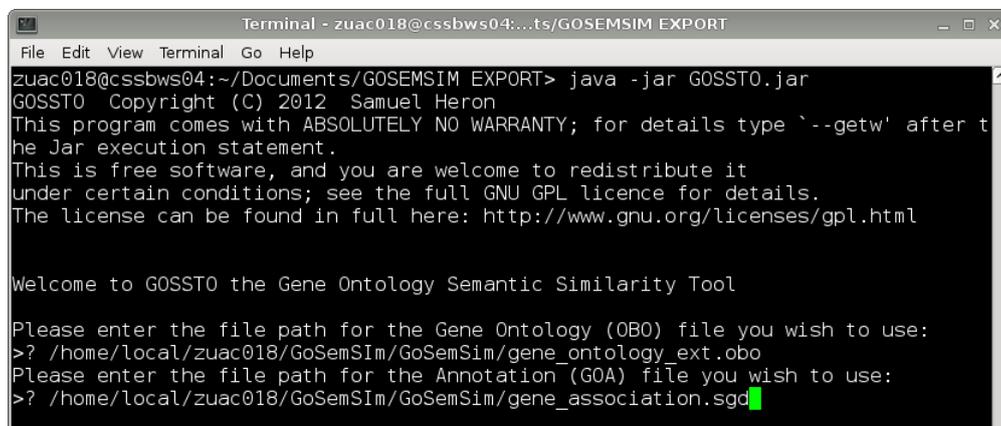
This section aims to give a full walkthrough for an average use of the system. This will be shown using the command line application.

1. The first step is to execute the program itself, this is done by navigating on the console to the location of the GOSSTo .Jar file and executing it by calling 'java -jar GOSSTo.jar' as shown below:



```
Terminal - zuac018@cssbws04:...ts/GOSEMSIM EXPORT
File Edit View Terminal Go Help
zuac018@cssbws04:~/Documents/GOSEMSIM_EXPORT> java -jar GOSSTO.jar
```

2. Once entered the application will start and prompt you for the location a Gene Ontology Project .OBO file and also that of the Gene Ontology Project GOA file. You will need to enter the full file path for both as shown below:



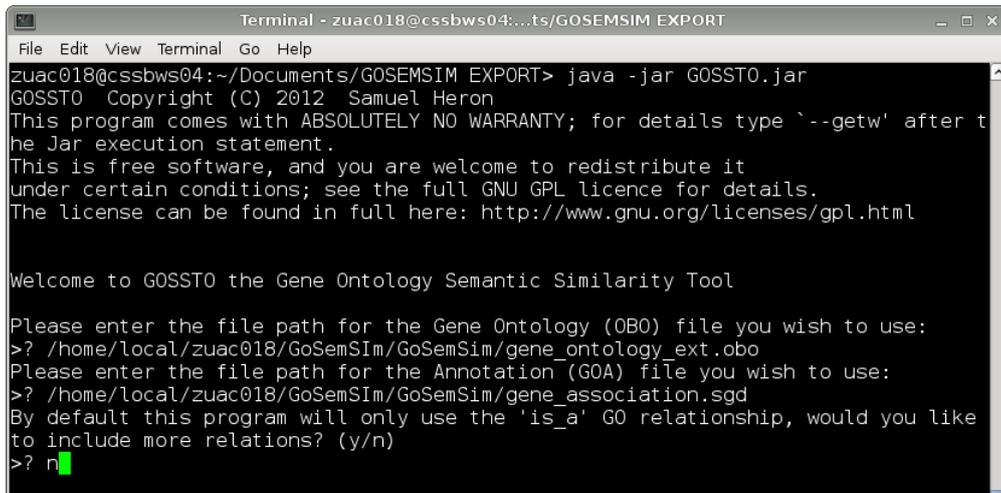
```
Terminal - zuac018@cssbws04:...ts/GOSEMSIM EXPORT
File Edit View Terminal Go Help
zuac018@cssbws04:~/Documents/GOSEMSIM_EXPORT> java -jar GOSSTO.jar
GOSSTO Copyright (C) 2012 Samuel Heron
This program comes with ABSOLUTELY NO WARRANTY; for details type '--getw' after t
he Jar execution statement.
This is free software, and you are welcome to redistribute it
under certain conditions; see the full GNU GPL licence for details.
The license can be found in full here: http://www.gnu.org/licenses/gpl.html

Welcome to GOSSTO the Gene Ontology Semantic Similarity Tool

Please enter the file path for the Gene Ontology (OBO) file you wish to use:
>? /home/local/zuac018/GoSemSim/GoSemSim/gene_ontology_ext.obo
Please enter the file path for the Annotation (GOA) file you wish to use:
>? /home/local/zuac018/GoSemSim/GoSemSim/gene_association.sgd
```

3. Now that we have entered the source files we are working with, the application will begin to prompt us for the parameters of the calculation. Firstly we are given the option to choose whether we wish to enter the Gene Ontology relationships we wish to use or whether to use is\_a as default. For the purposes

of this walkthrough we will choose the default option:

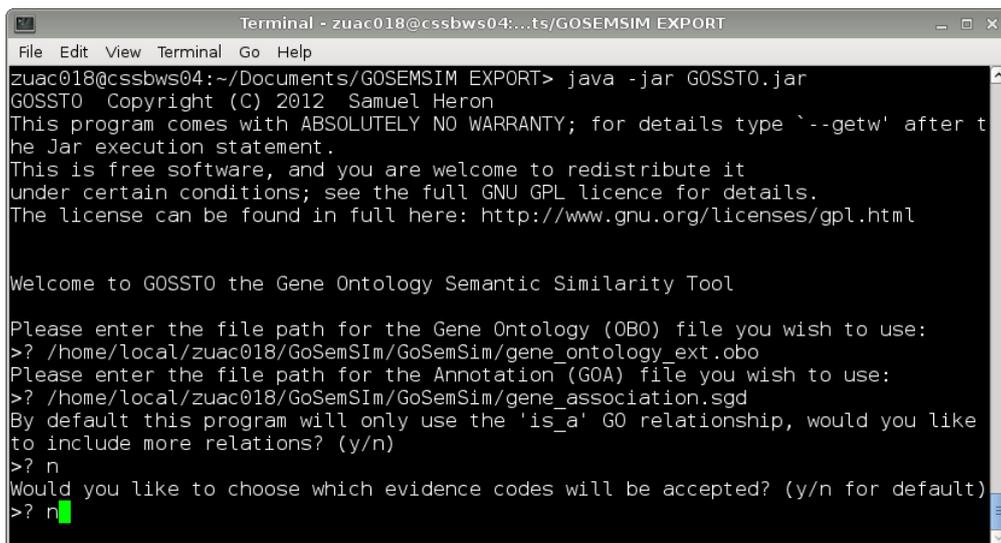


```
Terminal - zuac018@cssbws04:~/Documents/GOSEMSIM EXPORT
File Edit View Terminal Go Help
zuac018@cssbws04:~/Documents/GOSEMSIM EXPORT> java -jar GOSST0.jar
GOSST0 Copyright (C) 2012 Samuel Heron
This program comes with ABSOLUTELY NO WARRANTY; for details type '--getw' after the
Jar execution statement.
This is free software, and you are welcome to redistribute it
under certain conditions; see the full GNU GPL licence for details.
The license can be found in full here: http://www.gnu.org/licenses/gpl.html

Welcome to GOSST0 the Gene Ontology Semantic Similarity Tool

Please enter the file path for the Gene Ontology (OBO) file you wish to use:
>? /home/local/zuac018/GoSemSim/GoSemSim/gene_ontology_ext.obo
Please enter the file path for the Annotation (GOA) file you wish to use:
>? /home/local/zuac018/GoSemSim/GoSemSim/gene_association.sgd
By default this program will only use the 'is_a' GO relationship, would you like
to include more relations? (y/n)
>? n
```

4. We are now presented with the option of choosing the evidence codes that will determine which aspects of the annotation file will be read into the calculation process. For this execution we shall use the default selection that is, for note; EXP, IDA, IPI, IMP, IGI, IEP, TAS and IC. This choice is shown below:



```
Terminal - zuac018@cssbws04:~/Documents/GOSEMSIM EXPORT
File Edit View Terminal Go Help
zuac018@cssbws04:~/Documents/GOSEMSIM EXPORT> java -jar GOSST0.jar
GOSST0 Copyright (C) 2012 Samuel Heron
This program comes with ABSOLUTELY NO WARRANTY; for details type '--getw' after the
Jar execution statement.
This is free software, and you are welcome to redistribute it
under certain conditions; see the full GNU GPL licence for details.
The license can be found in full here: http://www.gnu.org/licenses/gpl.html

Welcome to GOSST0 the Gene Ontology Semantic Similarity Tool

Please enter the file path for the Gene Ontology (OBO) file you wish to use:
>? /home/local/zuac018/GoSemSim/GoSemSim/gene_ontology_ext.obo
Please enter the file path for the Annotation (GOA) file you wish to use:
>? /home/local/zuac018/GoSemSim/GoSemSim/gene_association.sgd
By default this program will only use the 'is_a' GO relationship, would you like
to include more relations? (y/n)
>? n
Would you like to choose which evidence codes will be accepted? (y/n for default)
>? n
```

5. Before we choose our HSM, we are now asked whether we wish to calculate semantic similarity based upon GO term by GO term calculations or gene by gene calculations; termwise or genewise. For this walkthrough we shall choose termwise:

```
Terminal - zuac018@cssbws04:...ts/GOSEMSIM EXPORT
File Edit View Terminal Go Help
GOSSTO Copyright (C) 2012 Samuel Heron
This program comes with ABSOLUTELY NO WARRANTY; for details type '--getw' after t
he Jar execution statement.
This is free software, and you are welcome to redistribute it
under certain conditions; see the full GNU GPL licence for details.
The license can be found in full here: http://www.gnu.org/licenses/gpl.html

Welcome to GOSSTO the Gene Ontology Semantic Similarity Tool

Please enter the file path for the Gene Ontology (OBO) file you wish to use:
>? /home/local/zuac018/GoSemSim/GoSemSim/gene_ontology_ext.obo
Please enter the file path for the Annotation (GOA) file you wish to use:
>? /home/local/zuac018/GoSemSim/GoSemSim/gene_association.sgd
By default this program will only use the 'is_a' GO relationship, would you like
to include more relations? (y/n)
>? n
Would you like to choose which evidence codes will be accepted? (y/n for default)
>? n
Would you like to get genewise (g) or termwise (t) results?: (g/t)
>? t
```

6. Now that the base parameters have been decided upon, we are given the choice of similarity measure. First we are to decide if we want to select one from a list or enter the name of one we wish to use. Note that if you wish to use your own HSM, you must enter it by name. For this walkthrough we will select one from the list:

```
Terminal - zuac018@cssbws04:...ts/GOSEMSIM EXPORT
File Edit View Terminal Go Help
By default this program will only use the 'is_a' GO relationship, would you like
to include more relations? (y/n)
>? n
Would you like to choose which evidence codes will be accepted? (y/n for default)
>? n
Would you like to get genewise (g) or termwise (t) results?: (g/t)
>? t
Would you like to pick an HSM from a list? (y) or enter the name of the desired H
SM? (n)
>? y
Please select the Host Similarity Measure (HSM) that you would like to use from t
he following list:

Termwise & Genewise Measures:
1. Resnik
2. Lin
3. Jiang
4. simGraSM

Please make your choice by entering the number of the desired measure.
>? 3
```

7. The next choice to make is the ontology we want to work with:

```
Terminal - zuac018@cssbws04:...ts/GOSEMSIM EXPORT
File Edit View Terminal Go Help
SM? (n)
>? y
Please select the Host Similarity Measure (HSM) that you would like to use from the following list:

Termwise & Genewise Measures:
1. Resnik
2. Lin
3. Jiang
4. simGraSM

Please make your choice by entering the number of the desired measure.
>? 3
Please select your choice of ontology from the list below by entering its corresponding number:

1. Biological Process
2. Molecular Function
3. Cellular Component
4. All Three
>? 4
```

8. The application will now want to know if we wish to work with particular terms or whether we wish to calculate the similarity between all terms available. For this walkthrough let us calculate them all:

```
Terminal - zuac018@cssbws04:...ts/GOSEMSIM EXPORT
File Edit View Terminal Go Help
Please select the Host Similarity Measure (HSM) that you would like to use from the following list:

Termwise & Genewise Measures:
1. Resnik
2. Lin
3. Jiang
4. simGraSM

Please make your choice by entering the number of the desired measure.
>? 3
Please select your choice of ontology from the list below by entering its corresponding number:

1. Biological Process
2. Molecular Function
3. Cellular Component
4. All Three
>? 4
Would you like to work with a specific set of GO terms? (y/n)
>? n
```

9. The following prompt asks us if we wish to calculate the HSM similarity matrices or if we also wish to calculate the ISM similarity matrices. It is important to note however that the HSM similarity matrices that are produced when calculating the ISM will not be the same as the regular HSM matrices, due to a few calculation differences that aid the calculation of the ISM results. We wish, naturally, to calculate the ISM:

```
Terminal - zuac018@cssbws04:...ts/GOSEMSIM EXPORT
File Edit View Terminal Go Help

Termwise & Genewise Measures:
1. Resnik
2. Lin
3. Jiang
4. simGraSM

Please make your choice by entering the number of the desired measure.
>? 3
Please select your choice of ontology from the list below by entering its corresponding number:

1. Biological Process
2. Molecular Function
3. Cellular Component
4. All Three
>? 4
Would you like to work with a specific set of GO terms? (y/n)
>? n
Would you like to calculate solely HSM results? (y/n)
>? n
```

10. As we approach the end of the user prompt section, we are lastly asked to enter the names we wish to use for the HSM and ISM matrix files. Note that you may wish to precede the name by the location you wish the file to be created in. Please note also that the files will be output as .txt by default so do not enter a file extension in your name. In this example however, we shall create the files in our current location and therefore only specify the name of the files:

```
Terminal - zuac018@cssbws04:...ts/GOSEMSIM EXPORT
File Edit View Terminal Go Help

3. Jiang
4. simGraSM

Please make your choice by entering the number of the desired measure.
>? 3
Please select your choice of ontology from the list below by entering its corresponding number:

1. Biological Process
2. Molecular Function
3. Cellular Component
4. All Three
>? 4
Would you like to work with a specific set of GO terms? (y/n)
>? n
Would you like to calculate solely HSM results? (y/n)
>? n
Please enter the desired file name for the HSM results:
>? Jiang_HSM
Please enter the desired file name for the ISM results:
>? Jiang_ISM
```

11. Now that the parameters have been entered the calculation will begin and the results will be output to either the location specified, or if not; that of the .Jar file.

No more interaction will be required on the part of the user.

If you have any questions or are uncertain of the running of the application, please refer to the FAQ in the next section.

## 4.2 User FAQ

This section aims to satisfy any simple questions or concerns you may have with the GOssTo package. Below is a listing of questions and answers pertaining to the application:

**Q** : The calculation is taking a very long time, should I be concerned?

**A** : No. Calculations involving many terms or genes naturally involve calculating very large matrices of values. This takes quite some time.

**Q** : Help! I've got an error, what should I do?

**A** : Please take note of the error message and refer to it in the Error Messages section at the end of the manual.

**Q** : I want to use an OBO/GOA file from a different provider to the Gene Ontology Project, is this okay?

**A** : If it is formatted to the respective GO file format then theoretically there should not be a problem, give it a try. However GOssTo was built to use files from the GO Project and so we cannot ensure the proper functioning of files from other sources.

**Q** : My results file contains infinite values, why is this so?

**A** : Due to the calculation formulae of certain semantic similarity methods, the similarity between two very dissimilar GO terms or genes will sometimes give an infinite value. In regard to similarity, treat it as a zero.

**Q** : Why are the HSM matrices produced in a ISM calculation not correct as results for that HSM?

**A** : When undertaking an ISM calculation, the annotation values are altered slightly so as to accommodate for and speed up operations when calculating the ISM matrices later on. These changes naturally alter the the values of the HSM matrix.

**Q** : Why is my HSM method not working with the application?

**A** : This could be for a number of reasons. Firstly check that it implements the HSM package, secondly check that the parameters of the HSM constructor are in order, thirdly check if the java file is in the correct location. If none of these are wrong, please refer any error messages you may have to the Error Messages section.

**Q** : I don't like the way you've done ---- / I think you're code here would be better if you did ---- .

**A** : If you have a suggestion or a criticism please feel free to get in touch. Please also have a read of the GNU GPL and feel very much free to alter the source code in any way you see fit.

**Q** : I like GOssTo and would like to use it in my program, is there an easy way I could do this?

**A** : Please refer to the section 'Using GOssTo as a Library' in the Technical Information section. Please also have a read through the GNU GPL if you are thinking of releasing your program.

**Q** : Where can I find more info about the developers of this package?

**A** : <http://www.paccanarolab.org/>

If your question does not appear above, feel free to ask the developers at: .

### **4.3 Using GOssTo as a Library**

This package can be altered for integration as a library for an external software package. This can be done very simply by the removal of the ISM.java file that contains the software's main method. Once removed, the methods and classes are free to be interacted with at will, however the following paragraph will detail the advised order of function and classes.

There are three interface classes packaged within the software that are used to access all key functionality of the system. These classes are 'GOTree\_Interfaces', HSM\_Interfaces and ISM\_Interfaces. These three contain all the desirable functionality with respect to their area of specialty; GOTree deals with all aspects of the source data, HSM deals with retrieving and executing the required similarity measure and ISM focuses on accessing the appropriate ISM class and calculation. It is naturally advised that these classes are accessed in the order that they have been listed here. Note that the output method, should it be required, resides within the 'ioValidation' class.

## 5 Error Messages

This section details the error messages that you may encounter when using GOssTo and any subsequent action that should be taken.

### 5.1 Errors Relating to Console Parameters

ERROR: OBO Path Not Specified: The OBO file path has not been specified when using the console parameters. Ensure you are using the '-obopath' parameter.

ERROR: GOA Path Not Specified: The GOA file path has not been specified when using the console parameters. Ensure you are using the '-goapath' parameter.

ERROR: GO Relations Not Specified: The GO relations have not been specified when using the console parameters. Ensure you are using the '-relations' parameter.

ERROR: Evidence Codes Not Specified: The evidence codes have not been specified when using the console parameters. Ensure you are using the '-evidencecodes' parameter.

ERROR: HSM Not Specified]: The HSM measure has not been specified when using the console parameters. Ensure you are using the '-hsm' parameter.

ERROR: Ontology Choice Not Specified: The ontology to be used has not been specified when using the console parameters. Ensure you are using the '-ontology' parameter.

ERROR: It Has Not Been Specified Whether an HSM or ISM is Being Calculated]: Whether we are calculating an HSM or an ISM has not been specified when using the console parameters. Ensure you are using the '-calculationtype' parameter.

ERROR: It Has Not Been Specified Whether Gene or GO Terms are to be Used: Whether we are doing a termwise or genewise calculation has not been specified when using the console parameters. Ensure you are using the '-calculationdata' parameter.

ERROR: Terms (Gene IDs or GO Terms) Not Specified or Not Set To "all": The terms (genes or GO terms) to be used for the calculation have not been specified when using the console parameters. Ensure you are using the '-terms' parameter, if you wish not to use any specific terms, add 'all' without inverted commas after -terms.

ERROR: HSM Output Path Not Specified: The HSM output name/location has not been specified when using the console parameters. Ensure you are using the '-hsmoutput' parameter.

ERROR: ISM Output Path Not Specified: The ISM output name/location has not been specified when using the console parameters. Ensure you are using the '-ismoutput' parameter.

ERROR: HSM/ISM specification has been entered incorrectly: The calculation type was not either 'hsm' or 'ism'. Ensure that the '-calculationtype' parameter is followed only by 'ism' or 'hsm' in either upper or lower case.

ERROR: Genewise/Termwise specification has been entered incorrectly: The calculation data option was not either 'termwise' or 'genewise'. Ensure that the '-calculationdata' parameter is followed only by 'termwise' or 'genewise' in either upper or lower case.

ERROR: Parse failed: A problem has occurred relating either to the execution of the program or with the parameters you have used, please check that you are executing the program correctly and that all parameters are present and correct.

## 5.2 Errors Relating to Input Validation

ERROR: Invalid Entry: The entry you have just made is either: not an available option or the wrong type of data.

ERROR: Invalid Choice: The choice you have made was not one that is available.

ERROR: Inputted relation does not exist; try again: The GO relation you have entered is not a valid GO relation, please enter a valid GO relation.

ERROR: Inputted relation has already been input: The GO relation you have entered is one that has already been entered. Do not enter the same relation more than once.

ERROR: Choice of Ontology Invalid: The choice of ontology you have entered is not a choice that is available, please check your entry and try again.

ERROR: Relation: \*specific relation\*, is not acceptable or does not exist: The GO relation you have entered is not a valid GO relation, please check your entry and try again.

ERROR: Evidence code: \*specific evidence code\*, is not acceptable or does not exist: The evidence code you have entered is not a valid evidence code, please check your entry and try again.

ERROR: No OBO file at specified filepath: The file path you have specified for the GO OBO file does not produce a valid GO file, please check your OBO file's file path and try again.

ERROR: No GOA file at specified filepath: The file path you have specified for the GO GOA file does not produce a valid GO file, please check your GOA file's file path and try again.

ERROR: Invalid output location: The location you have specified to output a results file to is not a valid location, please check the location exists and that this software has permissions to write there and try again.

ERROR: One or more goterm IDs entered does not exist: One or more of the GO terms you have entered is not recognised. Please check your entry and your OBO file and try again.

### 5.3 Errors Relating to Calculation and Parsing

ERROR: GOssTo has run out of memory: the memory allocated is insufficient for this calculation. To increase the memory allocated use the -Xmx parameter when launching the execution. For example use: `java -Xmx3000M -jar gossto.jar` to increase the maximum available memory to 3 Gb. Alternatively, please use the web service at [www.paccanarolab.org/gosstoweb](http://www.paccanarolab.org/gosstoweb)

ERROR: Too many indirection levels found for GO term id: *\*specific GO ID\**: This means that this term is obsolete and points to another term which is also obsolete and either no alternative terms have been processed for it, or, there is more than a level of indirection. Update your OBO file, comment this entry out by placing exclamation marks at the start of each row containing its data or remove this entry.

ERROR: consider tag present for: *\*specific GO ID\**: This error raises attention to the fact that 'consider' tags have been used in the definition of this term. This will not stop calculations taking place but may affect them. To find out more about consider tags refer to the optional tags section on this page: <http://geneontology.org/GO.format.obo-1.2.shtml>

ERROR: The kind of relations that are going to be parsed should be set first via `GOTerm .setRelations`: A `GOTerm` instance has been created without a relation parameter, check the integrity of your OBO file.

ERROR: The relation *\*specific relation\** is not being parsed in this instance of the Gene Ontology: The relation in question is unrecognised or not accepted, please check your OBO file.

ERROR: GO term doesn't exist]: A GO term is being used that has not been recognised. Please check your OBO file is correct and up to date.

ERROR: Gene IDs entered could not be found in the GOA file supplied: One or more gene IDs is being used that is not recognised. Please check your GOA file is correct and up to date.

ERROR: Problem with source data, please check OBO and GOA files are correct: A problem has arisen when attempting to manipulate the source data. Check that your GOA and OBO files are correct and try again.

ERROR: class \*class name\* is not instantiable: There was an error in accessing the HSM class in question. Please check that it is in the correct location, that it implements the 'HSM' interface and that it is defined and formatted correctly.

ERROR: class \*class name\* has an inaccessible constructor: The constructor for the HSM class in question is inaccessible by GOssTo, please ensure that its permissions are set to public.

ERROR: class \*class name\* not found: The HSM class in question could not be found. Please ensure that you typed the name correctly and that the file for that class is in the correct location.

ERROR: class \*class name\* could not be invoked: There was a problem invoking the HSM class in question. Please ensure that the code for this class is correct and parsable.

ERROR: class \*class name\* has an illegal argument: This error was triggered by an illegal argument the HSM class in question. Please ensure that the code for this class is correct and parsable.

ERROR: class \*class name\* has triggered a security exception.: The HSM class in question has caused a security violation. Please ensure that the code for this class is correct and parsable.

ERROR: Problem with the Gene Ontology file.]:The Gene Ontology file is possibly corrupted. Term GO:000xxxx points to term GO:000yyyy using the 'is\_a' relationship. However, the term GO:000yyyy does not appear in your Gene Ontology file (probably it is obsolete). If the error persists with a newer Gene Ontology file, substitute the reference to GO:000yyyy in the term GO:000xxxx y one of the alternatives provided in its 'obsolete term' entry.

ERROR:An unexpected error has occurred. Error Signature: \*error message or exception listing\*: An error has occurred that has not been anticipated, the cause is uncertain. Please refer to the 'Other Errors' Section.

## 5.4 Other Errors

In the unlikely event that an error is triggered that is not listed here, please contact the developers with a screenshot of the error message and instructions on what you have

done to produce this error.

If it is possible to recreate the error, then it would also be very useful if you could recreate it using the '-logfile' console parameter and supply the resultant file 'LOG.txt' with your message. All help in solving any bugs will be much appreciated.

# Licensing

We have tried our best to provide the user with free software. We have, therefore, selected a GPLv3 licence. In the future, more permissive licensing might be available.

## 5.4.1 Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The

systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## 5.4.2 Terms and conditions

### 0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

## 1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which

you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no

permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation’s users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

## 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of

what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an

implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may

add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

#### 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream

recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others’ Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public

License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR

A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
```

```
Copyright (C) <textyear> <name of author>
```

```
This program is free software: you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation, either version 3 of the License, or  
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License  
along with this program. If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type 'show c' for details.

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.