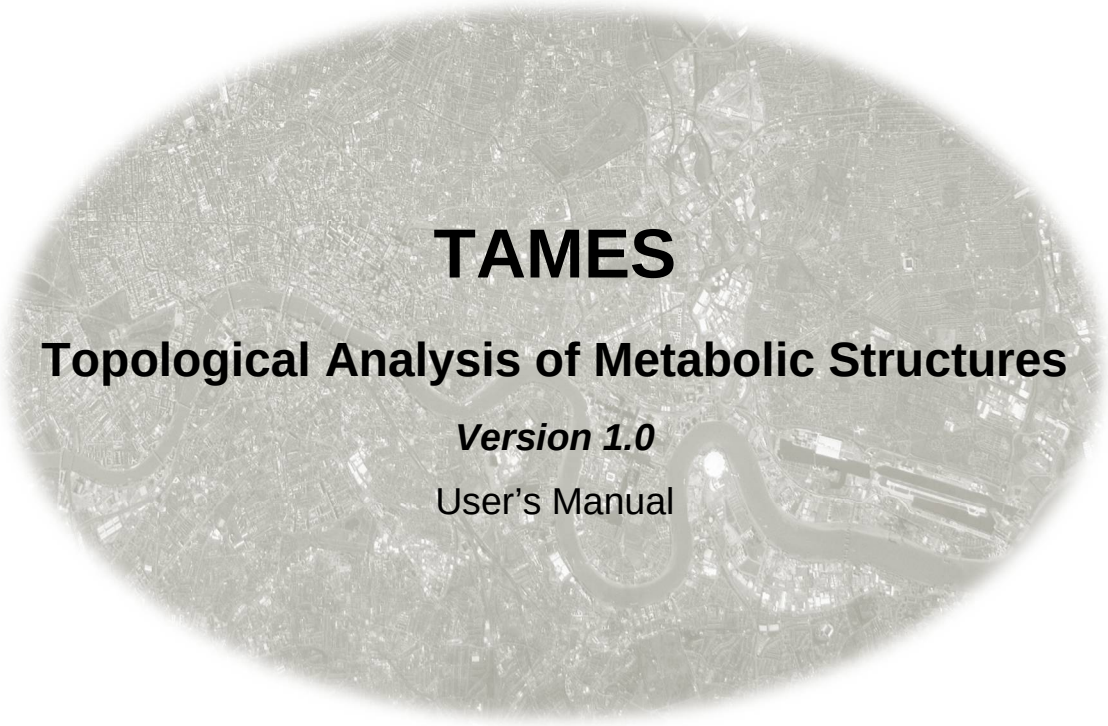
University of Coimbra

Department of Chemical Engineering



TAMES

Topological Analysis of Metabolic Structures

Version 1.0

User's Manual

Developed by:

Paulo A. N. Dias

Marco Seabra dos Reis

Armindo Salvador

Pedro Martins

Coimbra, 23 January 2017

Copyright (C) 2017
Paulo A. N. Dias
Marco Seabra dos Reis
Armindo Salvador
Pedro Martins

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in Annex I, entitled "GNU Free Documentation License".

Contents

1 – INTRODUCTION 1

1.1 – BACKGROUND 1
1.2 – APPROACH 2
1.3 – ORGANIZATION OF THE DOCUMENT 3

2 – DEFINITIONS 4

3 – TAMES: BRIEF DESCRIPTION 6

3.1 – OVERVIEW 6
3.2 – STRUCTURES ENUMERATION [*STRUCTURES ENUMERATION.EXE*] 10
3.3 – INSTANCES COUNTING [*INSTANCES COUNTING.EXE*] 21
3.4 – ORGANIZING INSTANCES [*ORGANIZING INSTANCES.EXE*] 28
3.5 – STAND-ALONE INSTANCES [*STAL-INSTCOUNT.EXE*] 31
3.6 – PREFERENTIAL PATHS [*PREFERENTIAL PATHS.EXE*] 36
3.7 – CHILDREN INFORMATION [*CHILDREN INFORMATION.EXE*] 38
3.8 – LIMITS ON THE USE OF TAMES 44

4 – RECOMMENDATIONS 45

5 – APPLICABLE LICENSES 46

5.1 – TAMES 46
5.2 – MANUAL 46

6 – ACKNOWLEDGMENTS 47

7 – REFERENCES 48

ANNEX I – GNU FREE DOCUMENTATION LICENSE 49

Nomenclature

$Ch(n)_x$	Set of all the children of structure n with a parent/child relationship x
$CNum[y]$	Column number with the information regarding *y*
$Equ(i)_{X/Y}$	Topological equivalence of the two nodes in positions *X* and *Y* of structure *i*
$f_E(i)$	Embedded frequency of structure *i*, defined as the total number of instances of the structure, whether or not embedded in any of its offspring
$f_{Met\,x}(i)$	Frequency of metabolite *x* at structure *i* (all positions considered)
$f_{Met\,x}(i)_j$	Frequency of metabolite *x* at the position *j* of structure *i*
$f_N(i)$	Naked frequency of structure *i*, which corresponds to the number of instances matching the structure but not any of its offspring
$f_{React\,x}(i)$	Frequency of reaction *x* in structure *i* (all positions considered)
$f_{React\,x}(i)_j$	Frequency of reaction *x* in the position *j* of structure *i*
$H_{max}(n)_x$	Maximum possible (or feasible) entropy for structure n for a parent/child relationship x
$H(n)_x$	Shannon entropy of structure n for a parent/child relationship x
$N_{Ch}(n)_x$	Number of children of structure n with a parent/child relationship x
$NGen_{max}$	Maximum number of generations (higher generation indicates higher complexity) considered for structures of the topological scope
$NMet(i)$	Number of metabolite nodes of structure *i*
$NMet_{max}$	Maximum number of metabolite nodes that can be found in structures of the topological scope
$NReact(i)$	Number of reaction nodes of structure *i*
$NReact_{max}$	Maximum number of reaction nodes that can be found in structures of the topological scope
$NumCol[file]$	Number of columns in the file
$Rf_N(j)_{n,x}$	Ratio of frequencies of a child j of structure n with a parent/child relationship x
$RNum[y]$	Row number with the information regarding *y*
TAMES	"Topological Analysis of Metabolic Structures" – software package developed to study local structures in biological networks
$\Delta H(n)_x$	Children information of structure n for a parent/child relationship x

1 – Introduction

The software package “TAMES – Topological Analysis of Metabolic Structures” is to study local structures in biological networks. This document is the instructions manual of TAMES. In this section we present a short bibliographic review with the state-of-the-art in biological networks analysis only to refer the inherent problems facing the current methods. The approach followed in this work to avoid and circumvent these problems is then described. This section ends with the structure adopted for this document.

1.1 – Background

The detection of over- or under-represented local structures (motifs and anti-motifs, respectively) in biological networks has attracted much interest as a way of detecting potential selective constraints [1-3]. The implicit rationale is that over-representation with respect to expectation from a prescribed null model is likely to be a consequence of natural selection for maintenance of the motif, driven by functional advantages provided by its dynamic properties. However, physically realizable and biologically plausible null models of metabolic networks are not easy to be artificially conceived. Null models generated by the randomization procedures considered in previous publications [4] are not physically realizable, as they violate atom conservation. An example of a randomized network obtained by preserving the degree of the nodes, i.e., the number of connections or arcs a node has with other nodes, is shown in Figure 1. For directed networks, wherein the arcs point in one direction from one node to another node, two different type of degrees are considered: the in-degree, obtained by the number of incoming arcs, and the out-degree, obtained by the number of outgoing arcs [5]. The meaningfulness of statistical over-representation with respect to such unrealistic null models is therefore questionable [6]. As a more reasonable alternative, Noor et al. [7] considered randomly generated networks such that all the allowed reactions could in principle be catalysed by known enzyme activities considered at the third level of the Enzyme Commission (E.C.) classification. However, the extent to which networks generated in this way are legitimate “no-selection” null models is debatable. Enzymes are products of natural selection and expensive to maintain. So, the fact that an enzyme exists to catalyse a given reaction is already an indication that such a reaction has selective advantages for at least some organisms. Furthermore, this procedure is liable to knowledge bias.

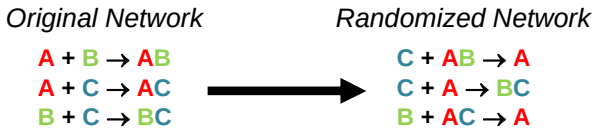


Figure 1 – Example of a randomized network obtained by preserving the degree of the nodes [5]. Each one of the letters (A, B and C) correspond to a given atom. Atom conservation is not attained in the randomized network.

The representation of metabolic connectivity structures in motif analysis can also be problematic. Previous analyses [4] defined structures as simple directed graphs where each metabolite node A is connected to another one B by a directed arc if there is a reaction converting A to B. However, such metabolite connectivity structures are ambiguous and difficult to interpret and to relate with dynamics and function. This is because the same topological structure may correspond to very different reaction structures. Thus, metabolic networks require more complex representations such as directed bipartite graphs with both metabolite and reaction nodes being considered [2]. Figure 2 shows an example of different reaction structures represented as directed bipartite graphs, where one can notice the structural ambiguity of simple directed graphs.

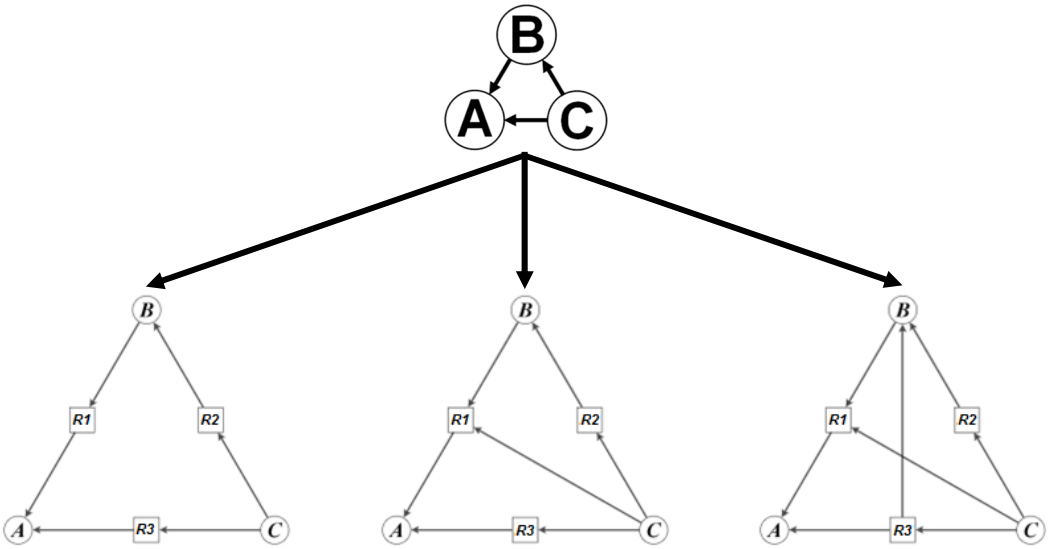


Figure 2 –The simple directed graph representation of a structure known as the feed-forward loop [5] can be obtained from three different reactions structures represented as directed bipartite graphs.

1.2 – Approach

A new approach was developed to systematically highlight and visualize strong statistical preferences in local metabolic connectivity structures without requiring an artificial null model [8]. In order to avoid ambiguity arising by using simple directed graphs, structures are represented as directed bipartite graphs, with reaction and metabolite nodes [2]. The

approach consists in first constructing a topological hierarchy of structures of increasing complexity. Then, comparisons among the frequencies of structures differing by a single structural element are performed using this framework. The structural elements can be: i) a reactant or product to a reaction, ii) a new reaction converting one metabolite into one other, iii) the reverse of a reaction, iv) a new metabolite, v) a new reaction connecting a new metabolite to an existing one to form a pair reactant/product.

1.3 – Organization of the Document

The software package TAMES is described in this document. Section 2 presents basic definitions required to analyse the results given by TAMES. Additional relevant definitions are presented in section 3. It also describes the executable programs of TAMES, including their input and output files. Section 4 presents recommendations on the use of TAMES. The licenses that apply to TAMES and to this document are presented in section 5 and the acknowledgements in section 6. Finally, a list of references is presented in section 7.

2 – Definitions

Relevant concepts involving the results obtained by TAMES are presented below.

Structure

A "topological structure", abbreviated "structure", is a specific pattern of interconnections between the two types of nodes of the metabolic network: metabolites and reactions. Isomorphic representations obtained from each other by relabelling metabolites and/or reactions are considered as the same structure. The simplest structure comprises a reaction node connected to two metabolite nodes, one product and one reactant.

Directly Related Structures

Two structures are said to be directly related if the most complex one (henceforth denoted by "child") can be obtained from the simplest one (denoted by "parent") by addition of just one of the following structural components: (i) a reactant or product to a reaction; (ii) a new reaction converting one metabolite into one other; (iii) the reverse of a reaction; (iv) a new metabolite; (v) a new reaction connecting a new metabolite to an existing one to form a pair reactant/product.

Each structure may have multiple parents and children. We denote by "offspring" of a structure the set of all of its children, grand-children, etc. Similarly, we denote "ancestors" of a structure the set of all its parents, grand-parents, etc.

Instance

An "instance" is any concrete realization of a given structure in a metabolic network, i.e., an actual set of metabolites and reactions connected as defined in the structure. We define the naked frequency of structure *i*, $f_N(i)$, as the number of instances matching the structure but not any of its offspring. On the other hand, the embedded frequency of structure *i*, $f_E(i)$, is defined as the total number of instances of the structure, whether or not embedded in any of its offspring. The value of $f_E(i)$ is obtained by adding $f_N(i)$ and the naked frequencies of all of structure *i*'s offspring.

Network of Directly Related Structures

The concept of directly related structures allows to generate a layered network of topological relationships among structures. The nodes in this network are the non-isomorphic topological structures arranged in generations, from the sparsest (*i.e.*, with the fewest structural components) structures to the densest. The arcs join directly related structures. Importantly, the topology of this network is uniquely determined by the

topological relationships of the structures and independent of the properties of concrete metabolic networks. It therefore provides a convenient and sound referential to represent the local-topology characteristics of the metabolism of different microorganisms and assess their affinities and particularities through topological oriented statistical descriptors.

3 – TAMES: Brief Description

This section starts with an overview of TAMES by listing its executable programs and the respective order of execution. The input and output files of each one of the executable programs are then described, followed by the limits allowed by the software.

3.1 – Overview

All algorithms were developed and implemented in FORTRAN 90. They are able to identify strong statistical preferences in local metabolic connectivity structures without requiring an artificial null model. Directed bipartite graphs were used to represent topological structures, with reaction and metabolite nodes. The developed code was compiled (GFortran compiler) in order to obtain the executable programs of TAMES (Table 1). These programs are divided into two main groups: i) those that operate on generic topological structures, and ii) those that operate on metabolic networks. The executable programs must be run following a pre-defined order.

Table 1 – General information regarding the executable programs of TAMES: order of execution, name, executable file, original code file (FORTRAN 90) and operation.

Execution Order	Executable Program		Code File (FORTRAN 90)	Operation
	Name	***File***		
1	Structures Enumeration	*Structures Enumeration.exe*	*Structures_v1.f90*	Generic topological structures
2	Instances Counting	*Instances Counting.exe*	*InstCount_v1.f90*	Metabolic Network
3	Organizing Instances	*Organizing Instances.exe*	*OrgInst.f90*	Metabolic Network
4	StAI-InstCount	*StAI-InstCount.exe*	*StAIInstCount_v1.f90*	Metabolic Network
5	Preferential Paths	*Preferential Paths.exe*	*PrefPaths.f90*	Metabolic Network
6	Children Information	*Children Information*	*ShannonEntropy.f90*	Metabolic Network

The code file is edited by an integrated development environment (IDE) software application such as the Eclypse Luna. IDE applications usually have user manuals that should be consulted. For the Eclypse Luna, the following simplified steps are followed in order to edit and compile the code: i) install Eclipse Luna; ii) install a compiler (for example, GFortran); iii) open Eclypse Luna – a folder “workspace” is created, where all the folders and files of the several FORTRAN Projects are kept; iv) confirm that the setup environmental variable PATH of Eclipse Luna includes the folder location of the installed compiler; v) create a new FORTRAN Project with the name of the intended executable program; vi) put the code file of the corresponding executable program in the folder of the newly created FORTRAN Project; vii) open the FORTRAN Project in Eclypse Luna; viii) click on the code file; ix) introduce the intended changes to the code and save when

finished; x) compile the FORTRAN Project to obtain the new version of the executable program of TAMES.

The features of the executable programs of TAMES are described in Table 2. The software uses plane text file format in which values in the same row are separated by commas. Before running TAMES, input files must be prepared with the following information: i) topological scope and ii) metabolic network.

The topological scope studied is defined by setting the following maximum values: i) number of metabolites and reactions of the structures, and ii) number of structural elements in the most complex structures considered (corresponds to the generation number in the network of directly related structures). These values are described in the file *Topological_Scope.txt* (see section 3.2). The suggested predefined values allow the study of all the topological structures with 2 to 3 metabolites and with 1 to 3 reactions (9 generations). In this document, a much smaller scope is considered in order to present the results given by the software in an easily readable manner. Accordingly, the scope shown in this manual consists only of the topological structures of the first two generations of the network of directly related structures.

The initial input data describing the metabolic network is given in the following files: *List_Metabolites.txt*, *List_Reactions.txt*, *Products_X_Reactions.txt*, *Reactions_X_Reactants.txt* and *RevReact.txt*. In this document, the metabolic network of *Saccharomyces cerevisiae* [9] was used. For other metabolic networks, two options can be followed: i) use the same names for the required files or ii) change the names of the files in the original code files of the package and compile new executable programs. For further details, see section 3.3.

Table 2 – Description of the executable programs of TAMES.

Execution Order	Executable Program	Description
1	Structures Enumeration	i) Determine all the possible topological structures of a defined scope by using a bottom-up approach; ii) Obtain the network of directly related structures (or family network); iii) Obtain the inclusion matrix (or descendants network), which indicates if a given structure can be found within a more complex structure; iv) Determine the equivalent topological nodes (metabolites and reactions) of each one of the structures;
2	Instances Counting	i) Obtain the instances of all the structures of a defined scope. Some of the instances may be repeated;
3	Organizing Instances	i) Obtain the actual number of instances by determining the repeated ones; ii) Organize the instances according to their structure;
4	StAI-InstCount	i) Obtain the embedded frequency and the instances with overlaps by up to one metabolite; ii) Determine the frequency of specific metabolites and reactions in the instances of a given structure: overall and in each node position (takes into account the existence of equivalent topological nodes);
5	Preferential Paths	i) Obtain the preferential path within the network of directly related structures by starting from a given structure, i.e., obtains the most frequent descendants by generation of a given structure;
6	Children Information	i) Obtain the ratios of the naked frequencies of the Childs, Rf_N; ii) Determine the children information, ΔH, for each one of the considered Parent/Child relationships. The value of ΔH is null if all the children are equally frequent, and high if the distribution of children frequencies is very uneven;

The determination of the actual number of instances for a defined topological scope in a given metabolic network is carried out with resort to two executable programs: “Instances Counting” and “Organizing Instances”. This allows for a significant decrease of the computational time needed, achieved by comparing only instances of the same structure to determine the repeated ones.

The input and output files of the executable programs are listed in Table 3. The files are classified according to their function within the package: i) setting the topological scope, ii) metabolic network data file, iii) intermediate file or iv) results file. The files are described in detail in the next sections. A simplified flowchart of TAMES is presented in Figure 3.

Table 3 – Input and output files of the executable programs of TAMES.

File	Type	Executable Program					
		1	2	3	4	5	6
Topological_Scope.txt	Topological Scope	IN	--	--	--	--	--
Structures_Metabolite_A.txt	Result	OUT	IN	--	--	--	--
Structures_Metabolite_B.txt	Result	OUT	IN	--	--	--	--
Structures_Metabolite_C.txt	Result	OUT	IN	--	--	--	--
Structures_Metabolite_D.txt	Result	OUT	IN	--	--	--	--
Structures_Metabolite_E.txt	Result	OUT	IN	--	--	--	--
Structures_Metabolite_F.txt	Result	OUT	IN	--	--	--	--
Structures_General.txt	Result	OUT	IN	IN	IN	IN	IN
Structures_RevReact.txt	Result	OUT	--	--	--	--	--
Family_Structures.txt	Result	OUT	--	--	--	IN	IN
InclusionMatrix_Structures.txt	Result	OUT	--	--	IN	--	--
MetDegreeRevStru.txt	Result	OUT	IN	--	--	--	--
ReactDegreeRevStru.txt	Result	OUT	IN	--	--	--	--
EquiMetRevStru.txt	Result	OUT	--	--	IN	--	--
EquiReactRevStru.txt	Result	OUT	--	--	IN	--	--
DifNodesStru.txt	Intermediate	OUT	IN	--	--	--	--
in_nGen_nStru.txt	Intermediate	OUT	IN	IN	IN	IN	IN
in_nMet_nReact.txt	Intermediate	OUT	IN	IN	IN	IN	IN
List_Metabolites.txt	Metabolic Network	--	IN	--	IN	--	--
List_Reactions.txt	Metabolic Network	--	IN	--	IN	--	--
Products_X_Reactions.txt	Metabolic Network	--	IN	--	--	--	--
Reactions_X_Reactants.txt	Metabolic Network	--	IN	--	--	--	--
RevReact.txt	Metabolic Network	--	IN	--	--	--	--
MetInst.txt	Intermediate	--	OUT	IN	--	--	--
ReactInst.txt	Intermediate	--	OUT	IN	--	--	--
InstStruct.txt	Intermediate	--	OUT	IN	--	--	--
MetInstOrdRefNum.txt	Intermediate	--	OUT	IN	--	--	--
ReactInstOrdRefNum.txt	Intermediate	--	OUT	IN	--	--	--
SumMetInst.txt	Intermediate	--	OUT	IN	--	--	--
SumReactInst.txt	Intermediate	--	OUT	IN	--	--	--
OutRevStruct.txt	Intermediate	--	OUT	IN	--	--	--
numInstances.txt	Intermediate	--	OUT	IN	--	--	--
OutRevStructReal.txt	Intermediate	--	--	OUT	IN	--	--
numRefInstLowHigh.txt	Result	--	--	OUT	IN	--	--
numInstancesReal.txt	Result	--	--	OUT	IN	--	--
OrgMetInst.txt	Result	--	--	OUT	IN	--	--
OrgReactInst.txt	Result	--	--	OUT	IN	--	--
ResultsRevStruct.txt	Result	--	--	--	OUT	IN	IN
MetRevStruct.txt	Result	--	--	--	OUT	--	--
ReactRevStruct.txt	Result	--	--	--	OUT	--	--
MetABCRevStruct.txt	Result	--	--	--	OUT	--	--
React123RevStruct.txt	Result	--	--	--	OUT	--	--
RankStru.txt	Result	--	--	--	--	OUT	--
PrefPath.txt	Result	--	--	--	--	OUT	--
RfN_AllRel.txt	Result	--	--	--	--	--	OUT
RfN_I.txt	Result	--	--	--	--	--	OUT
RfN_II.txt	Result	--	--	--	--	--	OUT
RfN_III.txt	Result	--	--	--	--	--	OUT
RfN_IV.txt	Result	--	--	--	--	--	OUT
RfN_V.txt	Result	--	--	--	--	--	OUT
ChildrenInf_AllRel.txt	Result	--	--	--	--	--	OUT
ChildrenInf_I.txt	Result	--	--	--	--	--	OUT
ChildrenInf_II.txt	Result	--	--	--	--	--	OUT
ChildrenInf_III.txt	Result	--	--	--	--	--	OUT
ChildrenInf_IV.txt	Result	--	--	--	--	--	OUT
ChildrenInf_V.txt	Result	--	--	--	--	--	OUT

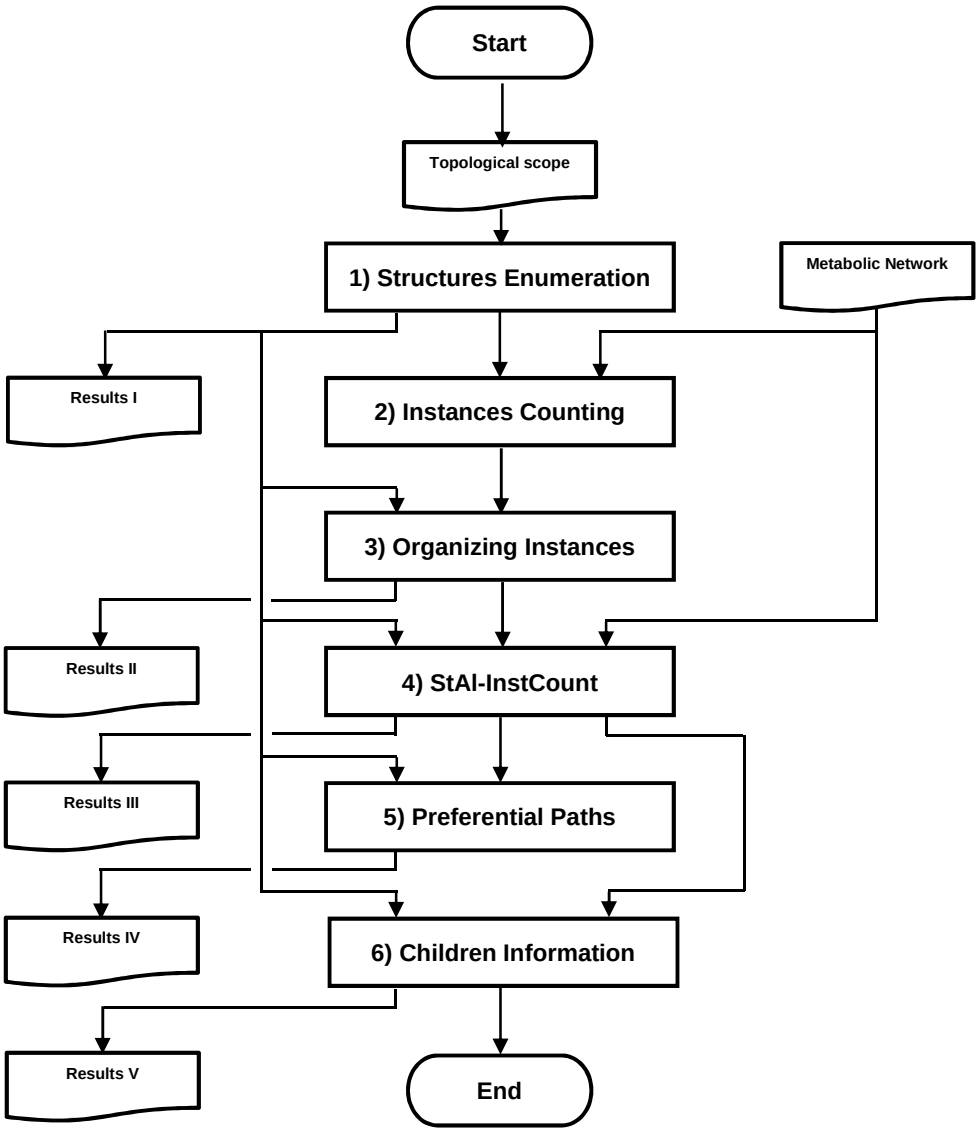


Figure 3 – Simplified flowchart to apply TAMES.

3.2 – Structures Enumeration [*Structures Enumeration.exe*]

The executable program "Structures Enumeration" is obtained by compiling the code file *Structures_v1.f90* with GFortran. This program operates on the generic topologies and has the following goals:

i. Determine all the possible topological structures of a defined scope by using a bottom-up approach;
ii. Obtain the network of directly related structures (or family network);
iii. Obtain the inclusion matrix (or descendants network), which indicates if a given structure can be found within a more complex structure;

iv. Determine the equivalent topological nodes (metabolites and reactions) of each one of the structures;

A general description of the input (Table 4) and output (Table 5) files of this program is presented below.

Table 4 – General description of the input files of the executable program "Structures Enumeration".

File		Created by	Description
Name	**Type**		
Topological_Scope.txt	Topological Scope	*n.a.*	Definition of the topological scope by setting the maximum number of metabolites and of reactions for the structures and the number of generations

Table 5 – General description of the output files of the executable program "Structures Enumeration".

File		Used by	Description
Name	**Type**		
Structures_Metabolite_A.txt *Structures_Metabolite_B.txt* *Structures_Metabolite_C.txt* *Structures_Metabolite_D.txt* *Structures_Metabolite_E.txt* *Structures_Metabolite_F.txt*	Result	*2) Instances Counting*	The files store the pattern of the structures of the defined scope. Each file stores the part of the pattern corresponding to one of the metabolites
Structures_General.txt	Result	*2) Instances Counting* *3) Organizing Instances* *4) StAl-InstCount* *5) Preferential Paths* *6) Children Information*	General information regarding the structures such as: number of generation; number of metabolites, reactions and reversible reactions and number of interconnections
Structures_RevReact.txt	Result	*n.a.*	Information regarding the reversibility of the reactions of a structure
Family_Structures.txt	Result	*5) Preferential Paths* *6) Children Information*	Matrix with the family relationships Parent/Child (network of directly related structures)
InclusionMatrix_Structures.txt	Result	*4) StAl-InstCount*	Inclusion matrix. Gives the information regarding the Offspring of a given structure
MetDegreeRevStru.txt	Result	*2) Instances Counting*	Degree (total, in and out) for each metabolite of a given structure
ReactDegreeRevStru.txt	Result	*2) Instances Counting*	Degree (total, in and out) for each reaction of a given structure
EquiMetRevStru.txt	Result	*4) StAl-InstCount*	Information regarding the topologically equivalent metabolites of a given structure
EquiReactRevStru.txt	Result	*4) StAl-InstCount*	Information regarding the topologically equivalent reactions of a given structure
DifNodesStru.txt	Intermediate	*2) Instances Counting*	Information indicating if a given structure has different nodes of given type (metabolites, irreversible reactions and reversible reactions) by taking into account their degree (total and in)
in_nGen_nStru.txt	Intermediate	*2) Instances Counting* *3) Organizing Instances* *4) StAl-InstCount* *5) Preferential Paths* *6) Children Information*	Information regarding the number of generations considered, the number of structures obtained, and the minimum and maximum reference numbers of the structures belonging to each one of the considered generations
in_nMet_nReact.txt	Intermediate	*2) Instances Counting* *3) Organizing Instances* *4) StAl-InstCount* *5) Preferential Paths* *6) Children Information*	Maximum number of metabolites and of reactions considered for the structures within the assumed number of generations

The topological scope is previously defined by setting the maximum number of nodes (metabolites and reactions) in the structures and the number of considered generations in the file *Topological_Scope.txt*. The suggested predefined numbers are: i) maximum

number of metabolites, $NMet_{max}$ = 3; ii) maximum number of reactions, $NReact_{max}$ = 3; iii) number of generations, $NGen_{max}$ = 9. These settings allow the study of all the topological structures with 2 to 3 metabolites and with 1 to 3 reactions (9 generations). In the example considered in this document, a much smaller scope is considered in order to present the results given by the software in an easily readable manner (Table 6).

Table 6 – Topological scope considered in this manual as defined in file *Topological_Scope.txt*.

Maximum Number	Values
Metabolites, $NMet_{max}$	3
Reactions, $NReact_{max}$	2
Generations, $NGen_{max}$	2

In the example presented in this manual, only the topological structures of the first two generations of the network of directly related structures are considered. The output files obtained for this scope are described next.

The information regarding the interconnections of each structure are given in the files *Structures_Metabolite_A.txt* (Table 7), *Structures_Metabolite_B.txt* (Table 8) and *Structures_Metabolite_C.txt* (Table 9). Each one of these files provides the interconnections between the node of one metabolite and all the reaction nodes of all the structures within the scope. So, the number of files created depends on the maximum number of metabolites considered for the structures of the scope. The columns represent the reaction nodes and the rows represent the reference number of the structures (i.e., their label). For each reaction, the two possible unidirectional reactions were considered: the direct and the inverse reactions. If the metabolite of a given structure is a reactant or a product of a unidirectional reaction, "1" is inserted in the corresponding cell. On the other hand, if no relationship exists between the metabolite and the unidirectional reaction, the cell is filled with "0".

Table 7 – Interconnections between the metabolite A and all the reactions for each structure of the topological scope as given by the file *Structures_Metabolite_A.txt*.

Structure Number	Reaction 1				Reaction 2			
	Direct (+R1)		*Inverse (-R1)*		*Direct (+R1)*		*Inverse (-R1)*	
	Reactant	*Product*	*Reactant*	*Product*	*Reactant*	*Product*	*Reactant*	*Product*
1	0	1	0	0	0	0	0	0
2	0	1	0	0	1	0	0	0
3	0	1	0	0	0	1	0	0
4	0	1	1	0	0	0	0	0
5	0	1	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0
7	0	1	0	0	1	0	0	0
8	0	1	0	0	0	1	0	0
9	1	0	0	0	1	0	0	0

Table 8 – Interconnections between the metabolite B and all the reactions for each structure of the topological scope as given by the file *Structures_Metabolite_B.txt*.

Structure Number	Reaction 1				Reaction 2			
	Direct (+R1)		*Inverse (-R1)*		*Direct (+R1)*		*Inverse (-R1)*	
	Reactant	*Product*	*Reactant*	*Product*	*Reactant*	*Product*	*Reactant*	*Product*
1	1	0	0	0	0	0	0	0
2	1	0	0	0	0	1	0	0
3	1	0	0	0	1	0	0	0
4	1	0	0	1	0	0	0	0
5	1	0	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0
7	0	0	0	0	0	1	0	0
8	1	0	0	0	0	0	0	0
9	0	1	0	0	0	0	0	0

Table 9 – Interconnections between the metabolite C and all the reactions for each structure of the topological scope as given by the file *Structures_Metabolite_C.txt*.

Structure Number	Reaction 1				Reaction 2			
	Direct (+R1)		*Inverse (-R1)*		*Direct (+R1)*		*Inverse (-R1)*	
	Reactant	*Product*	*Reactant*	*Product*	*Reactant*	*Product*	*Reactant*	*Product*
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0
9	0	0	0	0	0	1	0	0

The information in the previous Tables is used to draw the structures using a directed and bipartite graph representation (Figure 4).

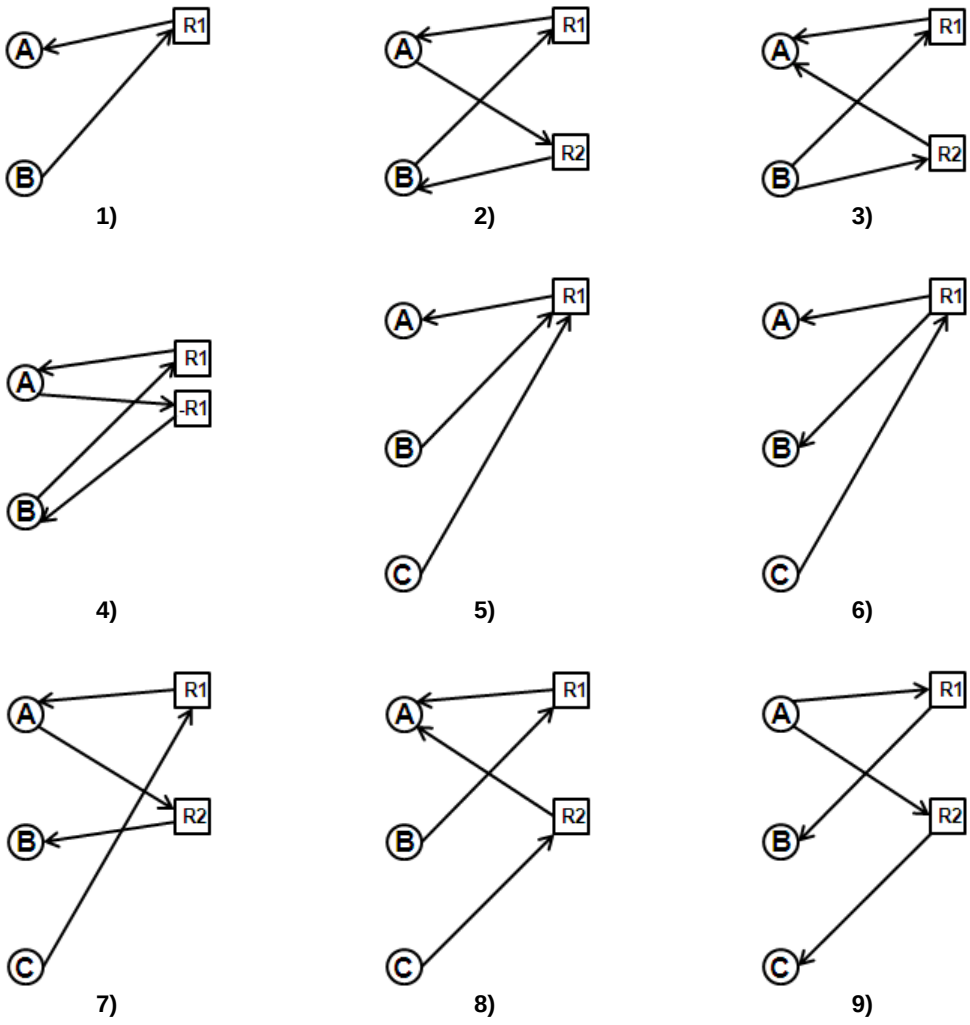


Figure 4 –Possible topological structures of the scope set in this manual, represented in an oriented bipartite graph.

Table 10 provides general information regarding the structures as given by file *Structures_General.txt*. The rows correspond to the structures and the columns to the following characteristics: 1) generation layer; 2) number of metabolites; 3) number of reactions; 4) number of reversible reactions; 5) number of ancestors; 6) number of parents; 7) number of children; 8) number of offspring; 9) number of interconnections between the nodes of the structure; 10) maximum possible number of interconnections for a structure with the same number of nodes; 11) number of different connections between metabolite and reaction nodes (note: both reversible and irreversible reactions are considered); 12) maximum possible number of different connections between metabolite and reaction nodes (both reversible and irreversible reactions are considered) for a structure with the same number of nodes.

Table 10 – General information regarding the structures as given by the file *Structures_General.txt*.

Row Number	Column Number					
	1	**2**	**3**	**4**	**5**	**6**
Structure Number	***Number Generation***	***Number Metabolites***	***Number Reactions***	***Number Rev. Reactions***	***Number Ancestors***	***Number Parents***
1	1	2	1	0	0	0
2	2	2	2	0	1	1
3	2	2	2	0	1	1
4	2	2	1	1	1	1
5	2	3	1	0	1	1
6	2	3	1	0	1	1
7	2	3	2	0	1	1
8	2	3	2	0	1	1
9	2	3	2	0	1	1
Row Number	***Column Number***					
	7	***8***	***9***	***10***	***11***	***12***
Structure Number	***Number Children***	***Number Offspring***	***Number intercon.***	***Max. Number intercon.***	***Number connect. Met/React***	***Max. Number connect. Met/React***
1	8	8	2	4	2	2
2	0	0	4	8	4	4
3	0	0	4	8	4	4
4	0	0	4	4	2	2
5	0	0	3	6	3	3
6	0	0	3	6	3	3
7	0	0	4	12	4	6
8	0	0	4	12	4	6
9	0	0	4	12	4	6

Table 11 presents the information in the file *Structures_RevReact.txt*, which indicates the reversibility of the reaction nodes of the structures. The rows correspond to the structures and the columns to the reaction nodes. For a reversible reaction, the corresponding cell is filled with “1”; otherwise, is filled with “0”. The cells of column numbers higher than the total number of reaction nodes of a given structure are all filled with “0”´s.

Table 11 – Information regarding the reversibility of the reaction nodes of the structures within the scope as stored in the file *Structures_RevReact.txt*.

Structure Number	Reaction 1	Reaction 2
1	0	0
2	0	0
3	0	0
4	0	0
5	1	0
6	0	0
7	0	0
8	0	0
9	0	0

The data needed to represent the network of directly related structures is stored in file *Family_Structures.txt* (Table 12). The rows represent the parent structure and the columns represent the child structures. If two structures have no relationship parent/child, the corresponding cell is filled with “0”. Otherwise, a number is introduced to represent the structural component that must be added to the parent to obtain the child:

(i) A reactant or product to a reaction (new internal connection):
- Irreversible – cell filled with "1";
- Reversible – cell filled with "2";

(ii) A new reaction converting one metabolite into another one – cell filled with "3";

(iii) The reverse of a reaction – cell filled with "4";

(iv) A new metabolite added to a reaction:
- Irreversible – cell filled with "5";
- Reversible – cell filled with "6";

(v) A new reaction connecting a new metabolite to an existing one to form a pair reactant/product – cell filled with "7".

Table 12 – Family relationships parent/child between the structures as stored in file *Family_Structures.txt*. The network of directly related structures is obtained by using this data.

Parent Structure	Child Structure								
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***
1	0	3	3	4	5	5	7	7	7
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

File *InclusionMatrix_Structures.txt* contains the inclusion matrix (see Table 13). It indicates if a given structure can be found among more complex structures. The row represent the ascendant structures (less complex, i.e., of a lower generation) and the columns the descendant structures (higher complexity). If an ascendant structure can be found within a descendant structure, the corresponding cell is filled with "1"; otherwise, is filled with "0". This data is used to obtain the descendants network.

Table 13 – Inclusion matrix as found in file *InclusionMatrix_Structures.txt*. It indicates if a given structure can be found within a more complex structure. It is used to obtain the descendants network.

Ascendant Structure	Descendant Structure								
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***
1	0	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

The degree of the metabolite nodes for all the structures is provided in the file *MetDegreeRevStru.txt* (see Table 14). The rows represent the structures and the columns their metabolite nodes. For each metabolite the following data is given: i) total number of arcs starting (out) and coming (in) to the node; ii) number of incoming arcs (in); and iii) number of outgoing arcs (out).

Table 14 – Degree (total, in and out) of each metabolite node of a given structure as given by file *MetDegreeRevStru.txt*.

Structure Number	Degree of the Metabolites								
	A			B			C		
	Total	*In*	*Out*	*Total*	*In*	*Out*	*Total*	*In*	*Out*
1	1	1	0	1	0	1	0	0	0
2	2	1	1	2	1	1	0	0	0
3	2	2	0	2	0	2	0	0	0
4	2	1	1	2	1	1	0	0	0
5	1	1	0	1	0	1	1	0	1
6	1	1	0	1	1	0	1	0	1
7	2	1	1	1	1	0	1	0	1
8	2	2	0	1	0	1	1	0	1
9	2	0	2	1	1	0	1	1	0

The degree of the reaction nodes for all the structures is given by file *ReactDegreeRevStru.txt* (see Table 15). The rows represent the structures and the columns their reaction nodes. The reaction nodes consist of one or two unidirectional reactions, depending on their reversibility. The information regarding the degree (total, in and out) includes all the arcs connected to all the unidirectional reactions of a given reaction node.

Table 15 – Degree (total, in and out) of each reaction node of a given structure as given by file *ReactDegreeRevStru.txt*.

Structure Number	Degree of the Reactions					
	R1			R2		
	Total	*In*	*Out*	*Total*	*In*	*Out*
1	2	1	1	0	0	0
2	2	1	1	2	1	1
3	2	1	1	2	1	1
4	4	2	2	0	0	0
5	3	2	1	0	0	0
6	3	1	2	0	0	0
7	2	1	1	2	1	1
8	2	1	1	2	1	1
9	2	1	1	2	1	1

Two nodes of the same type (metabolites, irreversible reactions or reversible reactions) of a given structure are topologically equivalent if they have the same degree (total and in) and equivalent topological neighbourhoods. The topological equivalence of the neighbourhoods of the two nodes can be tested as follows. Two graph representations of the structure are prepared in which the two nodes are kept at the same position, one

in the first graph and the other in the second graph. If there is a set of permutations for the other nodes of the structure that allows the two graphs to have the same representation, i.e., the same label, then the neighbourhoods are declared to be equivalent. Figure 5 shows an example of two graphs representing the same structure, where the metabolite nodes B and C, and the reaction nodes R1 and R2, are topologically equivalent.

Figure 5 – Two graph representations of the same structure, showing the topological equivalence of the metabolite nodes A and B and the reaction nodes R1 and R2.

The information regarding the equivalent topological nodes is relevant to study the prevalence of specific metabolites (e.g., AMP, ATP and ADP) and of specific reactions (e.g., AMP + ATP ↔ 2 ADP) in instances at a particular node position of a given structure. Considering the example shown in Figure 5, the specific metabolite AMP appearing at the node position B is equivalent to that appearing at the node position C. So, equivalent topological nodes should be analysed together. Table 16 shows the information regarding the topologically equivalent metabolite nodes of each structure as stored in file *EquiMetRevStru.txt*. The rows correspond to the structures and the columns to pairs of metabolite nodes. If two metabolite nodes are topologically equivalent, the corresponding cell is filled with "1"; otherwise, is filled with "0". The pairs of metabolite nodes are listed as follows: metabolite node A, which is in the 1st position, is compared with the ones in the following positions (nodes B, C, …); metabolite node B (corresponds to the 2nd position) is compared to metabolites that are also in the following positions (nodes C, D, …), and so forth, until we compare the last two metabolite nodes. Considering the case where the metabolite node in the position *X* is compared to the metabolite node in a higher position *Y*, the information regarding the topological equivalence of the two nodes of structure *i*, $Equ(i)_{X/Y}$, will be stored in the column number, $CNum\left[Equ(i)_{X/Y}\right]$, given by equation (1a):

$$CNum\left[Equ(i)_{X/Y}\right] = \left(\sum_{j=1}^{X-1}[NMet(i) - j]\right) + (Y - X) \quad \text{Eq. (1a)}$$

Where $NMet(i)$ is the number of metabolite nodes of structure *i*; and *j* is a given position of a metabolite node.

Table 16 – Information regarding the topologically equivalent metabolite nodes within each structure as stored in file *EquiMetRevStru.txt*.

Structure Number	Pair of Metabolites		
	A – B	***n.a.***	***n.a.***
1	0	0	0
2	1	0	0
3	0	0	0
4	1	0	0
	A – B	***A – C***	***B – C***
5	0	0	1
6	1	0	0
7	0	0	0
8	0	0	1
9	0	0	1

Table 17 shows the information regarding the topologically equivalent reaction nodes of each structure as stored in file *EquiReactRevStru.txt*. Similarly to the previous file, the rows correspond to the structures and the columns to pairs of reaction nodes. If two reaction nodes are topologically equivalent, the corresponding cell is filled with "1"; otherwise, is filled with "0". The pairs of reaction nodes are listed as follows: reaction node R1 (corresponds to the 1st position) is compared with the ones in the following positions (reaction nodes R2, R3, …); reaction node R2 (2nd position) is compared to reaction nodes that are also in the following positions (reaction nodes R3, R4, …), and so forth, till we compare the last two reaction nodes. Considering the case where the reaction node in the position *X* is compared to the reaction node in a higher position *Y*, the information regarding the topological equivalence of the two nodes of structure *i*, $Equ(i)_{X/Y}$, will be stored in the column number, $CNum\left[Equ(i)_{X/Y}\right]$, given by equation (1b):

$$CNum\left[Equ(i)_{X/Y}\right] = \left(\sum_{j=1}^{X-1}[NReact(i) - j]\right) + (Y - X) \quad \text{Eq. (1b)}$$

Where $NReact(i)$ is the number of reaction nodes of structure *i*; and *j* represents the position of a reaction node.

Table 17 – Information regarding the topologically equivalent reaction nodes within each structure as stored in file *ReactDegreeRevStru.txt*.

Structure Number	Pair of Reactions
	n.a.
1	0
	R1 – R2
2	1
3	1
	n.a.
4	0
5	0
6	0
	R1 – R2
7	0
8	1
9	1

The file *DifNodesStru.txt* (see Table 18) is used as a support or intermediate file for the executable program "Instances Counting". It stores information indicating if a structure has topologically different nodes by taking into account only their degree (total and in). This data helps decreasing the computational time needed to determine the structure of a newly found instance by the program "Instances Counting". The rows correspond to the structures while the columns correspond to the node type: i) metabolites, ii) irreversible reactions and iii) reversible reactions (note: the reversibility of the reactions allowed us to consider in practice the existence of three types of nodes to implement the algorithms). The default number introduced in the cells is "0". For structures with at least two nodes of a given type (for example, two irreversible reactions) and with different degrees, the corresponding cell is filled with "1". However, if there are at least two nodes of the same type with the same degree, the corresponding cell is kept at "0".

Table 18 – Information indicating if a structure has topologically different nodes by taking into account only their degree (total and in) as stored in the intermediate file *DifNodesStru.txt*.

Structure Number	Types of Nodes		
	Metabolites	***Irreversible Reactions***	***Reversible Reactions***
1	1	0	0
2	0	0	0
3	1	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	1	0	0
8	0	0	0
9	0	0	0

Table 19 shows the information contained in the intermediate file *in_nGen_nStru.txt* needed for the next executable programs. The first row shows the number of generations, $NGen_{max}$, and of structures within the topological scope. The next rows present the lowest and highest reference numbers of the structures of each generation.

Table 19 – Information contained in the intermediate file *in_nGen_nStru.txt* needed in the executable programs 2 to 6: number of generation and of structures within the scope, and lowest and highest reference numbers of the structures of each generation.

Row number	Additional Information	Generations, $NGen_{max}$	Structures
1	***Total number of***	2	9
	Generation	**Reference number of Structures**	
		Lowest	***Highest***
2	*1*	1	1
3	*2*	2	9

The intermediate file *in_nMet_nReact.txt* stores the maximum number of metabolites and of reactions nodes of the considered scope (see Table 20). This file is used as an input by the next executable programs.

Table 20 – Maximum number of metabolites and of reactions nodes of the considered scope as stored in the intermediate file *in_nMet_nReact.txt*.

Node type	Maximum Number
Metabolites, NMetmax	3
Reactions, NReact	2

3.3 – Instances Counting [*Instances Counting.exe*]

The executable program “Instances Counting” is obtained by compiling the code file *InstCount_v1.f90* with GFortran. This program obtains the instances within a metabolic network of all the structures of the defined scope. However, some of the instances obtained may be repeated. Thus, the outputs of “Instances Counting” are intermediate files used by the next executable program, “Organizing Instances”. The determination of the current number of instances was divided into these two programs in order to substantially decrease the computational time. A general description of the input and output files of the executable program “Instances Counting” is presented in the next Tables.

Table 21 – General description of the input files of the executable program "Instances Counting".

File		Created by	Description
Name	**Type**		
List_Metabolites.txt	Metabolic Network	*n.a.*	Metabolic network: list of names of all metabolites of the metabolic network
List_Reactions.txt	Metabolic Network	*n.a.*	Metabolic network: list of names of all reactions of the metabolic network
Products_X_Reactions.txt	Metabolic Network	*n.a.*	Metabolic network: products x reactions data matrix
Reactions_X_Reactants.txt	Metabolic Network	*n.a.*	Metabolic network: reactions x reactants data matrix
RevReact.txt	Metabolic Network	*n.a.*	Metabolic network: List with the information of the reversibility of the reactions
in_nGen_nStru.txt	Intermediate	*1) Structures Enumeration*	Information regarding the number of generations considered, the number of structures obtained, and the minimum and maximum reference numbers of the structures belonging to each one of the considered generations
in_nMet_nReact.txt	Intermediate	*1) Structures Enumeration*	Maximum number of metabolites and of reactions considered for the structures within the assumed number of generations
Structures_Metabolite_A.txt *Structures_Metabolite_B.txt* *Structures_Metabolite_C.txt* *Structures_Metabolite_D.txt* *Structures_Metabolite_E.txt* *Structures_Metabolite_F.txt*	Result	*1) Structures Enumeration*	The files store the pattern of the structures of the defined scope. Each file stores the part of the pattern corresponding to one of the metabolites
Structures_General.txt	Result	*1) Structures Enumeration*	General information regarding the structures such as: number of generation; number of metabolites, reactions and reversible reactions and number of interconnections
MetDegreeRevStru.txt	Result	*1) Structures Enumeration*	Degree (total, in and out) for each metabolite of a given structure
ReactDegreeRevStru.txt	Result	*1) Structures Enumeration*	Degree (total, in and out) for each reaction of a given structure
DifNodesStru.txt	Intermediate	*1) Structures Enumeration*	Information indicating if a given structure has different nodes of given type (metabolites, irreversible reactions and reversible reactions) by taking into account their degree (total and in)

Table 22 – General description of the output files of the executable program "Instances Counting".

File		Used by	Description
Name	**Type**		
MetInst.txt	Intermediate	*3) Organizing Instances*	Reference numbers of the metabolites of all the instances within the considered scope ordered according to the pattern stored for its structure obtained by the program "Structures Enumeration". NOTE: some of the instances may be repeated
ReactInst.txt	Intermediate	*3) Organizing Instances*	Reference numbers of the reactions of all the instances within the considered scope ordered according to the pattern stored for its structure obtained by the program "Structures Enumeration". NOTE: some of the instances may be repeated
InstStruct.txt	Intermediate	*3) Organizing Instances*	Reference number of the structure of all the instances. NOTE: some of the instances may be repeated
MetInstOrdRefNum.txt	Intermediate	*3) Organizing Instances*	Metabolites of all the instances within the considered scope ordered from the higher to the lower reference numbers. NOTE: some of the instances may be repeated
ReactInstOrdRefNum.txt	Intermediate	*3) Organizing Instances*	Reactions of all the instances within the considered scope ordered from the higher to the lower reference numbers. NOTE: some of the instances may be repeated
SumMetInst.txt	Intermediate	*3) Organizing Instances*	Sum of the reference numbers of the metabolites of each one of the instances found. NOTE: some of the instances may be repeated
SumReactInst.txt	Intermediate	*3) Organizing Instances*	Sum of the reference numbers of the reactions of each one of the instances found. NOTE: some of the instances may be repeated
OutRevStruct.txt	Intermediate	*3) Organizing Instances*	Number of instances for all the structures of the considered scope. NOTE: some of the instances may be repeated
numInstances.txt	Intermediate	*3) Organizing Instances*	Number of instances found within the considered scope. NOTE: some of the instances may be repeated

The metabolic network is an input of the program and is given in 5 files. In this document, the metabolic network of *Saccharomyces cerevisiae* [9] was used. For other metabolic

networks, two options can be followed: i) use the same names for the required files or ii) change the names of the files in the original code file, *InstCount_v1.f90*, of the package and compile new executable programs. The list with the names of all the metabolites of the metabolic network is given by the file *List_Metabolites.txt* (Table 23). The last row of the file must be filed with the word "End" to allow the program to determine the end of the list. The metabolites are attributed a reference number (i.e., a label) that equals its row number.

Table 23 – List with the metabolite names of the metabolic network as stored in file *List_Metabolites.txt*.

Row	Metabolite
…	…
194	"Adenosine"
…	…
199	"ADP"
…	…
224	"AMP"
…	…
233	"ATP"
…	…
433	"GDP"
…	…
464	"GMP"
…	…
469	"GTP"
…	…
711	"Orthophosphate"
…	…
992	End

Similarly, the list with the equations of all the reactions of the metabolic network is given by the File *List_Reactions.txt* (Table 24). For reversible reactions, only the direct unidirectional reaction is considered. The last row of the file must be filled with the word "End" to allow the program to determine the end of the list. The reactions are attributed a reference number that equals its row number.

Table 24 – List with the equations of the reactions of the metabolic network as stored in file *List_Reactions.txt*.

Row	Reaction
…	…
31	"AMP" + "ATP" -> 2.*"ADP"
32	"AMP" + "GTP" -> "ADP" + "GDP"
…	…
308	"ADP" + "Orthophosphate" -> "ATP"
…	…
401	"ATP" + "GMP" -> "ADP" + "GDP"
…	…
902	"AMP" -> "Adenosine" + "Orthophosphate"
…	…
1059	End

Information regarding the metabolites that are products in the reactions of the network is given in file *Products_X_Reactions.txt* (see Table 25), representing a products/reactions incidence matrix. Once again, we only consider the direct unidirectional reaction for reversible reactions. If a metabolite is a product of a reaction, the corresponding cell is filled with “1”; otherwise, is filled with “0”.

Table 25 – Products x reactions data matrix as stored in file *Products_X_Reactions.txt*.

Product	Reaction									
	…	31	32	…	308	…	401	…	902	…
…	…	…	…	…	…	…	…	…	…	…
194	…	0	0	…	0	…	0	…	1	…
…	…	…	…	…	…	…	…	…	…	…
199	…	1	1	…	0	…	1	…	0	…
…	…	…	…	…	…	…	…	…	…	…
224	…	0	0	…	0	…	0	…	0	…
…	…	…	…	…	…	…	…	…	…	…
233	…	0	0	…	1	…	0	…	0	…
…	…	…	…	…	…	…	…	…	…	…
433	…	0	1	…	0	…	1	…	0	…
…	…	…	…	…	…	…	…	…	…	…
464	…	0	0	…	0	…	0	…	0	…
…	…	…	…	…	…	…	…	…	…	…
469	…	0	0	…	0	…	0	…	0	…
…	…	…	…	…	…	…	…	…	…	…
711	…	0	0	…	0	…	0	…	1	…
…	…	…	…	…	…	…	…	…	…	…

File *Reactions_X_Reactants.txt*, representing a reactions/products incidence matrix. It shows which metabolites are reactants in the reactions. As before, reversible reactions are represented only by its direct unidirectional reaction. If a metabolite is a reactant of a reaction, the corresponding cell is filled with “1”; otherwise, is filled with “0”.

Table 26 – Reactions x reactants data matrix as given by file *Reactions_X_Reactants.txt*.

Reaction	Reactant																
	…	194	…	199	…	224	…	233	…	433	…	464	…	469	…	711	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
31	…	0	…	0	…	1	…	1	…	0	…	0	…	0	…	0	…
32	…	0	…	0	…	1	…	0	…	0	…	0	…	1	…	0	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
308	…	0	…	1	…	0	…	0	…	0	…	0	…	0	…	1	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
401	…	0	…	0	…	0	…	1	…	0	…	1	…	0	…	0	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
902	…	0	…	0	…	1	…	0	…	0	…	0	…	0	…	0	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

The list with the information regarding the reversibility of reactions of the metabolic network is given by file *RevReact.txt*. If a reaction is reversible, the corresponding cell is filled with “1”; if it is irreversible, the cell is filled with “0”.

Table 27 – Information regarding the reversibility of reactions of the metabolic network as shown in file *RevReact.txt*: "1" is given for reversible reactions and "0" for irreversible.

Reaction	Reversibility
…	…
31	1
32	1
…	…
308	0
…	…
401	1
…	…
902	0
…	…

The other input files are described in section 3.2. The outputs of "Instances Counting" are intermediate files used by the next executable program, "Organizing Instances". The file *MetInst.txt* shows the reference numbers of the metabolites of the instances found with the program (see Table 28). The rows correspond to the instances, and the columns to the metabolite node positions. The reference number of the metabolites are introduced according to the pattern stored for the corresponding structure of the instance as obtained in the executable program "Structures Enumeration". Note that some of the instances found may be repeated.

Table 28 – Reference number of the metabolites of the instances found by the executable program "Instances Counting". The metabolites are ordered according to the pattern of the structure of the instance as obtained in the program "Structures Enumeration". This data is stored in file *MetInst.txt*.

Number of Instance*	Reference Number of the Metabolite Node		
	a	***b***	***c***
1	1	9	0
2	1	145	0
3	156	1	0
4	165	1	0
5	199	1	0
…	…	…	…

*Note: some of the instances may be repeated

The file *ReactInst.txt* shows the reference numbers of the reactions of the instances found with the program (see Table 29). Similarly to the file *MetInst.txt*, the rows correspond to the instances, and the columns to the reaction node positions. The reference number of the reactions are introduced according to the pattern stored for the corresponding structure of the instance, as obtained in the executable program "Structures Enumeration".

Table 29 – Reference number of the reactions of the instances found by the executable program "Instances Counting". The reactions are ordered according to the pattern of the structure of the instance as obtained in the program "Structures Enumeration". This data is stored in file *ReactInst.txt*.

Number of Instance*	Reference Number of the Reaction Node	
	r1	***r2***
1	18	0
2	23	0
3	959	0
4	28	0
5	959	0
…	…	…

*Note: some of the instances may be repeated

The number of the structure of each instance is listed in file *InstStruct.txt* (see Table 30). Once again, the row number corresponds to the instance.

Table 30 – List with the structure number of each instance found by the executable program "Instances Counting". This data is stored in file *InstStruct.txt*.

Number of Instance*	Number of Structure
1	4
2	4
3	1
4	1
5	1
…	…

*Note: some of the instances may be repeated

Similarly to *MetInst.txt*, file *MetInstOrdRefNum.txt* also stores information regarding the metabolites of the instances (Table 31). However, the metabolites are ordered from the higher to the lower reference numbers.

Table 31 – Reference number of the metabolites of the instances found by the executable program "Instances Counting". The metabolites are ordered from the higher to the lower reference numbers. This data is stored in file *MetInstOrdRefNum.txt*.

Number of Instance	Reference Number of the Metabolite Node		
	a'	***b'***	***c'***
1	9	1	0
2	145	1	0
3	156	1	0
4	165	1	0
5	199	1	0
…	…	…	…

*Note: some of the instances may be repeated

The file *ReactInstOrdRefNum.txt* stores the reactions of the instances. Differently from file *ReactInst.txt*, the reactions are ordered from the higher to the lower reference numbers (see Table 32).

Table 32 – Reference number of the reactions of the instances found by the executable program "Instances Counting". The reactions are ordered from the higher to the lower reference numbers. This data is stored in file *ReactInstOrdRefNum.txt*.

Number of Instance*	Reference Number of the Reaction Node	
	r1'	***r2'***
1	18	0
2	23	0
3	959	0
4	28	0
5	959	0
…	…	…

*Note: some of the instances may be repeated

File *SumMetInst.txt* stores a list with the sum of the reference numbers of the metabolites of the instances (Table 33). Similarly, file *SumReactInst.txt* stores a list with the sum of the reference numbers of the reactions of the instances (Table 34).

Table 33 – Sum of the reference numbers of the metabolites found by the executable program "Instances Counting". This data is stored in file *SumMetInst.txt*.

Number of Instance*	Sum of the Reference Numbers of the Metabolite Nodes
1	10
2	146
3	157
4	166
5	200
…	…

*Note: some of the instances may be repeated

Table 34 – Sum of the reference numbers of the reactions found by the executable program "Instances Counting". This data is stored in file *SumReactInst.txt*.

Number of Instance*	Sum of the Reference Numbers of the Reaction Nodes
1	18
2	23
3	959
4	28
5	959
…	…

*Note: some of the instances may be repeated

File *OutRevStruct.txt* gives a list with the number of instances found by the program for each structure of the topological network (Table 35). The total number of instances found is stored in file *numInstances.txt* (Table 36). Once again, one must note that some of the instances found may be repeated.

Table 35 – Number of instances found of each structure within the scope by the executable program "Instances Counting". This data is stored in file *OutRevStruct.txt*.

Number of Structure	Number of Instances*
1	2 653
2	2 018
3	12 292
4	954
5	2 944
6	3 490
7	25 736
8	61 664
9	68 418

*Note: some of the instances may be repeated

Table 36 – Total number of instances found by the executable program "Instances Counting" and stored in file *numInstances.txt*.

Total Number of Instances*
180 169

*Note: some of the instances may be repeated

3.4 – Organizing Instances [*Organizing Instances.exe*]

The executable program "Organizing Instances" is obtained by compiling the code file *OrgInst.f90* with GFortran. This program operates on metabolic networks and has the following goals:

i. Obtain the real number of instances by determining the repeated ones;
ii. Organize the instances according to their structure

This program complements the previous one, "Instances Counting". It just compares instances having the same structure in order to determine the repeated ones, allowing to reduce the computational effort. A general description of the input and output files of the program "Organizing Instances" is presented in the following Tables.

Table 37 – General description of the input files of the executable program "Organizing Instances".

File		Created by	Description
Name	Type		
in_nGen_nStru.txt	Intermediate	*1) Structures Enumeration*	Information regarding the number of generations considered, the number of structures obtained, and the minimum and maximum reference numbers of the structures belonging to each one of the considered generations
in_nMet_nReact.txt	Intermediate	*1) Structures Enumeration*	Maximum number of metabolites and of reactions considered for the structures within the assumed number of generations
Structures_General.txt	Result	*1) Structures Enumeration*	General information regarding the structures such as: number of generation; number of metabolites, reactions and reversible reactions and number of interconnections
OutRevStruct.txt	Intermediate	*2) Instances Counting*	Number of instances for all the structures of the considered scope. NOTE: some of the instances may be repeated
numInstances.txt	Intermediate	*2) Instances Counting*	Number of instances found within the considered scope. NOTE: some of the instances may be repeated
InstStruct.txt	Intermediate	*2) Instances Counting*	Reference number of the structure of all the instances. NOTE: some of the instances may be repeated
MetInst.txt	Intermediate	*2) Instances Counting*	Reference numbers of the metabolites of all the instances within the considered scope ordered according to the pattern stored for its structure obtained by the program "Structures Enumeration". NOTE: some of the instances may be repeated
ReactInst.txt	Intermediate	*2) Instances Counting*	Reference numbers of the reactions of all the instances within the considered scope ordered according to the pattern stored for its structure obtained by the program "Structures Enumeration". NOTE: some of the instances may be repeated
MetInstOrdRefNum.txt	Intermediate	*2) Instances Counting*	Metabolites of all the instances within the considered scope ordered from the higher to the lower reference numbers. NOTE: some of the instances may be repeated
ReactInstOrdRefNum.txt	Intermediate	*2) Instances Counting*	Reactions of all the instances within the considered scope ordered from the higher to the lower reference numbers. NOTE: some of the instances may be repeated
SumMetInst.txt	Intermediate	*2) Instances Counting*	Sum of the reference numbers of the metabolites of each one of the instances found. NOTE: some of the instances may be repeated
SumReactInst.txt	Intermediate	*2) Instances Counting*	Sum of the reference numbers of the reactions of each one of the instances found. NOTE: some of the instances may be repeated

Table 38 – General description of the output files of the executable program "Organizing Instances".

File		Used by	Description
Name	Type		
OutRevStructReal.txt	Result	*4) StAllInstCount*	Real number of instances (naked frequency) for all the structures of the considered scope (does not include repeated instances)
numRefInstLowHigh.txt	Result	*4) StAllInstCount*	Lower and higher reference numbers of instances of a given structure. The instances are stored according to their structure: first the instances of structures of lower reference numbers
numInstancesReal.txt	Result	*4) StAllInstCount*	Real number of instances found within the considered scope (does not include repeated instances)
OrgMetInst.txt	Result	*4) StAllInstCount*	Reference numbers of the metabolites of all the instances within the considered scope. The metabolites of an instance are ordered according to the pattern stored for its structure obtained by the program "Structures Enumeration"
OrgReactInst.txt	Result	*4) StAllInstCount*	Reference numbers of the reactions of all the instances within the considered scope. The reactions of an instance are ordered according to the pattern stored for its structure obtained by the program "Structures Enumeration"

The input files are described in the previous sections. So, this section will only focus on the output files. The list with the number of instances -naked frequency, $f_N(i)$ - for the structures is given in file *OutRevStructReal.txt* (Table 39).

Table 39 – Naked frequency, f_N, of the structures in the metabolic network as given in file *OutRevStructReal.txt*.

Number of Structure	Number of Instances - Naked Frequency, $f_N(i)$
1	2 653
2	1 009
3	6 146
4	954
5	1 472
6	1 745
7	12 868
8	30 832
9	34 209

The instances are organized according to their structure and stored in the files *OrgMetInst.txt* and *OrgReactInst.txt*. The lowest and highest reference numbers of the instances of each of the structures is given in file *numRefInstLowHigh.txt* (see Table 40).

Table 40 – Lowest and highest reference numbers of the instances of each of the structures. This data is stored in file *numRefInstLowHigh.txt*.

Number of Structure	Reference Number of Instances	
	Lowest	***Highest***
1	1	2 653
2	2 654	3 662
3	3 663	9 808
4	9 809	10 762
5	10 763	12 234
6	12 235	13 979
7	13 980	26 847
8	26 848	57 679
9	57 680	91 888

The total number of instances obtained in the metabolic network within the considered scope is given in the file *numInstancesReal.txt* (Table 41).

Table 41 – Total number of instances obtained in the metabolic network within the considered scope as given in file *numInstancesReal.txt*.

Total Number of Instances
91 888

The file *OrgMetInst.txt* shows the reference numbers of the metabolites of the instances obtained in the metabolic network within the considered scope (see Table 42). The rows correspond to the instances, and the columns to the metabolite node positions. The reference number of the metabolites are introduced according to the pattern stored for the corresponding structure of the instance as obtained in the executable program "Structures Enumeration".

Table 42 – Reference number of the metabolites of the instances found in the metabolic network within the considered scope. The metabolites are ordered according to the pattern of the structure of the instance as obtained in the program "Structures Enumeration". This data is stored in file *OrgMetInst.txt*.

Number of Instance	Reference Number of the Metabolite Node		
	a	***b***	***c***
1	156	1	0
2	165	1	0
3	199	1	0
4	1	233	0
5	1	420	0
…	…	…	…

The file *OrgReactInst.txt* shows the reference numbers of the reactions of the instances found in the metabolic network within the considered scope (see Table 43). Similarly to the file *OrgMetInst.txt*, the rows correspond to the instances, and the columns to the reaction node positions. The reference number of the reactions are introduced according to the pattern stored for the corresponding structure of the instance as obtained in the executable program "Structures Enumeration".

Table 43 – Reference number of the reactions of the instances found in the metabolic network within the considered scope. The reactions are ordered according to the pattern of the structure of the instance as obtained in the program "Structures Enumeration". This data is stored in file *OrgReactInst.txt*.

Number of Instance	Reference Number of the Reaction Node	
	r1	***r2***
1	959	0
2	28	0
3	959	0
4	22	0
5	22	0
…	…	…

3.5 – Stand-Alone Instances [*StAI-InstCount.exe*]

The executable program "StAI-InstCount" is obtained by compiling the code file *StAIInstCount_v1.f90* with GFortran. This program operates on metabolic networks and has the following goals:

i. Obtain the embedded frequency, $f_E(i)$, and the instances with overlaps by up to one metabolite;

ii. Determine the frequency of specific metabolites and reactions in the instances of a given structure: overall and in each node position (taking into account the existence of equivalent topological nodes).

A general description of the input (Table 44) and output (Table 45) files of this program is presented below.

Table 44 – General description of the input files of the executable program "StAI-InstCount".

File		Created by	Description
Name	**Type**		
List_Metabolites.txt	Metabolic Network	*n.a.*	Metabolic network: list of names of all metabolites of the metabolic network
List_Reactions.txt	Metabolic Network	*n.a.*	Metabolic network: list of names of all reactions of the metabolic network
in_nGen_nStru.txt	Intermediate	*1) Structures Enumeration*	Information regarding the number of generations considered, the number of structures obtained, and the minimum and maximum reference numbers of the structures belonging to each one of the considered generations
in_nMet_nReact.txt	Intermediate	*1) Structures Enumeration*	Maximum number of metabolites and of reactions considered for the structures within the assumed number of generations
Structures_General.txt	Result	*1) Structures Enumeration*	General information regarding the structures such as: number of generation; number of metabolites, reactions and reversible reactions and number of interconnections
EquiMetRevStru.txt	Result	*1) Structures Enumeration*	Information regarding the topologically equivalent metabolites of a given structure
EquiReactRevStru.txt	Result	*1) Structures Enumeration*	Information regarding the topologically equivalent reactions of a given structure
InclusionMatrix_Structures.txt	Result	*1) Structures Enumeration*	Inclusion matrix. Gives the information regarding the Offspring of a given structure
OutRevStructReal.txt	Result	*3) Organizing Instances*	Real number of instances (naked frequency) for all the structures of the considered scope (does not include repeated instances)
numInstancesReal.txt	Result	*3) Organizing Instances*	Real number of instances found within the considered scope (does not include repeated instances)
OrgMetInst.txt	Result	*3) Organizing Instances*	Reference numbers of the metabolites of all the instances within the considered scope. The metabolites of an instance are ordered according to the pattern stored for its structure obtained by the program "Structures Enumeration"
OrgReactInst.txt	Result	*3) Organizing Instances*	Reference numbers of the reactions of all the instances within the considered scope. The reactions of an instance are ordered according to the pattern stored for its structure obtained by the program "Structures Enumeration"
numRefInstLowHigh.txt	Result	*3) Organizing Instances*	Lower and higher reference numbers of instances of a given structure. The instances are stored according to their structure: first the instances of structures of lower reference numbers

Table 45 – General description of the output files of the executable program "StAI-InstCount".

File		Used by	Description
Name	**Type**		
ResultsRevStruct.txt	Result	*5) Preferential Paths* *6) Children Information*	Output data obtained for the structures, such as: naked frequency, embedded frequency and number of stand-alone instances
MetRevStruct.txt	Result	*n.a.*	Frequency of the specific metabolites in the instances of the structures
ReactRevStruct.txt	Result	*n.a.*	Frequency of the specific reactions in the instances of the structures
MetABCRevStruct.txt	Result	*n.a.*	Frequency of the specific metabolites in the instances of the structures at all the possible positions
React123RevStruct.txt	Result	*n.a.*	Frequency of the specific reactions in the instances of the structures at all the possible positions

The input files were described in the previous sections. Therefore, we only describe the output files of the program in this section.

The stand-alone instances or instances with overlaps by up to one metabolite are classified into one of the following types:

I. Instances with no overlaps – no metabolites or reactions shared with other instances of the same structure;

II. Instances with overlaps of one metabolite – shares 1 metabolite with at least one instance of the same structure.

Relevant data for the structures in the metabolic network is stored in file *ResultsRevStruct.txt* (Table 46). The rows correspond to the structures and the columns to relevant properties such as the naked and embedded frequency, number of stand-alone instances (type I and II, and the sum of both) and number of different specific metabolites and reactions of the network that appear in at least one instance of the structure.

Table 46 – Relevant properties of the structures in the metabolic network: naked and embedded frequency, number of stand-alone instances and number of different specific metabolites and reactions of the network that appear in at least one instance of the structure. This data is given by file *ResultsRevStruct.txt*.

Structure Number	Naked Frequency, $f_N(i)$	Embedded Frequency, $f_E(i)$	Number of Stand-Alone Instances			Number of Different Specific	
			Type I	*Type II*	*Type I + II*	*Metabolites*	*Reactions*
1	2 653	91 888	18	69	87	823	723
2	1 009	1 009	3	13	16	121	326
3	6 146	6 146	2	6	8	132	419
4	954	954	28	72	100	499	335
5	1 472	1 472	0	61	61	688	478
6	1 745	1 745	1	70	71	688	553
7	12 868	12 868	0	0	0	735	695
8	30 832	30 832	0	4	4	568	613
9	34 209	34 209	1	1	2	529	560

The frequencies of the metabolites *x* of the metabolic network in the instances of the structures *i* within the scope, $f_{Met\ x}(i)$, are stored in file *MetRevStruct.txt* (Table 47). The rows correspond to the structures and the columns to the metabolites.

Table 47 – Frequency of the metabolites of the metabolic network in instances of the structures as stored in file *MetRevStruct.txt*.

Structure Number	Metabolite																	
	…	*194*	*…*	*199*	*…*	*224*	*…*	*233*	*…*	*433*	*…*	*464*	*…*	*469*	*…*	*711*	*…*
1	…	10	…	149	…	133	…	295	…	17	…	10	…	24	…	111	…
2	…	1	…	193	…	3	…	227	…	0	…	1	…	0	…	35	…
3	…	0	…	2036	…	881	…	3940	…	7	…	1	…	14	…	159	…
4	…	4	…	64	…	11	…	62	…	8	…	4	…	4	…	55	…
5	…	3	…	99	…	118	…	428	…	12	…	5	…	20	…	60	…
6	…	5	…	196	…	256	…	254	…	18	…	14	…	15	…	169	…
7	…	198	…	1758	…	1817	…	2315	…	67	…	41	…	77	…	1871	…
8	…	205	…	2616	…	3985	…	4501	…	152	…	126	…	786	…	4715	…
9	…	55	…	9128	…	6967	…	25680	…	81	…	448	…	204	…	3561	…

Similarly, the frequencies of the reactions *x* of the metabolic network in the instances of the structures *i* within the scope, $f_{React\,x}(i)$, are stored in file *ReactRevStruct.txt* (Table 48).

Table 48 – Frequency of the reactions of the metabolic network in instances of the structures as stored in file *ReactRevStruct.txt*.

Structure Number	Reaction									
	...	***31***	***32***	...	***308***	...	***401***	...	***902***	...
1	...	0	0	...	2	...	0	...	2	...
2	...	0	0	...	81	...	0	...	1	...
3	...	0	0	...	3	...	0	...	0	...
4	...	2	4	...	0	...	0	...	0	...
5	...	0	0	...	1	...	4	...	0	...
6	...	0	0	...	0	...	0	...	1	...
7	...	0	0	...	386	...	0	...	260	...
8	...	0	0	...	1	...	0	...	106	...
9	...	0	0	...	4	...	0	...	8	...

The file *MetABCRevStruct.txt* gives the frequencies of the metabolites of the metabolic network in the instances of the structures at each possible position (Table 49). The columns correspond to the metabolites and the rows to the possible positions in the structures. These possible positions are equal to the maximum number of metabolite nodes considered for the topological scope, $NMet_{max}$. Considering a general structure *i*, the frequency of a given metabolite *x* at the position *j*, $f_{Met\,x}(i)_j$, is stored at the following row number, $RNum\left[f_{Met\,x}(i)_j\right]$ (eq. 2a):

$$RNum\left[f_{Met\,x}(i)_j\right] = [(i-1) \times NMet_{max}] + j \qquad \text{Eq. (2a)}$$

For topological equivalent metabolite nodes (see Table 16 in section 3.2), the results are stored at the cell of the lowest position node. For example, if the metabolite nodes A (1st position) and C (3rd position) are equivalent, the results will be stored at the cell of node A.

Table 49 – Frequencies of the metabolites of the network at each possible position of the structures. Equivalent topological nodes are taken into account. This data is stored in file *MetABCRevStruct.txt*.

Structure Number	Node Position	Metabolite																	
		…	***194***	…	***199***	…	***224***	…	***233***	…	***433***	…	***464***	…	***469***	…	***711***	…
1	***A***	…	4	…	136	…	127	…	7	…	13	…	8	…	0	…	105	…
	B	…	6	…	13	…	6	…	288	…	4	…	2	…	24	…	6	…
	C	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…
2	***A***	…	1	…	193	…	3	…	227	…	0	…	1	…	0	…	35	…
	B	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…
	C	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…
3	***A***	…	0	…	2031	…	881	…	4	…	7	…	1	…	0	…	158	…
	B	…	0	…	5	…	0	…	3936	…	0	…	0	…	14	…	1	…
	C	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…
4	***A***	…	4	…	64	…	11	…	62	…	8	…	4	…	4	…	55	…
	B	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…
	C	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…
5	***A***	…	1	…	82	…	118	…	5	…	8	…	5	…	0	…	50	…
	B	…	2	…	17	…	0	…	423	…	4	…	0	…	20	…	10	…
	C	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…
6	***A***	…	2	…	186	…	253	…	11	…	16	…	13	…	0	…	163	…
	B	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…
	C	…	3	…	10	…	3	…	243	…	2	…	1	…	15	…	6	…
7	***A***	…	22	…	913	…	751	…	1216	…	42	…	13	…	0	…	482	…
	B	…	143	…	420	…	1039	…	123	…	18	…	20	…	0	…	1014	…
	C	…	33	…	425	…	27	…	976	…	7	…	8	…	77	…	375	…
8	***A***	…	5	…	2464	…	3833	…	3	…	46	…	19	…	0	…	4697	…
	B	…	200	…	152	…	152	…	4498	…	106	…	107	…	786	…	18	…
	C	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…
9	***A***	…	12	…	39	…	12	…	25608	…	4	…	0	…	204	…	5	…
	B	…	43	…	9089	…	6955	…	72	…	77	…	448	…	0	…	3556	…
	C	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…	0	…

Similarly, the file *React123RevStruct.txt* gives the frequencies of the reactions of the metabolic network in the instances of the structures at each possible position (Table 50). The columns correspond to the reactions and the rows to the possible positions in the structures. These possible positions are equal to the maximum number of reaction nodes considered for the topological scope, $NReact_{max}$. Considering a general structure *i*, the frequency of a given reaction *x* at the position *j*, $f_{React\,x}(i)_j$, is stored at the following row number, $RNum\left[f_{React\,x}(i)_j\right]$ (eq. 2b):

$$RNum\left[f_{React\,x}(i)_j\right] = [(i-1) \times NReact_{max}] + j \qquad \text{Eq. (2b)}$$

For topological equivalent reaction nodes (see Table 17 in section 3.2), the results are stored at the cell of the lowest position node. For example, if the reaction nodes R1 (1st position) and R3 (3rd position) are equivalent, the results will be stored at the cell of node R1.

Table 50 – Frequencies of the reactions of the network at each possible position of the structures. Equivalent topological nodes are taken into account. This data is stored in file *React123RevStruct.txt*.

Structure Number	Node Position	Reaction									
		…	*31*	*32*	*…*	*308*	*…*	*401*	*…*	*902*	*…*
1	***R1***	…	0	0	…	2	…	0	…	2	…
	R2	…	0	0	…	0	…	0	…	0	…
2	***R1***	…	0	0	…	81	…	0	…	1	…
	R2	…	0	0	…	0	…	0	…	0	…
3	***R1***	…	0	0	…	3	…	0	…	0	…
	R2	…	0	0	…	0	…	0	…	0	…
4	***R1***	…	2	4	…	0	…	4	…	0	…
	R2	…	0	0	…	0	…	0	…	0	…
5	***R1***	…	0	0	…	1	…	0	…	0	…
	R2	…	0	0	…	0	…	0	…	0	…
6	***R1***	…	0	0	…	0	…	0	…	1	…
	R2	…	0	0	…	0	…	0	…	0	…
7	***R1***	…	0	0	…	324	…	0	…	10	…
	R2	…	0	0	…	62	…	0	…	250	…
8	***R1***	…	0	0	…	1	…	0	…	106	…
	R2	…	0	0	…	0	…	0	…	0	…
9	***R1***	…	0	0	…	4	…	0	…	8	…
	R2	…	0	0	…	0	…	0	…	0	…

3.6 – Preferential Paths [*Preferential Paths.exe*]

The executable program “Preferential Paths” is obtained by compiling the code file *PrefPaths.f90* with GFortran. This program obtains the preferential path within the network of directly related structures by starting from a given structure, i.e., obtains the most frequent descendants by generation of a given structure. It operates on metabolic networks. A general description of the input (Table 51) and output (Table 52) files of this program is presented below.

Table 51 – General description of the input files of the executable program “Preferential Paths”.

File		Created by	Description
Name	**Type**		
in_nGen_nStru.txt	Intermediate	*1) Structures Enumeration*	Information regarding the number of generations considered, the number of structures obtained, and the minimum and maximum reference numbers of the structures belonging to each one of the considered generations
Structures_General.txt	Result	*1) Structures Enumeration*	General information regarding the structures such as: number of generation; number of metabolites, reactions and reversible reactions and number of interconnections
Family_Structures.txt	Result	*1) Structures Enumeration*	Matrix with the family relationships Parent/Child (network of directly related structures)
ResultsRevStruct.txt	Result	*4) StAI-InstCount*	Output data obtained for the structures, such as: naked frequency, embedded frequency and number of stand-alone instances

Table 52 – General description of the output files of the executable program “Preferential Paths”.

File		Used by	Description
Name	**Type**		
RankStru.txt	Result	*n.a.*	Stores the relevant ranks of the prevalence (naked frequency) for each structure: i) absolute rank; ii) generation rank
PrefPath.txt	Result	*n.a.*	Preferential path in the network of directly related structures starting from a given structure. General information of the initial structure is stored in the first columns; the following columns stores information of the most frequent successive descendant.

The input files are described in the previous sections. Consequently, we only describe the output files of the program in this section.

The prevalence of a given structure *i* is determined by its naked frequency, $f_N(i)$. So, a higher $f_N(i)$ leads to a higher prevalence of the structure in the metabolic network. The file *RankStru.txt* stores the rank of the prevalence of the structures within the considered scope (Table 53). The rows correspond to the structures and the columns to the ranks of prevalence: i) "Absolute", by taking into account all the structures and ii) "Generation", by comparing only the $f_N(i)$ of the structures of the same generation.

Table 53 – Ranking the prevalence of the structures in the metabolic network by taking into account their naked frequency, $f_N(i)$. This data is stored in file *RankStru.txt*.

Structure Number	Ranking (naked frequency, $f_N(i)$)	
	Absolute	***Generation***
1	5	1
2	8	7
3	4	4
4	9	8
5	7	6
6	6	5
7	3	3
8	2	2
9	1	1

Starting from a given structure, the preferential path within the network of directly related structures is obtained by successively jumping to the child structure with the highest prevalence. These preferential paths are stored in file *PrefPath.txt*. The rows correspond to the reference number of the structures and the first four columns provide information regarding these structures, namely: generation, naked frequency ($f_N(i)$) and ranking of prevalence (absolute and generation). The next columns relate to the prevalent descendants of the structures, starting from the child and continuing to the offspring structures of higher generations. For each descendant structure, five columns are attributed: generation, number of the structure, naked frequency ($f_N(i)$) and ranking of prevalence (absolute and generation). Table 54 shows the data contained in file *PrefPath.txt* for the scope considered in this manual of only 2 generations.

Table 54 – Preferential path starting from a given structure. Information regarding the starting structure and the prevalent descendants is given, such us: generation, number of structure, naked frequency ($f_N(i)$) and prevalent rankings (absolute and generation). This data is stored in file *PrefPath.txt*.

Starting Structure					Descendant Structure				
Number Stru.	Gen.	$f_N(i)$	Ranking ($f_N(i)$)		Gen.	Number Stru.	$f_N(i)$	Ranking ($f_N(i)$)	
			Abs.	*Gen.*				*Abs.*	*Gen.*
1	1	2653	5	1	2	9	34209	1	1
2	2	1009	8	7	0	0	0	0	0
3	2	6146	4	4	0	0	0	0	0
4	2	954	9	8	0	0	0	0	0
5	2	1472	7	6	0	0	0	0	0
6	2	1745	6	5	0	0	0	0	0
7	2	12868	3	3	0	0	0	0	0
8	2	30832	2	2	0	0	0	0	0
9	2	34209	1	1	0	0	0	0	0

For scopes with a higher number of generations, $NGen_{max}$, the number of columns in the file *PrefPath.txt*, $NumCol[file]$, would be equal to (eq. 3):

$$NumCol[file] = 4 + 5 \times (NGen_{max} - 1) \quad \text{Eq. (3)}$$

3.7 – Children Information [*Children Information.exe*]

The executable program "Children Information" is obtained by compiling the code file *ShannonEntropy.f90* with GFortran. This program operates on metabolic networks and has the following goals:

i. Obtain the ratios of the naked frequencies of the Childs, Rf_N;

ii. Determine the children information, ΔH, for each one of the considered Parent/Child relationships. The value of ΔH is null if all the children are equally frequent, and high if the distribution of children frequencies is very uneven.

A general description of the input (Table 55) and output (Table 56) files of this program is presented below.

Table 55 – General description of the input files of the executable program "Children Information".

File		Created by	Description
Name	**Type**		
in_nGen_nStru.txt	Intermediate	*1) Structures Enumeration*	Information regarding the number of generations considered, the number of structures obtained, and the minimum and maximum reference numbers of the structures belonging to each one of the considered generations
in_nMet_nReact.txt	Intermediate	*1) Structures Enumeration*	Maximum number of metabolites and of reactions considered for the structures within the assumed number of generations
Structures_General.txt	Result	*1) Structures Enumeration*	General information regarding the structures such as: number of generation; number of metabolites, reactions and reversible reactions and number of interconnections
Family_Structures.txt	Result	*1) Structures Enumeration*	Matrix with the family relationships Parent/Child (network of directly related structures)
ResultsRevStruct.txt	Result	*4) StAI-InstCount*	Output data obtained for the structures, such as: naked frequency, embedded frequency and number of stand-alone instances

Table 56 – General description of the output files of the executable program "Children Information".

File		Used by	Description
Name	**Type**		
RfN_AllRel.txt	Result	*n.a.*	Stores the ratio of frequencies, Rf_N, of the Childs - all Parent/Child relationships considered - of a given Parent structures
RfN_I.txt	Result	*n.a.*	Stores the ratio of frequencies, Rf_N, of the Childs - Parent/Child relationship of type (i) - of a given Parent structures
RfN_II.txt	Result	*n.a.*	Stores the ratio of frequencies, Rf_N, of the Childs - Parent/Child relationship of type (ii) - of a given Parent structures
RfN_III.txt	Result	*n.a.*	Stores the ratio of frequencies, Rf_N, of the Childs - Parent/Child relationship of type (iii) - of a given Parent structures
RfN_IV.txt	Result	*n.a.*	Stores the ratio of frequencies, Rf_N, of the Childs - Parent/Child relationship of type (iv) - of a given Parent structures
RfN_V.txt	Result	*n.a.*	Stores the ratio of frequencies, Rf_N, of the Childs - Parent/Child relationship of type (v) - of a given Parent structures
ChildrenInf_AllRel.txt	Result	*n.a.*	For a given Parent structure, stores relevant information regarding its Childs - all Parent/Child relationships considered: number of Childs; maximum entropy; sum of the naked frequencies of the Childs; entropy of Shannon; children information
ChildrenInf_I.txt	Result	*n.a.*	For a given Parent structure, stores relevant information regarding its Childs - Parent/Child relationship of type (i): number of Childs; maximum entropy; sum of the naked frequencies of the Childs; entropy of Shannon; children information
ChildrenInf_II.txt	Result	*n.a.*	For a given Parent structure, stores relevant information regarding its Childs - Parent/Child relationship of type (ii): number of Childs; maximum entropy; sum of the naked frequencies of the Childs; entropy of Shannon; children information
ChildrenInf_III.txt	Result	*n.a.*	For a given Parent structure, stores relevant information regarding its Childs - Parent/Child relationship of type (iii): number of Childs; maximum entropy; sum of the naked frequencies of the Childs; entropy of Shannon; children information
ChildrenInf_IV.txt	Result	*n.a.*	For a given Parent structure, stores relevant information regarding its Childs - Parent/Child relationship of type (iv): number of Childs; maximum entropy; sum of the naked frequencies of the Childs; entropy of Shannon; children information
ChildrenInf_V.txt	Result	*n.a.*	For a given Parent structure, stores relevant information regarding its Childs - Parent/Child relationship of type (v): number of Childs; maximum entropy; sum of the naked frequencies of the Childs; entropy of Shannon; children information

A structure parent n has a relationship with its children of type x, which corresponds to the structural component added to the parent: (i) a reactant or product to a reaction; (ii) a new reaction converting one metabolite into one other; (iii) the reverse of a reaction; (iv) a new metabolite; (v) a new reaction connecting a new metabolite to an existing one to form a pair reactant/product. Additionally, one can consider a type (vi), which includes all the previous relationships parent/child. We define children information of structure n for a parent/child relationship x, $\Delta H(n)_x$, as (eq. 4):

$$\Delta H(n)_x = H_{max}(n)_x - H(n)_x =$$
$$= -log_2\left(\frac{1}{N_{Ch}(n)_x}\right) - \left(-\sum_i Rf_N(j)_{n,x}\ log_2 Rf_N(j)_{n,x}\right),\quad \forall j \in Ch(n)_x \qquad \text{Eq. (4)}$$

where $H_{max}(n)_x$ is the maximum possible (or feasible) entropy for structure n for a parent/child relationship x, $H(n)_x$ is the Shannon entropy of structure n for a parent/child relationship x, $N_{Ch}(n)_x$ is the number of children of structure n with a parent/child relationship x, and $Ch(n)_x$ is the set of all the children of structure n with a parent/child relationship x. Furthermore, $Rf_N(j)_{n,x}$ is the ratio of frequencies of a child j of structure n with a parent/child relationship x and is obtained by (eq. 5):

$$Rf_N(j)_{n,x} = \frac{f_N(j)_{n,x}}{\sum_k f_N(k)_{n,x}},\quad \forall k \in Ch(n)_x \qquad \text{Eq. (5)}$$

where $f_N(j)_{n,x}$ is the naked frequency of child j of structure n with a parent/child relationship x. The value of $\Delta H(n)_x$ is null if all the children of structure n with a parent/child relationship x are equally frequent, and high if the distribution of children frequencies is high.

The input files are described in the previous sections. Consequently, we only describe the output files of the program in this section. The next Tables shows the ratio of frequencies, $Rf_N(j)_{n,x}$, for the children j of a given structure n for each type of parent/child relationships x. The rows correspond to the parent structures n and the columns to the child structures j.

Table 57 – Ratio of frequencies, $Rf_N(j)_{n,all}$, for the children *j* of a given parent structure *n* for all the types of parent/child relationships. This data is stored in file *RfN_AllRel.txt*.

Parent Structure	Child Structure								
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***
1	0.00000	0.01131	0.06887	0.01069	0.0165	0.01956	0.1442	0.34551	0.38336
2	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
5	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
6	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Table 58 – Ratio of frequencies, $Rf_N(j)_{n,i}$, for the children *j* of a given parent structure *n* for a parent/child relationship of type (i): a reactant or product added to a reaction (new internal connection) of the parent. This data is stored in file *RfN_I.txt*.

Parent Structure	Child Structure								
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***
1	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
5	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
6	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Table 59 – Ratio of frequencies, $Rf_N(j)_{n,ii}$, for the children *j* of a given parent structure *n* for a parent/child relationship of type (ii): a new reaction converting one metabolite into one other added to the parent. This data is stored in file *RfN_II.txt*.

Parent Structure	Child Structure								
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***
1	0.00000	0.14102	0.85898	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
5	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
6	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Table 60 – Ratio of frequencies, $Rf_N(j)_{n,iii}$, for the children *j* of a given parent structure *n* for a parent/child relationship of type (iii): the reverse of a reaction added to the parent. This data is stored in file *RfN_III.txt*.

Parent Structure	Child Structure								
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***
1	0.00000	0.00000	0.00000	1.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
5	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
6	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Table 61 – Ratio of frequencies, $Rf_N(j)_{n,iv}$, for the children *j* of a given parent structure *n* for a parent/child relationship of type (iv): a new metabolite added to the parent. This data is stored in file *RfN_IV.txt*.

Parent Structure	Child Structure								
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***
1	0.00000	0.00000	0.00000	0.00000	0.45757	0.54243	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
5	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
6	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Table 62 – Ratio of frequencies, $Rf_N(j)_{n,v}$, for the children *j* of a given parent structure *n* for a parent/child relationship of type (v): add to the parent a new reaction connecting a new metabolite to an existing one to form a pair reactant/product. This data is stored in file *RfN_V.txt*.

Parent Structure	Child Structure								
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***
1	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.16517	0.39574	0.43909
2	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
5	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
6	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

The next Tables show the data regarding the children information of the structures n for each type of parent/child relationships x. The rows correspond to the parent structures and the columns to the following properties: number of children, $N_{Ch}(n)_x$; maximum entropy, $H_{max}(n)_x$; sum of the naked frequencies of the children; entropy of Shannon $H(n)_x$; children information, $\Delta H(n)_x$.

Table 63 – Number of children - $N_{Ch}(n)_{All}$, maximum entropy - $H_{max}(n)_{All}$, sum of the naked frequencies of the children and children information - $\Delta H(n)_{All}$ - of the parent structures *n* for all the types of parent/child relationships. This data is stored in file *ChildrenInf_AllRel.txt*.

Parent Structure *n*	$N_{Ch}(n)_{All}$	$H_{max}(n)_{All}$	Sum of the f_N of the Children	$H(n)_{All}$	$\Delta H(n)_{All}$
1	8	3.00000	89235	2.08054	0.91946
2	0	0.00000	0	0.00000	0.00000
3	0	0.00000	0	0.00000	0.00000
4	0	0.00000	0	0.00000	0.00000
5	0	0.00000	0	0.00000	0.00000
6	0	0.00000	0	0.00000	0.00000
7	0	0.00000	0	0.00000	0.00000
8	0	0.00000	0	0.00000	0.00000
9	0	0.00000	0	0.00000	0.00000

Table 64 – Number of children - $N_{Ch}(n)_i$, maximum entropy - $H_{max}(n)_i$, sum of the naked frequencies of the children and children information - $\Delta H(n)_i$ - of the parent structures *n* for the parent/child relationships of type (i): a reactant or product added to a reaction (new internal connection) of the parent. This data is stored in file *ChildrenInf_I.txt*.

Parent Structure *n*	$N_{Ch}(n)_i$	$H_{max}(n)_i$	Sum of the f_N of the Children	$H(n)_i$	$\Delta H(n)_i$
1	0	0.00000	0	0.00000	0.00000
2	0	0.00000	0	0.00000	0.00000
3	0	0.00000	0	0.00000	0.00000
4	0	0.00000	0	0.00000	0.00000
5	0	0.00000	0	0.00000	0.00000
6	0	0.00000	0	0.00000	0.00000
7	0	0.00000	0	0.00000	0.00000
8	0	0.00000	0	0.00000	0.00000
9	0	0.00000	0	0.00000	0.00000

Table 65 – Number of children - $N_{Ch}(n)_{ii}$, maximum entropy - $H_{max}(n)_{ii}$, sum of the naked frequencies of the children and children information - $\Delta H(n)_{ii}$ - of the parent structures *n* for the parent/child relationships of type (ii): a new reaction converting one metabolite into one other added to the parent. This data is stored in file *ChildrenInf_II.txt*.

Parent Structure *n*	$N_{Ch}(n)_{ii}$	$H_{max}(n)_{ii}$	Sum of the f_N of the Children	$H(n)_{ii}$	$\Delta H(n)_{ii}$
1	2	1.00000	7155	0.58690	0.41310
2	0	0.00000	0	0.00000	0.00000
3	0	0.00000	0	0.00000	0.00000
4	0	0.00000	0	0.00000	0.00000
5	0	0.00000	0	0.00000	0.00000
6	0	0.00000	0	0.00000	0.00000
7	0	0.00000	0	0.00000	0.00000
8	0	0.00000	0	0.00000	0.00000
9	0	0.00000	0	0.00000	0.00000

Table 66 – Number of children - $N_{Ch}(n)_{iii}$, maximum entropy - $H_{max}(n)_{iii}$, sum of the naked frequencies of the children and children information - $\Delta H(n)_{iii}$ - of the parent structures *n* for the parent/child relationships of type (iii): the reverse of a reaction added to the parent. This data is stored in file *ChildrenInf_III.txt*.

Parent Structure *n*	$N_{Ch}(n)_{iii}$	$H_{max}(n)_{iii}$	Sum of the f_N of the Children	$H(n)_{iii}$	$\Delta H(n)_{iii}$
1	1	0.00000	0	0.00000	0.00000
2	0	0.00000	0	0.00000	0.00000
3	0	0.00000	0	0.00000	0.00000
4	0	0.00000	0	0.00000	0.00000
5	0	0.00000	0	0.00000	0.00000
6	0	0.00000	0	0.00000	0.00000
7	0	0.00000	0	0.00000	0.00000
8	0	0.00000	0	0.00000	0.00000
9	0	0.00000	0	0.00000	0.00000

Table 67 – Number of children - $N_{Ch}(n)_{iv}$, maximum entropy - $H_{max}(n)_{iv}$, sum of the naked frequencies of the children and children information - $\Delta H(n)_{iv}$ - of the parent structures *n* for the parent/child relationships of type (iv): a new metabolite added to the parent. This data is stored in file *ChildrenInf_IV.txt*.

Parent Structure *n*	$N_{Ch}(n)_{iv}$	$H_{max}(n)_{iv}$	Sum of the f_N of the Children	$H(n)_{iv}$	$\Delta H(n)_{iv}$
1	2	1.00000	3217	0.99480	0.00520
2	0	0.00000	0	0.00000	0.00000
3	0	0.00000	0	0.00000	0.00000
4	0	0.00000	0	0.00000	0.00000
5	0	0.00000	0	0.00000	0.00000
6	0	0.00000	0	0.00000	0.00000
7	0	0.00000	0	0.00000	0.00000
8	0	0.00000	0	0.00000	0.00000
9	0	0.00000	0	0.00000	0.00000

Table 68 – Number of children - $N_{Ch}(n)_v$, maximum entropy - $H_{max}(n)_v$, sum of the naked frequencies of the children and children information - $\Delta H(n)_v$ - of the parent structures *n* for the parent/child relationships of type (v): add to the parent a new reaction connecting a new metabolite to an existing one to form a pair reactant/product. This data is stored in file *ChildrenInf_V.txt*.

Parent Structure *n*	$N_{Ch}(n)_v$	$H_{max}(n)_v$	Sum of the f_N of the Children	$H(n)_v$	$\Delta H(n)_v$
1	3	1.58496	77909	1.47974	0.10522
2	0	0.00000	0	0.00000	0.00000
3	0	0.00000	0	0.00000	0.00000
4	0	0.00000	0	0.00000	0.00000
5	0	0.00000	0	0.00000	0.00000
6	0	0.00000	0	0.00000	0.00000
7	0	0.00000	0	0.00000	0.00000
8	0	0.00000	0	0.00000	0.00000
9	0	0.00000	0	0.00000	0.00000

3.8 – Limits on the Use of TAMES

Before running TAMES, input files must be prepared with the following information: i) topological scope and ii) metabolic network. The topological scope is defined in file *Topological_Scope.txt* by setting the maximum number of generations ($NGen_{max}$) and the maximum number of metabolites ($NMet_{max}$) and reactions ($NReact_{max}$) in the structures. Table 69 shows the range of values accepted by the TAMES for these variables. The recommended values are also shown.

Table 69 – Range of values for the variables used to define the topological scope accepted by TAMES. These values are defined in file *Topological_Scope.txt*.

Topological Scope's Variables	User's Limits		Recommended
	Minimum	***Maximum***	
Metabolites, $NMet_{max}$	2	4	3
Reactions, $NReact_{max}$	1	3	3
Generations, $NGen_{max}$	2	12	9

As mentioned before, the metabolic network is given by the files: *List_Metabolites.txt*, *List_Reactions.txt*, *Products_X_Reactions.txt*, *Reactions_X_Reactants.txt* and *RevReact.txt*. The size of a metabolic network is defined by the number of specific metabolites and reactions. Table 70 shows the size limits of a metabolic network accepted by TAMES.

Table 70 – Size limits of a metabolic network accepted by TAMES.

Number of Specific	User's Limits	
	Minimum	***Maximum***
Metabolites	2	1400
Reactions	1	1400

TAMES also require a minimum of 4 GB of RAM memory for these predefined settings.

4 – Recommendations

The following problems might occur while running the executable program “Instances Counting”:

i. The program stops shortly after being initiated due to insufficient RAM;
ii. The number of instances found exceeds the maximum value set in the code leading to the program to abruptly stop.

The first problem can easily be solved by using a machine with higher RAM. As to the second problem, it simply requires changing to the source code *InstCount_v1.f90*. In this file, the maximum number of instances is given by the variable *maxNumInst*. The variable is set at line 208 to the predefined value of *maxNumInst*=55000000. A higher number must be introduced and the code file must be compiled again. However, the user must be aware that the new value introduced for *maxNumInst* might lead to the first problem, i.e., insufficient RAM. In the worst case, the user should compensate the reduced RAM resources by decreasing the topological scope and the value of *maxNumInst*.

5 – Applicable Licenses

The licenses that apply to TAMES and to this document are mentioned bellow.

5.1 – TAMES

TAMES – Topological Analysis of Metabolic Structures
Copyright (C) 2017
Paulo A. N. Dias, paulo.dias29@gmail.com
Marco Seabra dos Reis, marco@eq.uc.pt
Armindo Salvador, armindo.salvador@gmail.com
Pedro Martins, pmartins@iscac.pt

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 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.

A copy of the GNU General Public License can be found at http://www.eq.uc.pt/~marco/research/pclab/ and http://www.gnu.org/licenses/.

5.2 – Manual

Copyright (C) 2017
Paulo A. N. Dias
Marco Seabra dos Reis
Armindo Salvador
Pedro Martins

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the Annex I, entitled "GNU Free Documentation License". A copy of the license can also be found at http://www.gnu.org/licenses/.

6 – Acknowledgments

The developers of TAMES which to acknowledge the support given by *COMPETE – Programa Operacional Factores de Competividade* and *FCT – Faculdade de Ciências e Tecnologia* through the following projects and grants:

- Research project PTDC/QUI-BIQ/119657/2010
- Grant FCOMP-01-0124-FEDER-020978
- Grant PEst-C/SAU/LA0001/2013-2014

7 – References

1. Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U: **Network Motifs: Simple Building Blocks of Complex Networks**. *Science* 2002, **298**(5594):824-827.
2. Aittokallio T, Schwikowski B: **Graph-based methods for analysing networks in cell biology**. *Brief Bioinform* 2006, **7**(3):243-255.
3. Barabasi AL, Oltvai ZN: **Network biology: Understanding the cell's functional organization**. *Nat Rev Genet* 2004, **5**(2):101-U115.
4. Shellman ER, Burant CF, Schnell S: **Network motifs provide signatures that characterize metabolism**. *Molecular BioSystems* 2013, **9**(3):352-360.
5. Junker BH, Schreiber F: **Analysis of Biological Networks (Wiley Series in Bioinformatics)**: Wiley-Interscience; 2008.
6. Artzy-Randrup Y, Fleishman SJ, Ben-Tal N, Stone L: **Comment on "Network motifs: simple building blocks of complex networks" and "Superfamilies of evolved and designed networks"**. *Science (New York, NY)* 2004, **305**(5687):1107; author reply 1107.
7. Noor E, Eden E, Milo R, Alon U: **Central Carbon Metabolism as a Minimal Biochemical Walk between Precursors for Biomass and Energy**. *Molecular Cell* 2010, **39**(5):809-820.
8. Dias PAN, Reis MSd, Martins P, Salvador A: **Identifying Strong Statistical Bias in the Local Structure of Metabolic Networks: The Metabolic Network of Saccharomyces Cerevisiae as a Test Case**. In: *6th International Conference on Bioinformatics Models, Methods and Algorithms, BIOINFORMATICS 2015: 2015; Lisbon, Portugal*; 2015: 207-212.
9. Forster J, Famili I, Fu P, Palsson BO, Nielsen J: **Genome-scale reconstruction of the Saccharomyces cerevisiae metabolic network**. *Genome Res* 2003, **13**(2):244-253.

Annex I – GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercial. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. Applicability and Definitions

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means

the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. Verbatim Copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. Copying in Quantity

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying

with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.
If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.
If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.
It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.

- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. Combining Documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. Collections of Documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. Aggregation with Independent Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

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, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. Future Revisions of this License

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation 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. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. Relicensing

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for

anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.