



# Tutorial for TARIS version 0.2

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	The purpose of TARIS 0.2 . . . . .	3
1.2	TARIS 0.2 main features . . . . .	4
1.3	XTARIS version 0.1 . . . . .	4
<b>2</b>	<b>Example molecules</b>	<b>4</b>
2.1	Which are the molecules? . . . . .	4
2.2	GAUSSIAN 98 input examples . . . . .	5
<b>3</b>	<b>TARIS Modules</b>	<b>6</b>
3.1	TARIS-BuildIsosurface . . . . .	6
3.1.1	Main purpose . . . . .	6
3.1.2	<b>Syntax</b> . . . . .	6
3.1.3	<b>Example 1:</b> . . . . .	7
3.1.4	<b>Example 2:</b> . . . . .	7
3.2	TARIS-BuildTree . . . . .	7
3.2.1	Main purpose . . . . .	7
3.2.2	<b>Syntax</b> . . . . .	8
3.2.3	<b>Example 1:</b> . . . . .	8
3.2.4	<b>Example 2:</b> . . . . .	9
3.3	TARIS-TreesDistance . . . . .	10
3.3.1	Main purpose . . . . .	10
3.3.2	<b>Syntax</b> . . . . .	11
3.3.3	<b>Example 1:</b> . . . . .	11
3.3.4	<b>Example 2:</b> . . . . .	11
3.4	TARIS-Matrices . . . . .	11
3.4.1	Main purpose . . . . .	11
3.4.2	<b>Syntax</b> . . . . .	11
3.4.3	<b>Example 1:</b> . . . . .	12
3.4.4	<b>Example 2:</b> . . . . .	12
3.5	TARIS-Dendrogram . . . . .	13
3.5.1	Main purpose . . . . .	13
3.5.2	<b>Syntax</b> . . . . .	13
3.5.3	<b>Example 1:</b> . . . . .	14
3.5.4	<b>Example 2:</b> . . . . .	14

# 1 Introduction

## 1.1 The purpose of TARIS 0.2

In few words, TARIS 0.2 (*Tree Analysis and Representation of Isopotential Surfaces*) is a software package designed to compare the Electrostatic Potential (EP) of molecules, avoiding the alignment step required in most of the molecular similarity methods. The main goal of the program is to compute a quantitative measure of similarity and/or dissimilarity among molecules, only regarding the electrostatic potential produced by them, in order to retrieve the desired distance or similarity matrix.

How does the program do this? TARIS 0.2 receives as input information, the electrostatic potential of each molecule in *.cube* format. This is a standard format to encode data about any molecular property when such property has been computed in a 3D grid surrounding the molecule, e. g. the electrostatic potential or the electronic density. For every molecule in the set, for which its corresponding *.cube* file has been provided, TARIS 0.2 performs a discrete potential scan that is determined by three parameters: (i) the initial cutoff (*cutoffBegin*): this is the potential value at which the scan begins; (ii) the final cutoff (*cutoffEnd*): this is the potential value at which the scan ends; (iii) the step size (*stepSize*): a constant quantity in which the potential is incremented until the *cutoffEnd* is reached. In each one of the steps of the scan, TARIS 0.2 computes the Isopotential surface for the corresponding potential value and according to the geometrical and topological changes in such surfaces from one potential to the next, a tree (graph) is built. This tree encodes the information of how the different connected components of the Isopotential surfaces merge with others, and their nodes are weighted by the superficial area of each one of such components. At the end of the scan, TARIS 0.2 has built one tree for each molecule.

It is important to notice that the scan must be carried out in such a way that the surfaces increase their size, therefore the scan must start at the potential that is closest to zero. TARIS 0.2 supports both types of scans (in the negative potentials or in the positive potentials), however, much more chemically significant results have been obtained for the negative potentials.

The next step in the analysis is to compute the tree edit distance between all possible pairs of trees. TARIS 0.2 provides the distance (or dissimilarity) matrix for the set of molecules. But depending on the user's requirements, the similarity matrix can also be computed from the distance matrix. In this matrix the values go from 0% to 100%.

Several statistical tools may be used to extract chemically relevant information from such matrices. One of them is the hierarchical clustering, which has been implemented in TARIS 0.2. When using this technique, the information in the similarity matrix is used to divide the initial set of molecules in several clusters, according to the similarity percentage. The final output of such analysis is a dendrogram that TARIS 0.2 returns in *.png* format. Now, it depends on the user how to apply this information to his own interests.

For a complete description of the method, the algorithm and some applications the reader may see the TARIS paper: Marín, R. M.; Aguirre, N. F.; Daza, E. E, Graph Theoretical Similarity Approach to Compare Molecular Electrostatic Potentials, *J. Chem. Inf. Model.*, **48**, 109–118.

## 1.2 TARIS 0.2 main features

TARIS 0.2 is composed of five command line programs that we call *modules*:

- **TARIS-BuildIsosurface:** Computes the Isopotential Surface for a given cutoff and saves it in oogl format. The .oogl file can be viewed with the program geomview.
- **TARIS-BuildTree:** Performs the potential scan for one molecule and the resulting tree is saved in .gml format.
- **TARIS-TreesDistance:** Computes the edit distance between two molecules. It can receive as input information the .cube files or the previously generated trees saved in .gml format. The distance value is printed in the standard output.
- **TARIS-Matrices:** Computes the distance and/or similarity matrices for a set of molecules. The matrices may be saved as a plain text file.
- **TARIS-Dendrogram:** Takes the similarity matrix produced by TARIS-Matrices and performs a hierarchical clustering analysis. The resulting dendrogram is saved in .png format. This program requires to have the statistical package R.

In the following sections, we will explain in detail the function of each of the five modules and the syntax, and we will also give some examples.

## 1.3 XTARIS version 0.1

This is the graphic users interface for TARIS in which we are working right now. We hope it will be released by June of 2008. The main features of this version are:

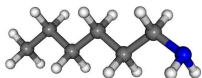
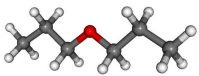
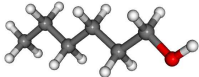
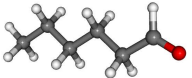
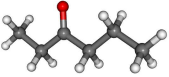
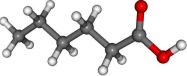
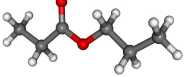
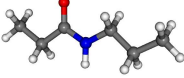
- Allows to load and view several molecules at the same time, before choosing which ones are going to be analyzed.
- Allows to view the desired number of Isopotential Surfaces for any molecule (independently), in order to have an idea of the possible cutoffs for the scan.
- Users can view the resulting tree for the analyzed molecules.
- It allows to analyze the similarity matrix by means of hierarchical clustering and displays the resulting dendrogram.

If you have any questions or comments please visit <http://taris.sourceforge.net> or contact us at [taris.contact@gmail.com](mailto:taris.contact@gmail.com)

## 2 Example molecules

### 2.1 Which are the molecules?

In order to explain each one of the modules with some examples, we compute the .cube files for some molecules. We took the eight organic molecules listed in table 1, for which the geometry optimization was performed at the B3LYP//6-31G(d,p) level. The corresponding Molecular Electrostatic Potential (MEP) cube file was computed with a resolution of  $10^6$  points using GAUSSIAN 98, and these are the files contained in the file `cubesForTutorial.tar.gz` available in the TARIS web site.

Cube file name	Molecule	“Photo”
hexnh.cube	n-hexyl amine	
eter3-3.cube	dipropyl ether	
hexoh.cube	n-hexanol	
hexcoh.cube	hexanal	
cetona2-3.cube	ethyl-propyl ketone	
hexcooh.cube	hexanoic acid	
ester3-3.cube	n-propyl propionate	
amida3-3.cube	N-propyl-propanamide	

**Table 1.** Molecules for which example cube files are provided in the TARIS web site.

## 2.2 GAUSSIAN 98 input examples

In order to show an example of how these cube files may be obtained with GAUSSIAN 98, here we show the input files employed in the case of `ester3-3.cube`.

This is the input for the geometry optimization:

```
-----
%Chk=ester3-3
#P B3LYP/6-31G(d,p) Opt gEOM=(NoDistance,NoAngle)
```

```
ester3-3
```

```
0,1
```

```

C    0.000000 0.000000 0.000000
O    1.056551 0.000000-0.610000
O    0.000000 0.000000 1.400000
C    1.319933 0.000000 1.866667
C   -1.255737 0.000000-0.725000
C    1.319933 0.000000 3.316667
H    1.833292-0.889165 1.503667
H    1.833292 0.889165 1.503667
C    2.687006 0.000000 3.800000
H    0.806573 0.889165 3.679667
H    0.806573-0.889165 3.679667
H    2.687006 0.000000 4.889000
H    3.200365-0.889165 3.437000
H    3.200365 0.889165 3.437000
C   -0.990780 0.000000-2.150587
H   -1.826784-0.889165-0.461917
H   -1.826784 0.889165-0.461917
H   -1.933881 0.000000-2.695086
H   -0.419733 0.889165-2.413669
H   -0.419733-0.889165-2.413669

```

variables:

-----

This is the input to obtain the cube file with the MEP data:

```

-----
%Chk=ester3-3
#P guess(read,only) density(checkpoint) geom(allcheckpoint) cube(100,potential)

ester3-3.cube
-----

```

## 3 TARIS Modules

### 3.1 TARIS-BuildIsosurface

#### 3.1.1 Main purpose

This is the simplest module of TARIS 0.2. It is designed to build an Isopotential Surface for a given cutoff and to save such surface in .oogl format. This oogl file can be viewed with the program geomview (<http://www.geom.uiuc.edu/software/geomview/>). But do not worry, you will be able to visualize these surfaces with XTARIS in few days.

#### 3.1.2 Syntax

```
$ TARIS-BuildIsosurface -c file -b cutoff -o output
```

Required parameters:

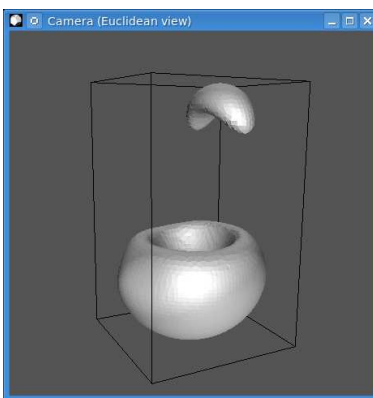
-c **file** This is the name of the cube file containing the electrostatic potential data

Optional parameters:

- b **cutoff** Potential value for the surface calculation potential data (default=-0.07)
- o **output** Name of the oogl output file(default=screen)

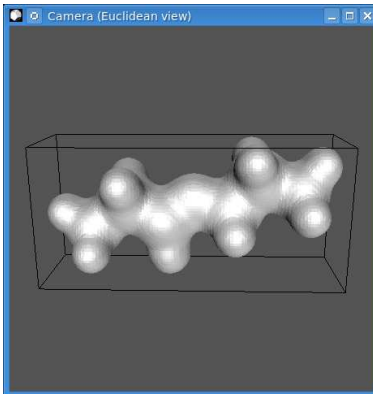
### 3.1.3 Example 1:

```
$ TARIS-BuildIsosurface -c ester3-3.cube -b -0.07 -o ester3-3.oogl  
$ geomview -nopanel ester3-3.oogl
```



### 3.1.4 Example 2:

```
$ TARIS-BuildIsosurface -c ester3-3.cube -b 0.1 -o ester3-3.oogl  
$ geomview -nopanel ester3-3.oogl
```



## 3.2 TARIS-BuildTree

### 3.2.1 Main purpose

This module of TARIS 0.2 builds the tree for one molecule according to the specified parameters for the scan. The output is the tree in .gml format which for the moment cannot be visualized,

but you can check the information printed in the standard output. In such output TARIS 0.2 shows how many connected components are found in each step, it shows the nodes and the potential at which they appear and the connections between nodes, i. e. the edges of the tree. Don't worry, you will be able to visualize the trees with XTARIS in few days.

### 3.2.2 Syntax

```
$ TARIS-BuildTree -c file -b cutoffBegin -e cutoffEnd -s stepSize -o output
```

Required parameters:

-c **file** This is the name of the cube file containing the electrostatic potential data

Optional parameters:

-b **cutoffBegin** Initial cutoff for the scan (default=-0.1)  
 -e **cutoffEnd** Final cutoff for the scan (default=-0.07)  
 -s **stepSize** Step-size for the scan (default=-0.005)  
 -o **output** Name of the gml output file (default=screen)

### 3.2.3 Example 1:

```
$ TARIS-BuildTree -c ester3-3.cube -b -0.2 -e -0.03 -s 0.01 -o ester3-3.gml
```

Standard output:

```
+-----+-----+
| Potential | Number of Components |
+-----+-----+
| -0.20000 | 0 |
| -0.19000 | 0 |
| -0.18000 | 0 |
| -0.17000 | 0 |
| -0.16000 | 0 |
| -0.15000 | 0 |
| -0.14000 | 0 |
| -0.13000 | 0 |
| -0.12000 | 0 |
| -0.11000 | 0 |
| -0.10000 | 0 |
| -0.09000 | 0 |
| -0.08000 | 0 |
| -0.07000 | 1 |
| -0.06000 | 1 |
| -0.05000 | 1 |
| -0.04000 | 2 |
| -0.03000 | 2 |
+-----+-----+
```

## Levels

```
-----
<Potential>    <Nodes>
      inf:      [5]
    -0.03000:  [4] [2]
    -0.04000:  [3]
    -0.07000:  [1]
```

## Connections

```
-----
[1]:: --> [2]
[3]:: --> [4]
[2]:: --> [5]
[4]:: --> [5]
[5]::
```

From this information we can see that one connected component appears first at -0.07 au and that at -0.04 au appears a second one. We also can see the nodes (numbers in square brackets) which represent each connected component and the edges among them (connections).

## 3.2.4 Example 2:

```
$ TARIS-BuildTree -c ester3-3.cube -b -0.2 -e -0.03 -s 0.005 -o ester3-3.gml
```

Standard output:

```
+-----+-----+
| Potential | Number of Components |
+-----+-----+
| -0.20000 | 0 |
| -0.19500 | 0 |
| -0.19000 | 0 |
| -0.18500 | 0 |
| -0.18000 | 0 |
| -0.17500 | 0 |
| -0.17000 | 0 |
| -0.16500 | 0 |
| -0.16000 | 0 |
| -0.15500 | 0 |
| -0.15000 | 0 |
| -0.14500 | 0 |
| -0.14000 | 0 |
| -0.13500 | 0 |
| -0.13000 | 0 |
| -0.12500 | 0 |
| -0.12000 | 0 |
| -0.11500 | 0 |
| -0.11000 | 0 |
```

```

|   -0.10500 |           0 |
|   -0.10000 |           0 |
|   -0.09500 |           0 |
|   -0.09000 |           0 |
|   -0.08500 |           0 |
|   -0.08000 |           0 |
|   -0.07500 |           1 |
|   -0.07000 |           1 |
|   -0.06500 |           1 |
|   -0.06000 |           1 |
|   -0.05500 |           1 |
|   -0.05000 |           1 |
|   -0.04500 |           2 |
|   -0.04000 |           2 |
|   -0.03500 |           2 |
|   -0.03000 |           2 |
+-----+

```

```

-----
Levels
-----

```

```

<Potential>    <Nodes>

      inf:      [5]
-0.03000:     [4] [2]
-0.04500:     [3]
-0.07500:     [1]

```

```

-----
Connections
-----

```

```

[1]:: -->[2]
[3]:: -->[4]
[2]:: -->[5]
[4]:: -->[5]
[5]::

```

In this example we have changed the `stepSize` from 0.01 to 0.005. We can see that the scan is sharper but the number of nodes and the connections among them remains unchanged. There is little dependence on the `stepSize`.

### 3.3 TARIS-TreesDistance

#### 3.3.1 Main purpose

This module of TARIS 0.2 computes the edit distance between two molecules. It receives the gml files previously generated for each molecule. The output data of this module is really simple: the distance value is printed in the standard output. As all modules are command line programs, you may use `TARIS-TreesDistance` to create your own shell script to compute distance matrices as you want, and therefore perform your similarity analysis in a more personalized way. Be aware that the simplicity of this module is what makes it the more versatile one.

### 3.3.2 Syntax

```
$ TARIS-TreesDistance file1 file2
```

Required parameters:

<code>file1</code>	This is the name of the first tree in gml format
<code>file2</code>	This is the name of the second tree in gml format

### 3.3.3 Example 1:

```
$ TARIS-BuildTree -c ester3-3.cube -b -0.2 -e -0.03 -s 0.01 -o ester3-3.gml
$ TARIS-BuildTree -c hexcooh.cube -b -0.2 -e -0.03 -s 0.01 -o hexcooh.gml
$ TARIS-TreesDistance ester3-3.gml hexcooh.gml
```

Standard output:

```
DISTANCE IS: 10.5813
```

### 3.3.4 Example 2:

```
$ TARIS-BuildTree -c hexoh.cube -b -0.2 -e -0.03 -s 0.01 -o hexoh.gml
$ TARIS-TreesDistance ester3-3.gml hexoh.gml
```

Standard output:

```
DISTANCE IS: 26.158
```

In these two examples we can see that the edit distance (dissimilarity) between the ester and the alcohol is bigger than the distance between the ester and the acid. Here we may see how the results from TARIS 0.2 reflect in good manner something that is known from classical chemistry.

## 3.4 TARIS-Matrices

### 3.4.1 Main purpose

This is the module that encompasses the ultimate goal of TARIS 0.2. It generates the distance and/or similarity matrices for a given set of molecules from their cube files. This duality distance/similarity allow you to use the kind of matrix that suits better your preferred technique to analyze these kind of data.

### 3.4.2 Syntax

```
$ TARIS-Matrices -i list -t format -b cutoffBegin -e cutoffEnd -s stepSize
-m type -o output
```

Required parameters:

**-i** **list** This is the name of the file containing the name of the cube/gml files to be analyzed

Optional parameters:

**-t** **format** Specifies the input data type: cube or gml files (default=cube)  
**-b** **cutoffBegin** Initial cutoff for the scan (default=-0.1)  
**-e** **cutoffEnd** Final cutoff for the scan (default=-0.07)  
**-s** **stepSize** Step-size for the scan (default=-0.005)  
**-m** **type** Type of matrix to be generated: distance, similarity or both (default=both)  
**-o** **output** Name of the output file (default=screen)

### 3.4.3 Example 1:

Given the following **list** file:

```
amida3-3.cube
cetona2-3.cube
ester3-3.cube
eter3-3.cube
hexcoh.cube
hexcooh.cube
hexnh.cube
hexoh.cube
```

type:

```
$ TARIS-Matrices -i list -t cube -b -0.2 -e -0.04 -s 0.005
-m distance -o distance_matrix.dat
```

The output file **distance\_matrix.dat** is:

```
DISTANCE MATRIX 'D'
-----
    amida3-3 :      0.000000  17.010949  28.930687  39.773215  19.313279  32.194470  38.509574  33.099564
  cetona2-3 :      17.010949   0.000000  12.868808  23.711335   3.884025  16.132591  22.447695  17.037685
  ester3-3 :      28.930687  12.868808   0.000000  18.272858  12.267582   4.735348  17.009218  13.621510
  eter3-3 :      39.773215  23.711335  18.272858   0.000000  20.459935  14.399086   1.547687   6.673650
   hexcoh :      19.313279   3.884025  12.267582  20.459935   0.000000  12.881191  19.196295  13.786285
  hexcooh :      32.194470  16.132591   4.735348  14.399086  12.881191   0.000000  13.135446   9.188332
   hexnh :      38.509574  22.447695  17.009218   1.547687  19.196295  13.135446   0.000000   5.410010
   hexoh :      33.099564  17.037685  13.621510   6.673650  13.786285   9.188332   5.410010   0.000000
```

THE MAXIMUM DISTANCE IS: 39.773215

### 3.4.4 Example 2:

For the same **list** file:

```
$ TARIS-Matrices -i list -t cube -b -0.2 -e -0.04 -s 0.005
-m similarity -o similarity_matrix.dat
```

The output file `similarity_matrix.dat` is:

SIMILARITY PERCENTAGE MATRIX 'S'

```
-----
amida3-3 :      100.000000   57.230139   27.260878    0.000000   51.441493   19.054895    3.177115   16.779258
cetona2-3 :      57.230139   100.000000   67.644537   40.383659   90.234571   59.438554   43.560773   57.162917
ester3-3 :      27.260878   67.644537   100.000000   54.057376   69.156171   88.094128   57.234491   65.752051
eter3-3 :        0.000000   40.383659   54.057376   100.000000   48.558507   63.797027   96.108719   83.220742
hexcoh :        51.441493   90.234571   69.156171   48.558507   100.000000   67.613403   51.735622   65.337766
hexcooh :       19.054895   59.438554   88.094128   63.797027   67.613403   100.000000   66.974141   76.898191
hexnh :         3.177115   43.560773   57.234491   96.108719   51.735622   66.974141   100.000000   86.397856
hexoh :        16.779258   57.162917   65.752051   83.220742   65.337766   76.898191   86.397856   100.000000
```

## 3.5 TARIS-Dendrogram

### 3.5.1 Main purpose

This is an special module in TARIS 0.2, that performs additional statistical analysis, from the similarity matrix provided by TARIS-Matrices. The type of analysis is called *hierarchical clustering* and its only output data is a dendrogram. You may choose among several linkage methods such as ward, single, complete, average, mcquitty, median or centroid before performing the analysis. In order to use this module you must install the statistical package R (<http://www.r-project.org>) along with the ade4 package: (<http://cran.r-project.org/web/packages/ade4/index.html>).

Complete instructions for their installation are in <http://taris.sourceforge.net>.

### 3.5.2 Syntax

```
$ TARIS-Dendrogram -i file -o output -m method -w width -h height
```

Required parameters:

`-i` **file** This is the name of the file containing the similarity matrix produced by TARIS-Matrices

Optional parameters:

`-o` **output** Name of the output dendrogram in .png format (default=output.png)

`-m` **type** The clustering method to be used for the dendrogram construction. The available clustering methods are: ward, single, complete, average, mcquitty, median or centroid (default = average)

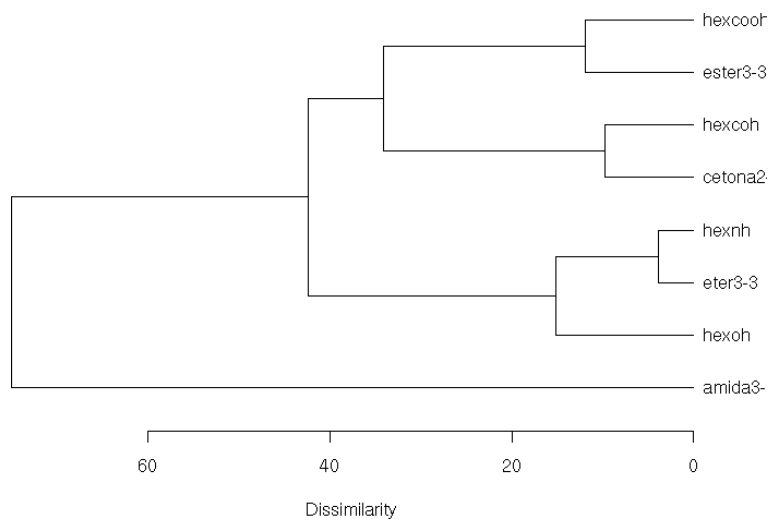
`-w` **width** The ouput file image width (default = 800)

`-h` **height** The ouput file image height (default = 600)

### 3.5.3 Example 1:

```
$ TARIS-Dendrogram -i similarity_matrix.dat -o dendrogram.png -m average
```

The output file `dendrogram.png` is:



### 3.5.4 Example 2:

```
$ TARIS-Dendrogram -i similarity_matrix.dat -o dendrogram.png -m single
```

The output file `dendrogram.png` is:

