Tutorial on Generative Topographic Mapping

G. MARCOU, D. HORVATH, O. KLIMCHUK, F. BONACHERA and A. VARNEK

1. Introduction

The tutorial aims at presenting the Generative Topographic Mapping (GTM) algorithm[1]. The GTM is an unsupervised method to map high dimensional data to a two-dimensional representation. In the process, the GTM builds a probabilistic model of the data that can be exploited for data characterization, comparison or classification and regression model building. The GTM approach will be used to analyze a dataset of flavors and to explore structure-flavor relationships. It will be the occasion to get some deeper insight into the method with a particular focus on the effects of the GTM parameterization on the obtained map.

1.1. Software and Files

The tutorial is based on three pieces of software:
`xGTMapTool`: a graphical user interface frontend for the preparation of a GTM.
`xGTMview`: an application to link the GTM trained on chemical data and the chemical structures.
`xGTMmanifold`: an application illustrating the concept of GTM manifold and data space.

The directory FDB contains the following files:

- train.sdf and test.sdf: the chemical structure annotated with flavor descriptions, separated into a training and a test set.
- `FLAVOR_DB_OK.sdf`: This file groups the training and test structural data for convenience.
- `train.svm` and `test.svm`: the ISIDA IIAB(2-5) fragment descriptors of the corresponding chemical structures. These data are also provided in arff format.
- `train.hdr` and `test.hdr`: the labels of ISIDA IIAB(2-5) fragment descriptors.
- `train_Freq_01.svm` and `test_Freq_01.svm`: the "essential" ISIDA IIAB(2-5) fragment descriptors of the corresponding chemical structures, monitoring only the 90% more frequent fragments (the descriptor vector elements corresponding to "exotic" fragments appearing in at most 10% of the structures are now discarded). These data are also provided in arff format.
- `train_Freq_01.hdr` and `test_Freq_01.hdr`: the labels of the "essential" ISIDA IIAB(2-5) fragment descriptors.

The directories `Exo1`, `Exo2`, `Exo3`, `Exo4` and `Exo5` contain examples files obtained during the tutorial.

1.2. Licence

The software are licensed by the University of Strasbourg. The license file is called licence.dat and is situated in the OS specific directories: Windows, Mac and Linux. The licence file must be installed in a proper location to be found.

- *On Windows*: create the directory AppData\local\ISIDAGTM2018 directory at the root of your home directory and copy the file license.dat in it. The absolute path of the file should be similar to this one:

  `C:\Users\username\AppData\local\ISIDAGTM2018\licence.dat`

  The file and the directory should have read and write permissions.

- *On Mac*: create the directory .config/ISIDAGTM2018 directory at the root of your home directory and copy the file license.dat in it. The absolute path of the file should be similar to this one:

  `/Users/username/.config/ISIDAGTM2018/licence.dat`

- *On Linux*: create the directory .config/ISIDAGTM2018 directory at the root of your home directory and copy the file license.dat in it. The absolute path of the file should be similar to this one:

  `/home/username/.config/ISIDAGTM2018/licence.dat`

1.3. The flavor dataset

The tutorial uses a dataset of organoleptic compounds mined (in February 2018) from the FlavorDB database[2]. The database is aggregating information from many existing sources: FooDB[3], BitterDB[4], SuperSweet[5], SuperScent[6], FlavorNet[7], Fenaroli's Handbook of Flavor Ingredients[8] among others.

The *sweet-like* annotation from the SuperSweet database, was removed because, as stated by the authors[5], "the sweet tasting molecules were extracted from the literature and publicly available databases like Pubchem, the PDB and MonoSaccharideDB and were filtered using different terms like 'sweetening agents'. In the next step the data set was extended by using similarity search methods." The *sweet-like* annotation refers to those compounds found by similarity and therefore their sweetening properties are only presumed.

A second problem with the SuperSweet database is the lack of information to support the labels. Most entries do not identify a source in support to the categorization of a substance as sweet. For this reason, an additional source of information on sweetening agents was incorporated: the dataset used by Todeschini et al[9]. Then all references originating from the SuperSweet database were removed unless they were confirmed by a second source.

This initial dataset followed a standardization process. Entries featuring stereoisomers were merged because the molecular descriptors used in this tutorial do not distinguish stereoisomers. Finally, after standardization of the chemical structures, duplicate structures were merged. The merging concerned all fields associated to chemical structures: the flavor descriptions, bibliographic sources, etc.

There are 70 mixtures in total in the dataset. These are substances with a flavor profile that are described as the constituents of the mixture as, for instance, bretylium tosylate. There are also ionic substances such as sodium chloride. Ionic forms were kept when the nature of one ion did change the perception of the substance. For instance, sulfate is sour, magnesium sulfate is bitter, iron sulfate is metallic and ammonium sulfate is astringent. However, if a

compound appeared as a component of several salts but the organoleptic description did not change with the counter-ion, then all the entries were merged and the counter-ions were deleted from the structure.

Then, the dataset was reviewed manually to disambiguate some entries when possible. For instance, the structures of santalene, santalol and santalyl acetate had to be homogenized over the different sources providing their structure. Some entries remain suspicious although there is no clear evidence that their chemical structure are wrong. This is the case of oxo-carboxylic acids glycerides, because they can be easily confused with precursors of triglycerides. Some entries were discarded because they were described as polymers, such as cellulose and the chemical structure could be properly rendered by the monomer represented in the entries.

Overall, the dataset contains 3438 substances, which is a substantial reduction from the 25595 entries in the FooDB database. Most of the removed compounds were filtered out following the filters based on the SuperSweet database. The dataset size, finally, is smaller but of the same order of magnitude to the less accessible but renown Lefingwell database[10].

The flavor labels also require a substantial amount of attention. Some of the flavors are described using natural language using qualifiers such as “weak” or “very strong”. Although important for the precision of the description they weaken the statistical analysis because they are much rarer than the noun they are qualifying. For instance “very sweet” is rare compared to “sweet” and we preferred to requalify these compounds as “sweet” rather then create a new and ill-defined category. The same logic applied to adjectives based on a noun or a noun used as an adjective. In such situation as “grassy” and “grass” the term “grassy” was preferred and all concerned items were grouped under the label “grass”. The noun was sometime preferred because it seemed more used, for instance “sweet” was preferred over “sweety”. In total, this required 132 rule. This procedure is rather conservative, by contrast to more systematical natural language processing analysis[11].

The files `FLAVOR_DB_OK.sdf`, `train.sdf` and `test.sdf` contain chemical structures and annotations of the dataset. The available fields are the following:

- `REFERENCES`: a string consisting of URLs to the data repository where the compounds and labels originated from
- `FLAVOR PROFILE`: the description of the flavor of the compound using a dictionary of 566 terms
- `FLAVOR-TERM`: if a flavor term is present in the flavor profile, this information is repeated as separate SDF fields. It is more easy to analyze the flavor profile in this format.

The file `FLAVOR_DB_OK.sdf` is split into equal sized training and test sets. ISIDA Molecular Fragment Descriptors of type IIAB(2-5) were computed on these datasets. The descriptor set was limited to the 99% most frequent fragments, in order to increase the robustness of models and the speed of the calculations for the tutorial.

The training and test sets are stored in the format LibSVM and ARFF, ready to be processed with machine learning tools.

1.4. The Generative Topographic Mapping algorithm

The GTM consists into fitting a finite 2D surface, termed the "manifold", onto a dataset embedded in a high dimensional space, the input space (IS), defined by the molecular descriptor vector. The surface is described by a Generalized Linear Regression (GLR), using as basis functions, a set of *m* Radial Basis Functions (RBF) of width *w* homogeneously distributed over the surface. The surface is the center of a normal probability distribution, with a predefined set of *k* locations of the manifold. These ones are called the nodes of the GTM. The resulting distributions are used to compute a probability for each element of the IS. It is therefore possible to estimate the probability of the dataset (the so-called likelihood) considering a particular geometry of the surface in the IS. The GLR is used to optimize the likelihood under the constraint of a regulation term, of intensity controlled by a parameter *l*. As a result, the contribution of each node to the likelihood of a compound can be computed. This quantity is termed the responsibility. Therefore, a compound *n* appears on the GTM as pattern of responsibilities R_{nk}, representing its relative degrees of association, or "residence" within every node *k*. It is common practice to compute an average position on the map based on the responsibilities of a compound. The corresponding (x,y) position is termed the projection of the compound on the map. Responsibilities are a key ingredient: they are used to locate instances on the map, represent the density of the chemical space, or build SAR and QSAR models.

➢ Pre-processing of the descriptors

Since GTM manifold construction is a non-linear process, its outcome is sensitive to the numerical ranges covered by each descriptor element. It may be helpful to therefore make sure that all descriptor elements undergo specific rescaling/recentering in order to fit into a same final range of values. The most common pre-processing steps of the molecular descriptor sets are supported by the GTM software. The first option, of course, is to not use any pre-processing. In that case, the molecular descriptors are not transformed.
The other options are the following, considering the value x_j^i of the *j*th molecular descriptor of the molecule *i*:

- *Standardize*: the average value m_j and the standard deviation s_j of the molecular descriptor *j* are estimated, then the standardize value is $x_{std,j}^i = (x_j^i - m_j)/s_j$.
- *Center*: the average m_j value is removed from the descriptor value: $x_{ctr,j}^i = x_j^i - m_j$
- *Normalize*: the molecular descriptors are confined in the range [-1,1], using $x_{min,j}^i$ and $x_{max,j}^i$, the min and max values of the descriptor: $x_{nrm,j}^i = \frac{x_j^i - (x_{max,j}^i + x_{min,j}^i)/2}{(x_{max,j}^i - x_{min,j}^i)/2}$
- *Normalize and center*: the molecular descriptors are confined in the range [-1,1] then they are centered. As a result, a descriptor element is no longer confined in the range [-1,1], but the range of value still covers 2 units. Using the same notations, the modified value of the descriptor is: $x_{ctn,j}^i = \frac{x_j^i - m_j}{(x_{max,j}^i - x_{min,j}^i)/2}$

All these transformations follow the general formula: $x_{tranformed,j}^i = \frac{x_j^i - M_j}{S_j}$. The data are shifted by a constant M_j and scaled by another constant S_j.

2. Step by step instructions

The exercises are developed as an introduction to the GTM approach. They start with the generation of a GTM (Exercise 1) and using it on new data (Exercise 2). Then the results are visualized (Exercise 3). In the next step, the convergence of the algorithm (Exercise 4) and the parameterization of the GTM are scrutinized (Exercise 5).

2.1. Exercise 1. Train a GTM.

Instructions	***Comments***
Open the **`xGTMapTool`** software	The interface of the software appears (Figure 1).
Click the button to the right of the **`Input`** label (Figure 1, area 1) and select the file `train_Freq_0.1.svm`.	This is the selection of the datafile used to train the GTM model. An automatically generated output base name is proposed by the soft unless explicitly set up by the user. The output base name will be used to name all the files produced by the software. All those files will be in the path specified in this field. The generated files will differ by their terminations only.
As a preprocessing option (Figure 1, area 2), use the **`standardize`** option.	An important aspect of the training of the GTM model is the pre-processing. The initial state of the manifold is a flat surface fitted to the two first principal component of the dataset. Therefore, the dataset must be centered. Furthermore, if there are some large differences in variance between the descriptors, this will bias the manifold toward the ones covering a wider numeric range. A reasonable choice to avoid these pitfalls is to standardize the dataset.
Set the **`Number of traits`** value to 9 (Figure 1, area 3) then click on the button **`OK`** (Figure 1, area 6).	The other parameters of the method are set to default values. These values are visible in the log window (Figure 1, area 5 and Figure 2). The width of the RBFs are set to two times the distance between two neighboring RBF on the latent space plane. The number of node is 25 times the number of traits and the regularization parameter is set to 1. While the calculations are running, the log window displays information (Figure 3) about the current state of the process: • a warning in case previous results are affected by the current run;

	• a reminder about key parameters setup; • the number of instances to process; • a first guess of the likelihood of the dataset. At each step, the log line gives: • the expectation-maximization iteration count; • the current value of the likelihood; • the variation of likelihood since the previous step; • the percentage of variation of the log likelihood compared to the present value of the log likelihood; • the largest variation of a value in the weight matrix defining the manifold; • the same number as a percentage. At the end of the calculations a message (Figure 4) informs that the process terminated successfully and the last iteration is informative about the log likelihood of the studied dataset.
Edit the file `train_Freq_01.xml`.	The process generated an XML file containing the GTM model.

The GTM model is stored as an XML file, based on the following tags.

- **`GTM`**, it is the main node of the XML model file. It supports the attributes
  - **D**, specifying the dimensionality of the input space (*ie* the number of molecular descriptors),
  - **N** is the number of instances used to train the GTM,
  - **`Type`** indicates which particular GTM algorithm is used,
  - **`nIter`** is the number of training iterations,
  - **`Preprocess`** indicating which kind of preprocessing was used.
- **`Mean`**, is the shift value on each molecular descriptor. It is the actual mean of the molecular descriptors if the preprocessing is a Standardization.
- **SD**, is the scaling value on each molecular descriptor. It is the actual standard deviation of the molecular descriptors if the preprocessing is a Standardization.
- **`PC123`**, are the coordinates of the approximated first three principal components of the dataset.
- **`Manifold`**, contains the values of the weight matrix defining the manifold. It needs the following attributes:
  - **D**, the dimension of the input space;
  - **K**, the number of nodes;
  - **M**, the number of RBFs;
  - **`sigma`**, the width of the RBFs;
  - **`alpha`**, the value of the regularization parameter;
  - **`beta`**, the standard deviation of the normal distribution around the manifold.

Therefore, this node is the core of the GTM model.

- **`LatentSamples`**, the 2D coordinates of the nodes on the latent space.
- **`LatentTraits`**, the 2D coordinates of the RBFs on the latent space.

Conclusion

In this exercise, the training set file `train_Freq_01.svm` is used to train a GTM model using mostly default parameter values. The resulting model is stored as an XML file. The training algorithm is an expectation-maximization, that can be assimilated to a gradient descent. Therefore, the likelihood shall evolve in a monotonic manner, here it is increasing at each step up to convergence. The likelihood itself is supposed to be a negative value. In some cases the value can be observed positive, but it is usually pathological and indicates that something wrong is happening. Generally, it is due to an unwise choice of the pre-processing.

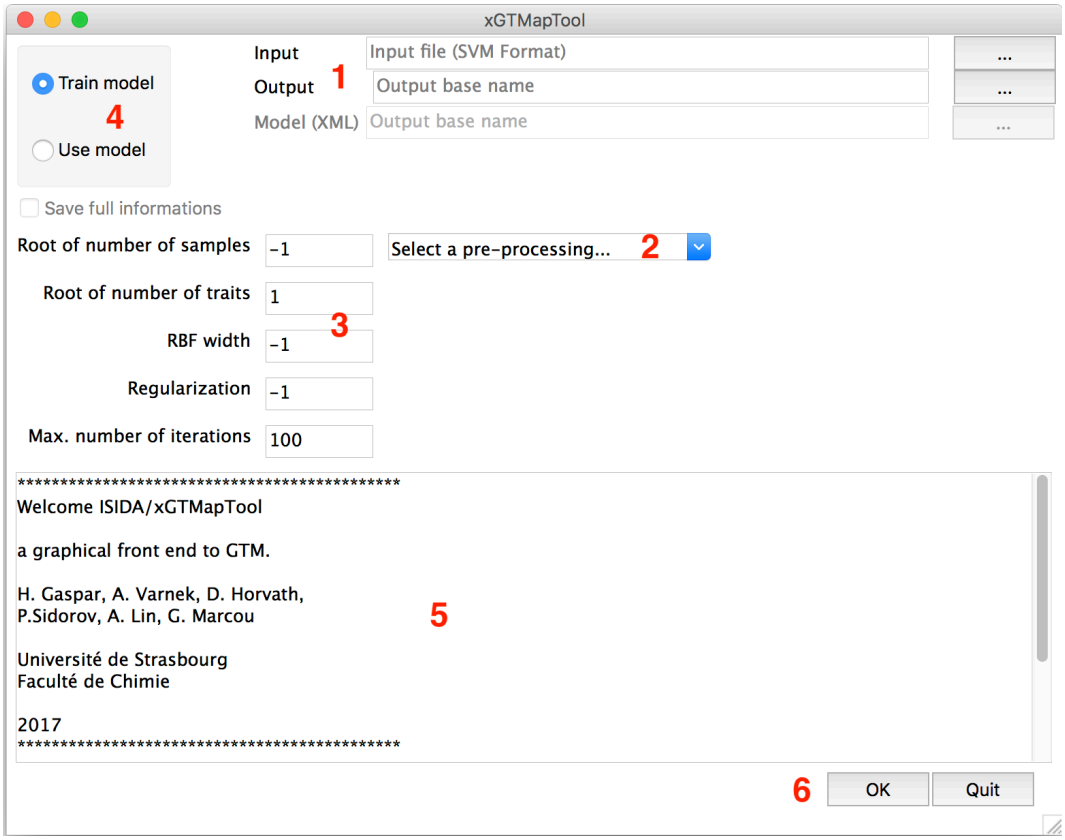

Figure 1. The interface of the xGTMapTool application. The file management is operated in the region (1) of the interface. The preprocessing is taken care of in (2) and the parameterization of the model is performed in (3). The use of the interface to train or apply a GTM model is controlled in (4). The log of the calculations are written in (5) and launching the calculations is performed in (6).

```
************************************************************
**************************YOUR OPTIONS**************************
************************************************************

************************************************************
Classical GTM
**********INPUT AND OUTPUT PATHS**********
Input file: /Users/marcou/Documents/CS3-2018/FDB2/CVIter1Fold1/t9l2u5/train_Freq_01.svm
Output file: /Users/marcou/Documents/CS3-2018/FDB2/CVIter1Fold1/t9l2u5/train_Freq_01
******************************************

********BASIC ALGORITHM PARAMETERS**********
Width of rbf: 1.33333333333333
Number of traits of the latent probability distribution (e.g. rbf centers): 9
Number of samples of the probability distribution: 225
Maximum number of iterations: 100
Convergence precision: +/- 0.001
******************************************
********NORMAL ALGORITHM PARAMETERS*********
Regularization coefficient: 1
Input attributes are standardized.
******************************************
```

Figure 2. Default parameter values when the only user setting is the number of traits equal to 9.

```
WARNING: File /Users/marcou/Documents/CS3-2018/FDB2/CVIter1Fold1/t9l2u5/train_Freq_01R.svm is deleted
WARNING: File /Users/marcou/Documents/CS3-2018/FDB2/CVIter1Fold1/t9l2u5/train_Freq_01Prj.mat is deleted
*****************Reminder*****************
Width of rbf: 1.33333333333333
Regularization coefficient: 1
******************************************
Number of training instances: 1719
First LLt=-169.933773986627
Iter.: 1 LLmap=-115.98053
Iter.: 2 LLmap=-109.66583 DLLmap=6.31470 %DLLmap=5.44462 DW=4.19294 %DW=0.54810
Iter.: 3 LLmap=-107.48498 DLLmap=2.18085 %DLLmap=1.98863 DW=3.41660 %DW=0.44661
Iter.: 4 LLmap=-106.74759 DLLmap=0.73739 %DLLmap=0.68604 DW=1.72880 %DW=0.22599
Iter.: 5 LLmap=-106.36303 DLLmap=0.38456 %DLLmap=0.36026 DW=1.09941 %DW=0.14371
Iter.: 6 LLmap=-106.09003 DLLmap=0.27300 %DLLmap=0.25667 DW=0.94138 %DW=0.12306
Iter.: 7 LLmap=-105.87716 DLLmap=0.21287 %DLLmap=0.20065 DW=0.83908 %DW=0.10968
Iter.: 8 LLmap=-105.68304 DLLmap=0.19412 %DLLmap=0.18334 DW=0.79294 %DW=0.10365
Iter.: 9 LLmap=-105.43344 DLLmap=0.24960 %DLLmap=0.23618 DW=1.25572 %DW=0.16415
Iter.: 10 LLmap=-105.13051 DLLmap=0.30293 %DLLmap=0.28731 DW=1.49925 %DW=0.19598
```

Figure 3. State messages during the GTM model training. It starts with warning in case previous results are affected by the current run, reminders about key parameters setup, reviewing the number of instances to process and a first guess of the likelihood of the dataset. Then at each step, the line give the step count, the current value of the likelihood, the variation of likelihood since the previous step, the same number as a percentage, the largest variation of the weight matrix defining the manifold and the same number as a percentage.

```
Iter.: 66 LLmap=-103.47416 DLLmap=0.00104 %DLLmap=0.00100 DW=0.03754 %DW=0.00491
Iter.: 67 LLmap=-103.47320 DLLmap=0.00096 %DLLmap=0.00093 DW=0.03594 %DW=0.00470
***All calculations finished successfully!***
```

Figure 4. Last iteration of the training of the GTM.

2.2. Exercise 2. Apply the GTM model

Instructions	***Comments***
Use the **xGTMapTool** interface. Reopen it if it was closed. Then chose the **use model** option (Figure 1, area 4).	In this mode, parameters of the GTM algorithm are no longer available. Simultaneously, the interface to select a GTM model becomes available. Indeed, the

	parameters of the GTM are included into the model definition.
Set up the input for the training set (Figure 1, area 1). • Choose as **input** the file `train_Freq_01.svm`. • Choose as **Model (XML)** the file `train_Freq_01.xml`. • Check if the **Save full information** box is not ticked. Untick if needed. • Click the **OK** button.	The log file of the calculation contains the parameter values of the GTM and an and estimates the likelihood of the dataset to -103.47, which is the same obtained at the end of the training stage (Figure 5). During this procedure the training set is projected on the GTM. The software will generate two files: `train_Freq_01R.svm` and `train_Freq_01Prj.mat`. The `train_Freq_01R.svm` file contains the responsibilities computed for each molecule in the libsvm format. The first column is the likelihood of each compound Then, each pair of column separated values represent first the identifier of a node on the map and the responsibility of this node to the molecule. The `train_Freq_01Prj.mat` file is a two column file containing the (x,y) coordinates of the projections of the molecules on the manifold. These coordinates are weighted average of the coordinates of each node of the GTM with the associated responsibilities.
Set up the input for the test set (Figure 1, area 1). • Choose as **input** the file `test_Freq_01.svm`. • Choose as **Model (XML)** the file `train_Freq_01.xml`. • Untick the **Save full information** box if needed. • Click the **OK** button.	During this process, the test set is projected on the GTM manifold. The likelihood of this test set is estimated to -104.08 (Figure 6). The value is smaller than the training set likelihood: the test set is a bit less well explained by the GTM model than the training set. This is a classical situation with machine learning methods. The software produces two new files: a responsibility file (`test_Freq_01R.mat`) and a projection file (`test_Freq_01Prj.mat`), as in the previous step.

Conclusion

In this exercise, the previously build GTM model was used to project data on it. Two categories of information are reported. First, the files named using the scheme <base name>R.svm contains the likelihood of each compound and the responsibility of each node for each compound. Second, the files named using the scheme <base name>Prj.svm report the projections of each molecules on the map. Usually, new data are less explained than the data

used to train the GTM. This is expected and if the likelihood differences between training and test data increases, it can be symptomatic of overfitting situations.

```
************************************************************
**************************YOUR OPTIONS**************************
************************************************************

************************************************************
Load model file: /Users/marcou/Documents/CS3-2018/FDB2/CVIter1Fold1/t9l2u5/train_Freq_01.xml
Classical GTM
***********INPUT AND OUTPUT PATHS***********
Input file:
Output file: /Users/marcou/Documents/CS3-2018/FDB2/CVIter1Fold1/t9l2u5/train_Freq_01
Projection of data from file: /Users/marcou/Documents/CS3-2018/FDB2/CVIter1Fold1/t9l2u5/train_Freq_01.svm
******************************************

********BASIC ALGORITHM PARAMETERS**********
Width of rbf: 1.33333
Number of traits of the latent probability distribution (e.g. rbf centers): 9
Number of samples of the probability distribution: 225
Maximum number of iterations: 100
Convergence precision: +/- 0.001
******************************************
********NORMAL ALGORITHM PARAMETERS*********
Regularization coefficient: 1
Input attributes are standardized.
******************************************

************************************************************
**********************BEGIN COMPUTATIONS**********************
************************************************************


WARNING: File /Users/marcou/Documents/CS3-2018/FDB2/CVIter1Fold1/t9l2u5/train_Freq_01R.svm is deleted
WARNING: File /Users/marcou/Documents/CS3-2018/FDB2/CVIter1Fold1/t9l2u5/train_Freq_01Prj.mat is deleted
Likelyhood of projected data: -103.47322
***All calculations finished successfully!***
```

Figure 5. Log of the mapping of the training set on the GTM model.

```
************************************************************
*************************YOUR OPTIONS*************************
************************************************************

************************************************************
Load model file: /Users/marcou/Documents/CS3-2018/FDB2/CVIter1Fold1/t9l2u5/train_Freq_01.xml
Classical GTM
***********INPUT AND OUTPUT PATHS***********
Input file:
Output file: /Users/marcou/Documents/CS3-2018/FDB2/CVIter1Fold1/t9l2u5/test_Freq_01
Projection of data from file: /Users/marcou/Documents/CS3-2018/FDB2/CVIter1Fold1/t9l2u5/test_Freq_01.svm
****************************************

********BASIC ALGORITHM PARAMETERS**********
Width of rbf: 1.33333
Number of traits of the latent probability distribution (e.g. rbf centers): 9
Number of samples of the probability distribution: 225
Maximum number of iterations: 100
Convergence precision: +/- 0.001
****************************************
********NORMAL ALGORITHM PARAMETERS*********
Regularization coefficient: 1
Input attributes are standardized.
****************************************

************************************************************
**********************BEGIN COMPUTATIONS**********************
************************************************************

WARNING: File /Users/marcou/Documents/CS3-2018/FDB2/CVIter1Fold1/t9l2u5/test_Freq_01R.svm is deleted
WARNING: File /Users/marcou/Documents/CS3-2018/FDB2/CVIter1Fold1/t9l2u5/test_Freq_01Prj.mat is deleted
Likelyhood of projected data: -104.08266
***All calculations finished successfully!***
```

Figure 6. Log of the mapping of the test set on the GTM model.

2.3. Exercise 3. Visualize the projected data

Instructions	***Comments***
Open the application **`xGTMView`**	The interface should look as illustrated in the Figure 7. The software aims at connecting the chemical content of the GTM with some plots of the GTM itself. Input is managed in (1). Navigation of the chemical structure file is performed using the controls in (2) and chemical structures are displayed in (5). The GTM data are plotted in (3) and the content of the plots are controlled in (4). The log are written in (6). The plot processing is launched in (7).
Setup the input files to process (Figure 7, area 1). • Click the **`GTM Model (XML format)`** button and chose the file `train_Freq_01.xml`. • If needed, click the **`Projection coordinates (MAT format)`**	At this step, the GTM model file is processed. The information about how the training/test data set are projected on the map is contained in the responsibility files generated during the previous exercise. When the **`GTM Model (XML format)`** interface is setup, the software will guess if

button and chose the file `train_Freq_01Prj.mat`. • Check also that the corresponding `train_Freq_01R.svm` file is selected as the **Responsibility file (SVM format)**. Otherwise click the corresponding button to choose this file. • Open the file chooser dialog of the **Molecular structure file (SDF format)** to locate and select the file `train.sdf`. • Click the **OK** button.	there exist some relevant projection and responsibility files. In the current situation, we will focus on the projection of the training data. The file `train.sdf` is connected to these data. The order of the molecules in these different files is assumed to be the same. In other words, molecules must appear in the SDF file in the same order as in the molecular descriptor file projected on the GTM. In turn, the GTM output will preserve the same order. In case of discrepancies between the files, the results might be meaningless and eventually, the application may crash.
Tick the **Traits** box (Figure 7, area 4).	The plot (Figure 7, area 3) displays the localization of the RBF on the latent space (Figure 8). The term trait is often used in the GTM literature, but in the context of these exercises it is a synonym for the RBFs. Here, the RBFs are shaping the manifold: the more they are, the more flexible it is. In other types of GTM algorithm, a trait will term other degrees of freedom of the model. Another observation is that the RBFs are distributed in a pseudo-regular way. This allows to compute GTM with an arbitrary number of traits.
• Untick the **Traits** box • Tick the **Samples** box	This configuration plots the positions of the nodes of the GTM (Figure 9). The nodes are also termed samples because they are the points of the manifold on which the probability density is estimated. In a sense, they are sampling the density. They are also distributed in a pseudo-regular way, which might not coincide with any RBF center. For each compound, the responsibility of every node is computed. Therefore, these responsibilities can be summed up on the nodes. The larger is the sum, the denser is chemical space described by this node. This is represented by the size of the circles representing each node: the larger is a circle, the more populate is the corresponding region of the chemical space.

• Untick the **Samples** box • Tick the **Projections** box • In the area 2, select from the list of available SDF fields, the `sweet` key. • *Optional*: if the plotted points are too small, you can use the slide bar at the bottom right hand corner of the plotting area and validate with the **OK** button.	The plot represents the location of each molecule on the map (Figure 10). The map itself is interactive. First, the points are colored according to the values of the SDF fields. Thus, selecting the field "sweet" in the area (2) of the interface, compounds described as having a sweet taste are indicated as black dots. When browsing molecules, their location is highlighted by a blue dot. When clicking on a dot, it is highlighted and the chemical structure is drawn in (5). It is then easy to notice a large "sweet way" across the chemical space and to notice that they are carbohydrates of increasing complexity.
• Go to the compound 118 using the SDF navigation bar (Figure 7, area 2). • Untick the **Projection** box • Tick the **Responsibility** box	From time to time, a compound can appear dissimilar to its neighbors. One explanation can be found by deeper looking into the responsibility pattern of the compound (see for instance the responsibility pattern of the molecule 118, Figure 11). In fact, the compound is located on the map at the "center of mass" of its responsibility pattern. Most compounds are mono-modal: they almost exclusively reside in a single node, and their (x,y) projection will match the node coordinates. But some compounds are delocalized over several nodes. This means that the compound shares some structural characteristics with different chemotypes in the dataset. From the point of view of the dataset those compounds are some kind of chimera.
Load the `test_Freq_01Prj.mat` file as the **Projection coordinates**. Load the `test_Freq_01R.svm` file if needed as the **Responsibility file**. Load the `test.sdf` file as **Molecular structure file**. Click the **OK** button.	During this step, the test set projection is loaded in the interface. As previously mentioned, the software expects the chemical structures, the projection and responsibility files to follow the same order. The same analysis can be repeated. But the main observation is that the organization of the chemical space differs very little considering the training data and the test data. This is expected if the model is not too overfitted.

Conclusion

This exercise, illustrates the analysis of the GTM model and its application to the training and to the test data. It illustrated the key concepts of the GTM model: the traits, the nodes, the responsibilities, the projection.

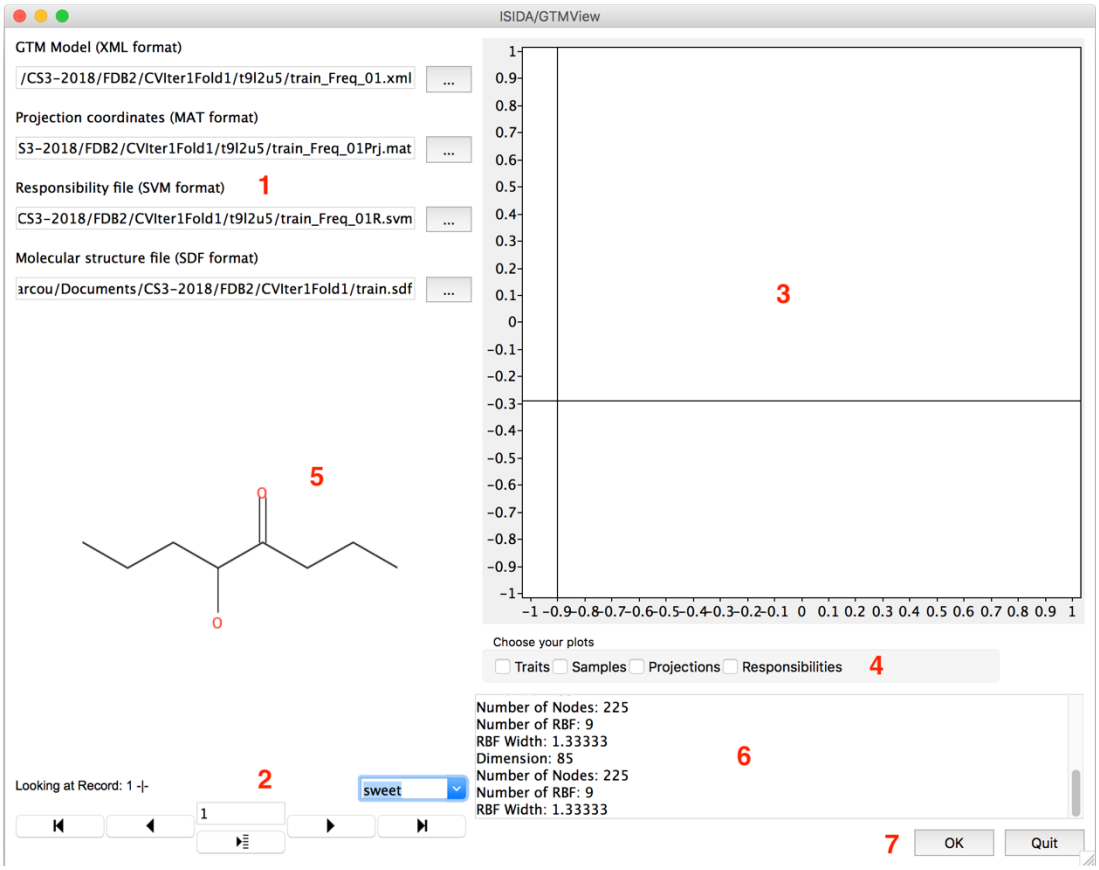

Figure 7. Interface of the xGTMView software. Input management is take care in (1). Navigation in the chemical structure file is performed in (2) and chemical structures are displayed in (5). The GTM data are plotted in (3) and controlled in (4). The log are written in (6) and the calculation are launched in (7).

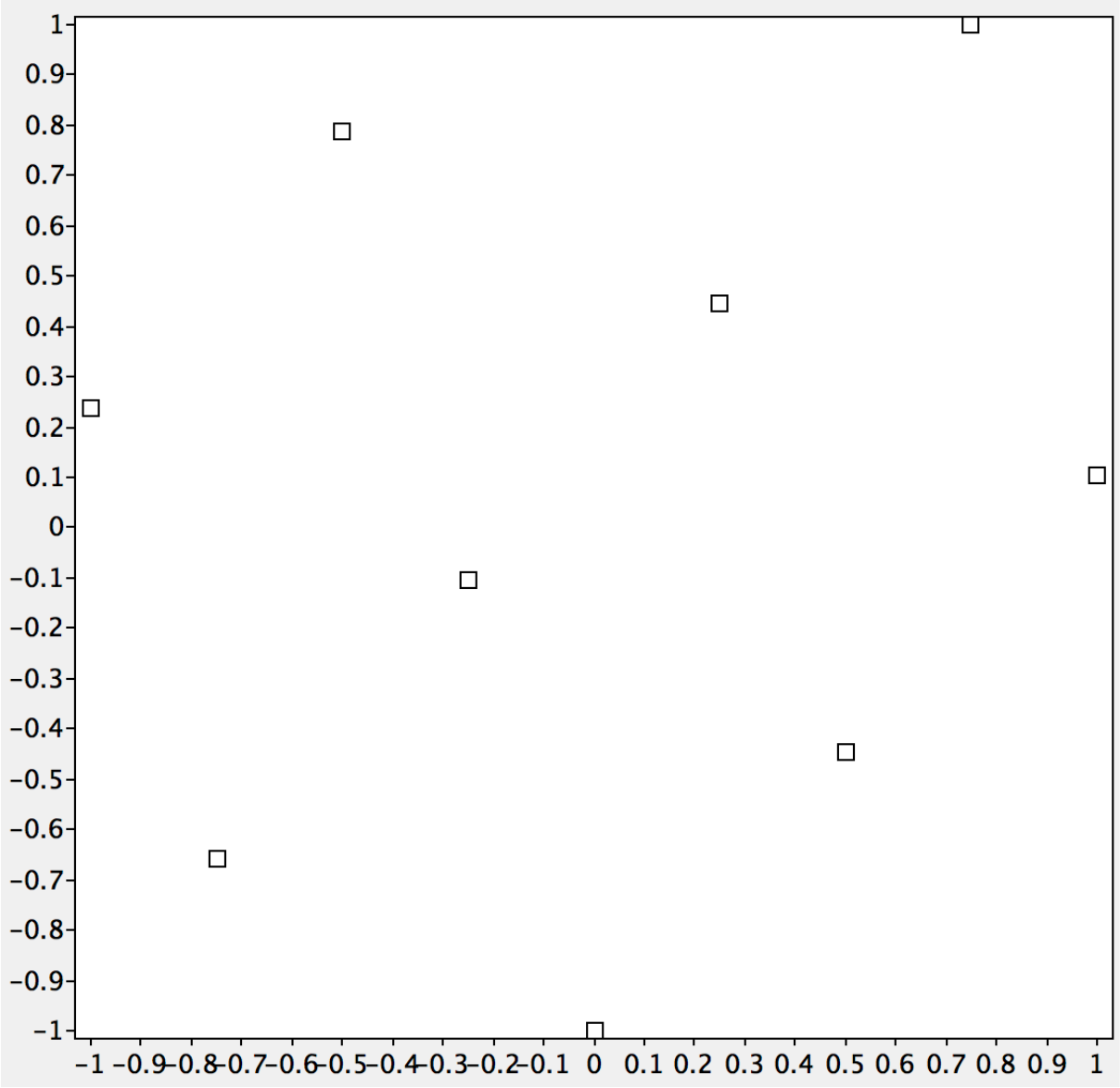

Figure 8. Position of the RBF centers (the traits) on the 2D manifold. The traits are positioned in a pseudo-regular way.

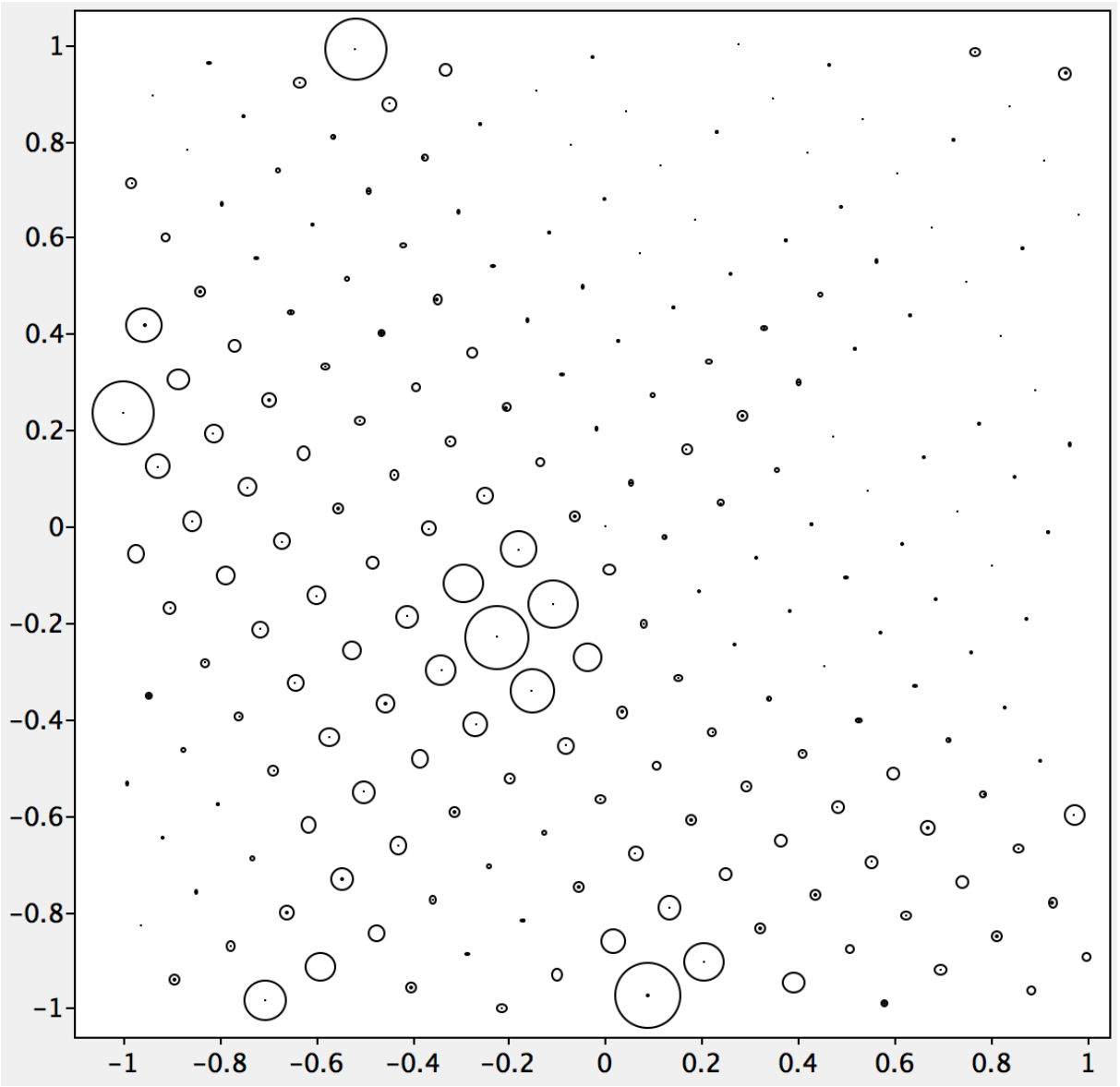

Figure 9. Positions of the sampling points of the manifold. These are the points were the density probability are estimated. The size of the circle around a sample point is proportional to the density of the chemical space region it is located in.

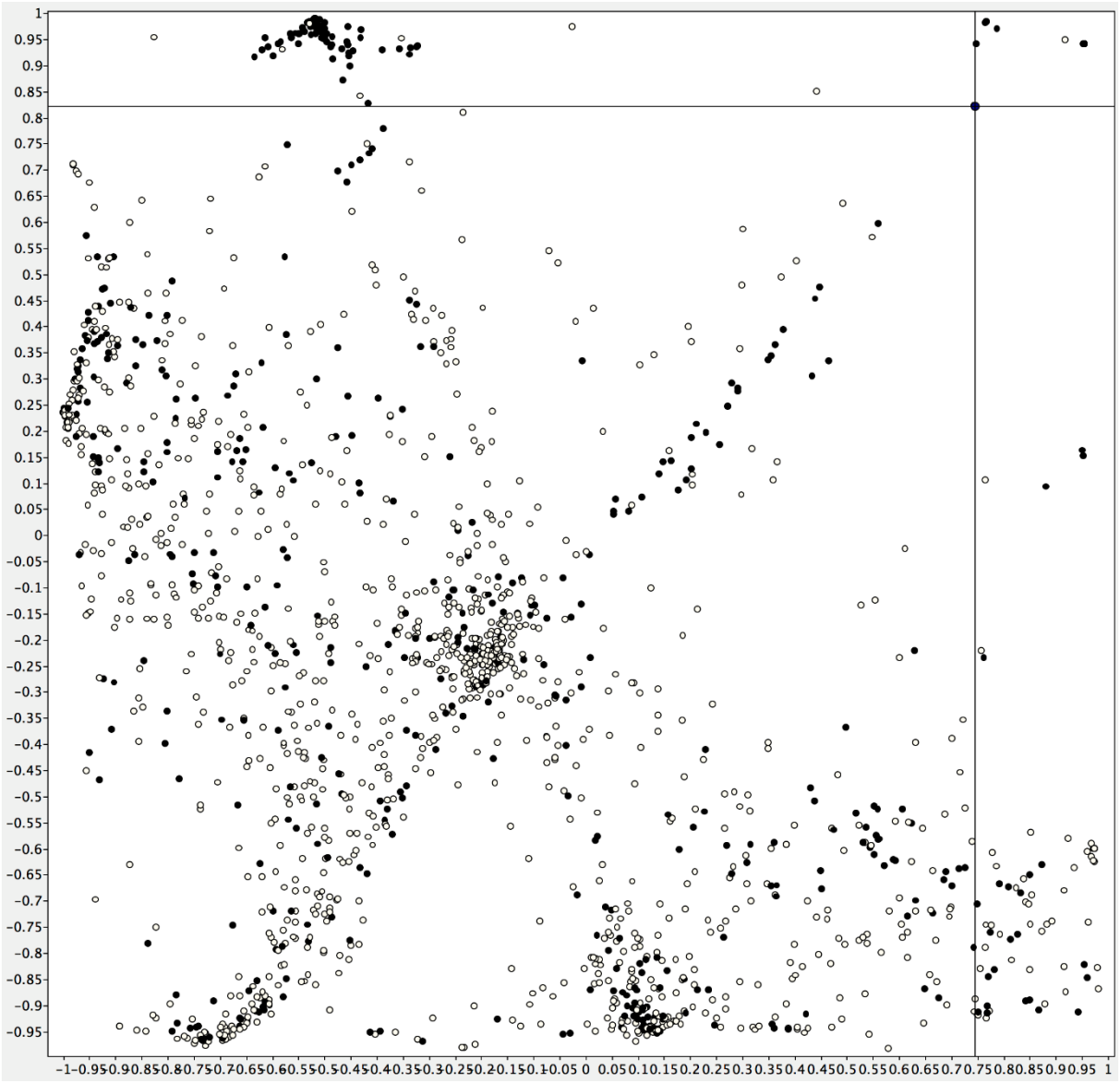

Figure 10. Projection of the training dataset on the GTM. Each point corresponds to a molecule. The black points are those compounds associated to the sweet taste. The cross and the emphasized point correspond to the selection of a particular molecule. The selected molecule is drawn in the region (5) of the interface (Figure 7).

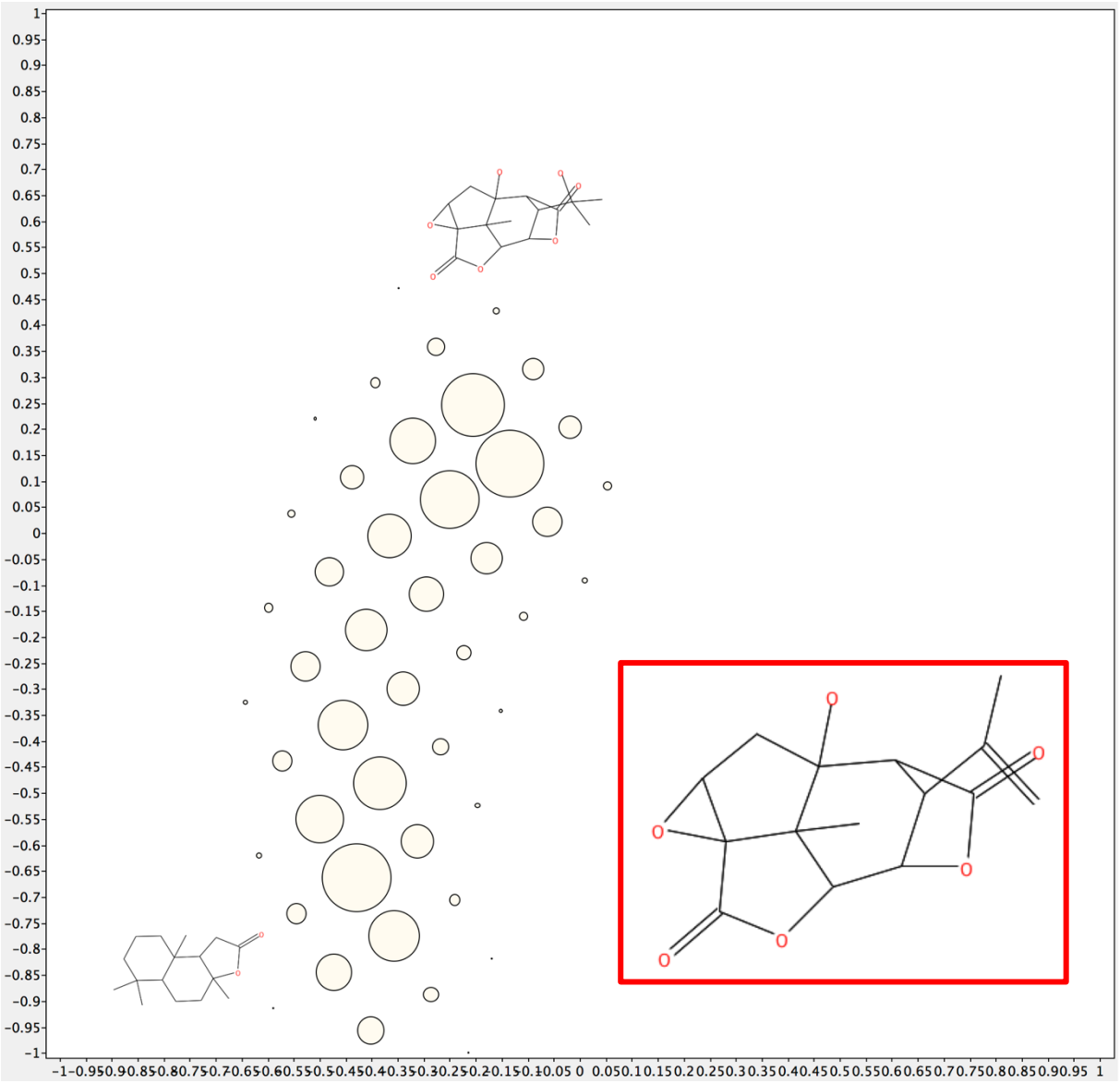

Figure 11. An example of an extended responsibility pattern. The corresponding molecule 118 is in the red frame. The molecules 537 is localized on the top of the responsibility pattern while the molecule 66 is localized near the bottom.

2.4. Exercise 4. Convergence of GTM fitting, visualization of the manifold

Instructions	***Comments***

• Open, if needed, the **xGTMapTool** application.
• Choose the **use model** option (Figure 1, area 4).
• Set up the input for the training set (Figure 1, area 1).
  ○ Choose as **input** the file `train_Freq_01.svm`.
  ○ Choose as **Model (XML)** the file `train_Freq_01.xml`.
	We will use the **xGTMapTool** application to generate additional information about the GTM manifold. More precisely, we search an alternative view to monitor the fit of the manifold to the dataset. The solution proposed is to visualize it in the 3D coordinates defined by the first three principal components[12] of the dataset.
	The software generates many additional files.

○ Tick the **Save full information** box. • Click the **OK** button.	`train_Freq_PC123.mat`: a three column file with the coordinates of the three first approximate principal component of the training set. These coordinates are used to initialize the GTM algorithm. `train_Freq_01Z.mat`: It contains the modified values of the molecular descriptors resulting from the pre-processing for each molecule projected on the GTM. `train_Freq_01Z3D.mat`: a three column file recording the coordinates of each molecule projected on the GTM, in the three first principal components coordinates system. `train_Freq_01WPhi`: Contains the coordinates of the manifold in the system of coordinates of the pre-processed molecular descriptors `train_Freq_01WPhi3D`: Contains the coordinates of the manifold in the system of coordinates of the three first principal component system.
• Open the **GTMmanifold** software (Figure 12). • Load the file `train_Freq_01Z3D.mat` in the top text box. • Load the file `train_Freq_01WPhi3D.mat` in the middle text box. • Load the file `train_Freq_01.xml` in the bottom text box. • Click the **OK** button. • *Optional*: if the plotted points are too small, you can use the slide bar at the bottom right hand corner of the plotting area.	The files train_Freq_01Z3D.mat and the train_Freq_01WPhi3D.mat are in theory sufficient to plot at the same time the manifold and the dataset in the same principal component coordinates system. However, in order to plot the surface using a tessellation rendering, it is needed to identify which nodes are members of the same triangle, which is an easy task using their coordinate on the manifold. This information is located in the model file `train_Freq_01.xml`. The picture illustrates how the manifold has twisted in order to accommodate the dataset. However, the picture is only approximative because the first three components are explaining less than 40% of the data. Besides, the manifold has a width corresponding to the standard deviation of the probability distribution that is not represented here. However, this is sufficient to monitor the training of the GTM and understand how it converges.
• Create a folder named Converge.	The following calculations will generate many files. It is wise to manipulate them in a

• Copy to this folder the file train_Freq_01.svm.	dedicated folder to keep the working space clean.
• Use the **xGTMapTool** application. • Choose the **train model** mode (Figure 1, area 4). • Set up the input for the training set (Figure 1, area 1). o Choose as **input** the file `train_Freq_01.svm`. o Choose as **output** the name `conv1`. o Set the **Preprocessing** to **standardize**. o Set the value **Number of traits** to 9 o Set the **Max. Number of Iterations** to 1. Click the **OK** button.	This setup will create a GTM model that will optimized during a single expectation-maximization step.
Repeat the previous procedure varying the number of iterations to 10, 20 ,30 ,40 and 50. Take care to changing the following values: o Set the **Max. Number of Iterations** to 10, 20, 30, 40 and 50. o **output** shall be set to `conv10`, `conv20`, `conv30`, `conv40`, `conv50`., respectively.	This will create a set of GTM models optimized over 10, 20, 30, 40 and 50 iterations of expectation-maximization.
Repeat the procedure of projection of the training data for each of the GTM models call conv1.xml, conv10.xml, conv20.xml, conv30.xml and conv50.xml. • Use the **xGTMapTool** application in the **use model** mode (Figure 1, area 4). • Tick the **Save full information** box. • Setup the **input** to the file `train_Freq_01.svm`. • For each of the GTM model files, set up the **Model (XML)** to the corresponding XML file. Then click the **OK** button.	To monitor the evolution of the manifold, it is now needed to apply each of the GTM models to the training data. The full information must be recorded to generate the files with the coordinates of the objects in the three dimensional coordinate system of the three first principal components.
• Report the likelihood as a function of the number of iterations of optimization steps.	The convergence of the likelihood is represented with more details in (Figure 13).

• Open with the **GTMmanifold** software the generated manifolds.	The corresponding shapes of the manifold should look as in Figure 14. The shape of the manifold is already stabilized after 30 iterations corresponding to a variation of the likelihood between two consecutive steps of about 0.01 units. Therefore, the default value of 0.001 of the parameter **Convergence: likelihood difference** seems sufficient. If the likelihood does not change more than this threshold during one optimization step, the optimization can be stopped.

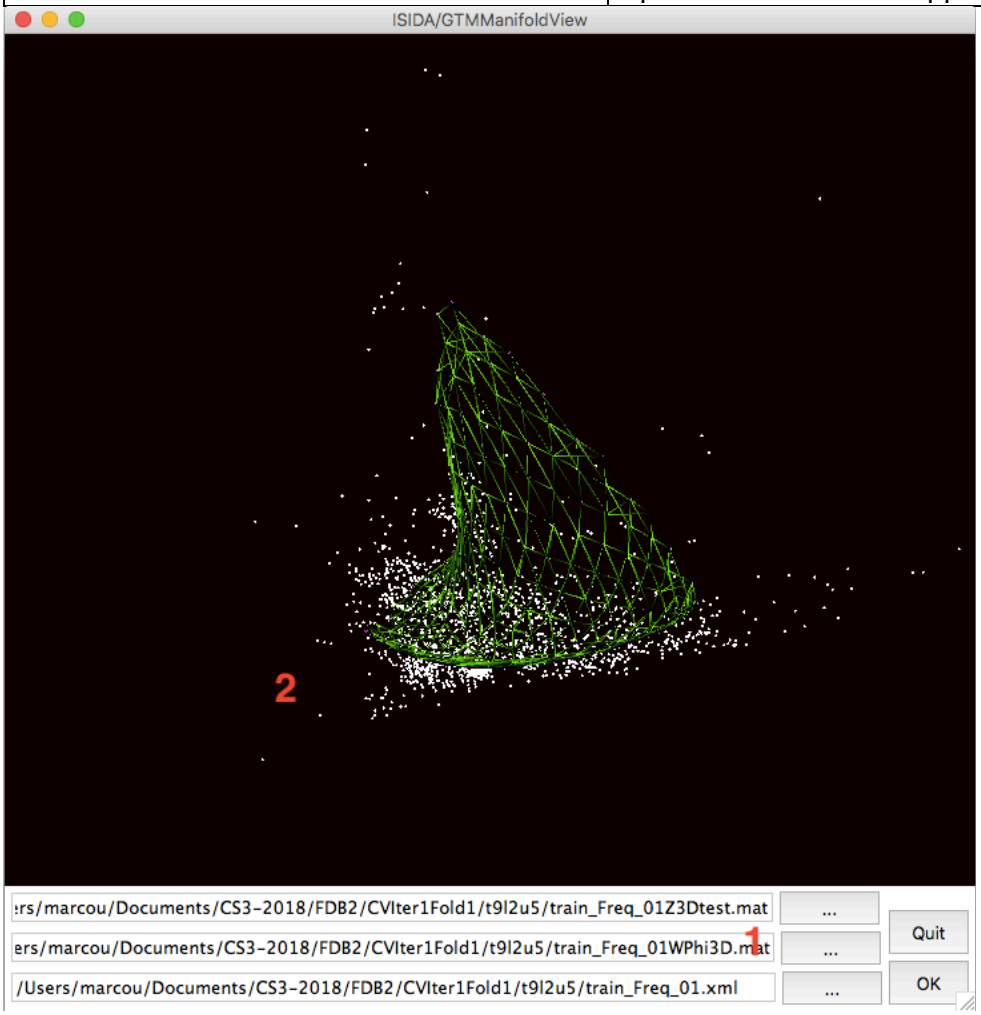

Figure 12. Interface of the GTMmanifold application. The zone (1) is used to load the 3D coordinates files illustrating the dataset and the manifold of the GTM. They are plotted in the area (2).

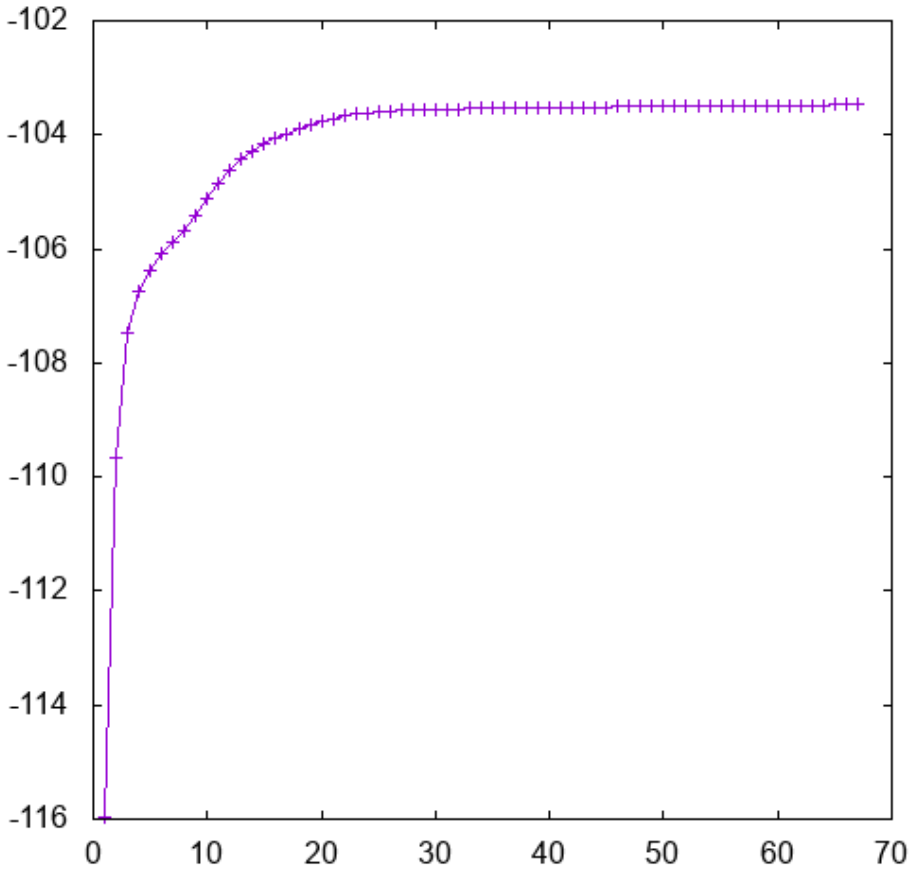

Figure 13. Evolution of the Likelihood with the increasing number of optimization steps of the GTM model.

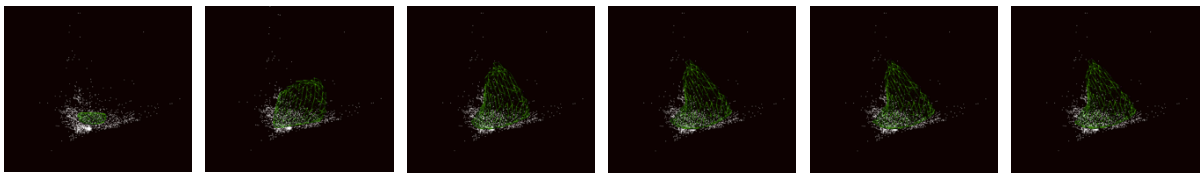

Figure 14. Evolution of the shape of the manifold over the increasing number of optimization steps of the GTM model.

Conclusion

This exercise offered the opportunity to take a closer look to the manifold at the heart of the GTM model. As the optimization process goes on, the manifold is tweaked towards the data points. The whole process is a balance between increasing the standard deviation of the of the normal distribution around the manifold and moving the RBF centers over the chemical space to improve the explanation of the dataset.

The optimization finishes when the likelihood change between two consecutive optimization steps is lower than a threshold value. The default value of this threshold, 0.001, seems relevant at least qualitatively.

2.5. Exercise 5. Optimization of parameters.

In the preceding exercises the number of RBF was set to 9 and all other parameters were left to default. In this exercise the question of optimal parameter choices will be asked.

Instructions	***Comments***
• Create a folder named `M`. • Copy to this folder the file `train_Freq_01.svm` and `test_Freq_01.svm`	As before, these systematic calculations are generating many files. It is wise to store them in dedicated folders to keep the working space tidy.

• Use the **`xGTMapTool`** application.
• Choose the **`train model`** mode (Figure 1, area 4).
• Set up the input for the training set (Figure 1, area 1).
  ○ Choose as **`input`** the file `train_Freq_01.svm`.
  ○ Choose as **`output`** the name `M1`.
  ○ Set the **`Preprocessing`** to **`standardize`**.
  ○ Set the value **`Number of traits`** to 1
  ○ Set the **`Max. Number of Iterations`** to 100.
• Click the **OK** button.
• Record in a spreadsheet the value of the likelihood of the last step of optimization (the value right to the word **`LLmap`** in the log window Figure 1, area 5).
	This will generate a GTM using only one RBF to define the manifold.
	The likelihood value during the last step of the optimization is estimating the likelihood of the training set according to the generated GTM model.

• Repeat the procedure to change systematically the number of RBF center from 5 to 15 by step of 2. ○ Set the value **`Number of traits`** to 5, 7, 9, 11, 13 and 15 • Change as **`output`** accordingly to `M5`, `M7`, `M9`, `M11`, `M13`, `M15`, respectively. • Record the likelihood values of each the last step of optimization (the value right to the word **`LLmap`** in the log window Figure 1, area 5).	A set of GTM models with varying number of traits is generated and the likelihood of the training set is stored.
• Choose the **`use model`** option (Figure 1, area 4). • Optionally, tick the **`Save full information`** box.	The likelihood increases systematically with the number of RBF centers. This is an expected behavior: the more they are, the more flexible becomes the manifold. It fits to the data more easily.

• Choose as **input** the file `test_Freq_01.svm`. • Apply all the GTM models generated so far. Set the input **Model (XML)** to M1.xml, M5.xml, M7.xml, M9.xml, M11.xml, M13.xml and M15.xml (Figure 1, area 1). • Record in a spreadsheet, the values of the likelihood.	However, when doing so, the difference of likelihood between the test set and the training set increases. This is symptomatic of overfitting. The results of a larger scale study on the same data are reproduced in Figure 15. It illustrate the situation. While the training set likelihood continues to increase, the test set is increasing at a lower rate. The choice of 9 RBFs in the previous exercises resulted from a choice to fit the training set and the test set approximately as well.
• Create a folder named `W`. Copy to this folder the file `train_Freq_01.svm` and `test_Freq_01.svm`	The next step is a systematic study of the influence of the width of the RBFs. This step will also generate a number of files and it is wise to keep them in separate place.

Using the **xGTMapTool** application.
• Choose the **train model** mode (Figure 1, area 4).
• Set up the input for the training set (Figure 1, area 1).
  ○ Choose as **input** the file `train_Freq_01.svm`.
  ○ Set the **Number of traits** to 9
  ○ Choose as **output** the name `W1_3`.
  ○ Set the **Preprocessing** to **standardize**.
  ○ Set the value of **RBF width** to 1.3
• Click the **OK** button.
Record in a spreadsheet the value of the likelihood of the last step of optimization (the value right to the word **LLmap** in the log window Figure 1, area 5).
	The default value of the RBF width is two times the average distance between two neighboring RBF centers. The manifold is a square extending into the range [-1,1]x[-1,1]. Its surface is therefore 4 squared units. Thus with 9 RBF, the default value of the RBF width is approximately 1.3.
	The current setup is close to the default.

• Repeat the procedure to change systematically the RBF width. ○ Set the value **RBF width** to 10, 1.0, 0.1, 0.01, and 0.001 • Change as **output** accordingly to `W10`, `W1`, `W0_1`, `W_01`, `W0_001`, respectively. • Record the likelihood values of each last step of optimization (the value	A set of GTM models using 9 RBF of varying width is generated and the likelihood of the training set is stored.

right to the word **LLmap** in the log window Figure 1, area 5).	
• Choose the **use model** option (Figure 1, area 4). • Optionally, tick the **Save full information** box. • Choose as **input** the file `test_Freq_01.svm`. • Apply all the GTM models generated so far with various RBF width. Set the input **Model (XML)** to W10.xml, W1.xml, W0_1.xml, W0_01.xml, and W0_001.xml (Figure 1, area 1). • Record in a spreadsheet, the values of the likelihood.	The coupling between the RBFs on the GTM is governed by their width. As the value increases, the coupling is stronger and the manifold cannot fit to the data. When the coupling disappears, the RBF are migrating freely and the notion of map is lost. At the same time, they tend to migrate over the center of the training set and the model globally loses its ability to explain the dataset. This explains the presence of a rather large optimum range of values of the RBF width, as illustrated in Figure 16. Here, it seems that setting the value of the width to 0.1 is beneficial.
• Create a folder named L. Copy to this folder the file `train_Freq_01.svm` and `test_Freq_01.svm`	Then, impact of the regularization parameter is scrutinized. As before, the study is realized in its own dedicated folder.

In the **xGTMapTool** application.
• Choose the **train model** mode (Figure 1, area 4).
• Set up the input for the training set (Figure 1, area 1).
  ○ Choose as **input** the file `train_Freq_01.svm`.
  ○ Set the **Number of traits** to 9
  ○ Set the value of **RBF width** to 0.1
  ○ Set the **Preprocessing** to **standardize**.
• Explore systematically the values of the regularization parameter
  ○ Set the value of **Regularization** to 100, 10, 1, 0.1, 0.01.
  ○ Set the output name to `L100`, `L10`, `L1`, `L0_1`, `L0_01` respectively.
  ○ Click the **OK** button after each complete setup.
Record the likelihood values of each last step of optimization (the right handed value to
	This step generates a collection of GTM models varying the value of the regularization parameter.
	The smaller the value of this parameter, the more free are the coefficients of the matrix defining the coordinates of the manifold. On contrary, large values of regularization will stiffen the manifold an prevent it to be deformed: it will stay flat.

the word **LLmap** in the log window Figure 1, area 5).	
• Choose the **use model** option (Figure 1, area 4). • Optionally, tick the **Save full information** box. • Choose as **input** the file `test_Freq_01.svm`. • Apply all the GTM models generated so far with various regularization values. Set the input **Model (XML)** to L100.xml, L10.xml, L1.xml, L0_01.xml and L0_001.xml (Figure 1, area 1). Record in a spreadsheet, the values of the likelihood.	The collected likelihood should follow a trend similar to Figure 17. At large values of the regularization, the manifold is stiff and it hardly differs from its initialization state. Upon decreasing he regularization value, the training set likelihood increases slowly and decreases slowly on the test set. An optimum value is located at a regularization value of 1.
• Create a folder named K. Copy to this folder the file `train_Freq_01.svm` and `test_Freq_01.svm`	The last part of the exercise will focus on the number of nodes in a GTM.

In the **xGTMapTool** application.
• Choose the **train model** mode (Figure 1, area 4).
• Set up the input for the training set (Figure 1, area 1).
  ○ Choose as **input** the file `train_Freq_01.svm`.
  ○ Set the **Number of traits** to 9
  ○ Set the value of **RBF width** to 0.1
  ○ Set the **Regularization** value to 1.0
  ○ Set the **Preprocessing** to **standardize**.
• Explore some values for the number of node
  ○ Set the value of **Number of samples** to 200, 300, 400, 500.
  ○ Set the output name to `K200`, `K300`, `K400`, and `K500` respectively.
  ○ Click the **OK** button after each complete setup.
Record the likelihood values of each last step of optimization (the right handed value to
	This step generates a collection of GTM models varying the number of nodes.
	The number of nodes is the least important parameter of a GTM. It is introduced in theory as a prior distribution over the manifold. Technically, it can also be interpreted as a numeric integration over the manifold to estimate the normal probability density around the manifold. Therefore, modifying its value is merely a change in the precision of this numerical integration.

the word **LLmap** in the log window Figure 1, area 5).	
• Choose the **use model** option (Figure 1, area 4). • Optionally, tick the **Save full information** box. • Choose as **input** the file `test_Freq_01.svm`. • Apply all the GTM models generated so far with various regularization values. Set the input **Model (XML)** to K200.xml, K300.xml, K400.xml, K500.xml (Figure 1, area 1). Record in a spreadsheet, the values of the likelihood.	As expected the number of nodes has a limited impact over the final estimation of the likelihood, for the training as well as for the test set. It is recommended to ensure a reasonable number of nodes for each RBF. In this implementation of the algorithm, the choice was to assign 25 nodes for each RBF.

Conclusion

The optimization of the parameters of the GTM can lead to very different pictures (Figure 18 and Figure 19). However, this is easily explained by diminishing the value of the RBF width. The manifold becomes very flexible and can eventually intersect itself. Therefore, the

responsibility patterns are redistributed. At the same time, the structural consistency of smaller clusters of compounds is improved.

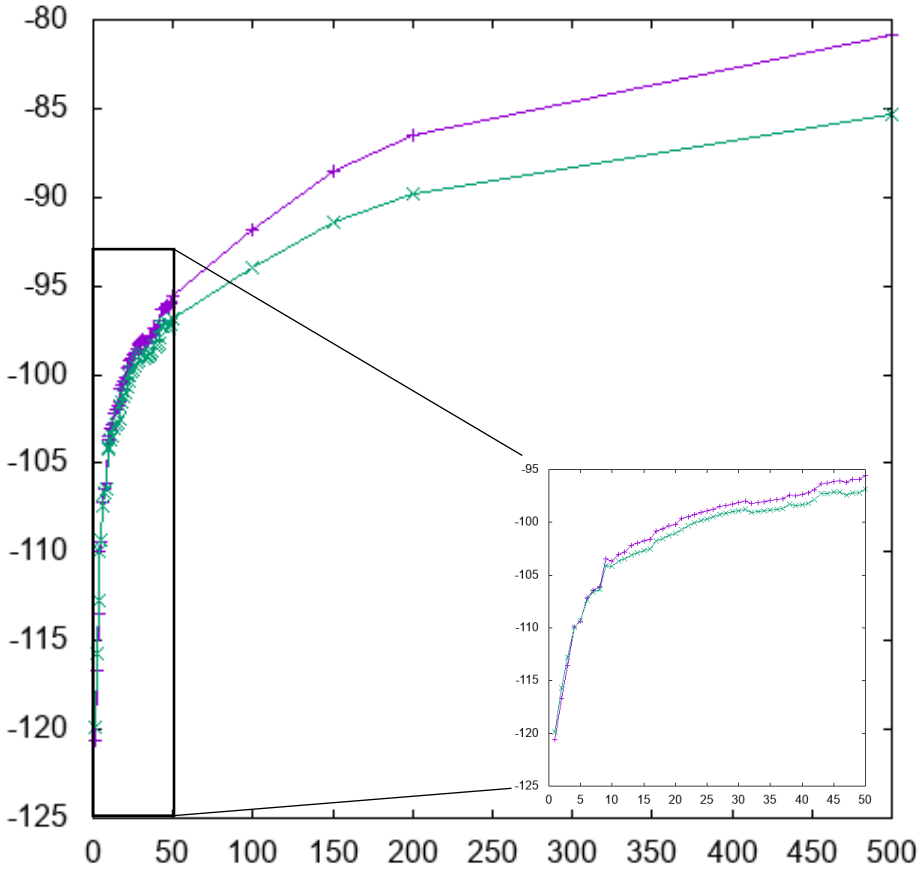

Figure 15. Evolution of the log likelyhood of the training set (violet line) and test set (green line) with the number of traits. A zoom on the lower values of the number of RBF is located on the right hand bottom corner.

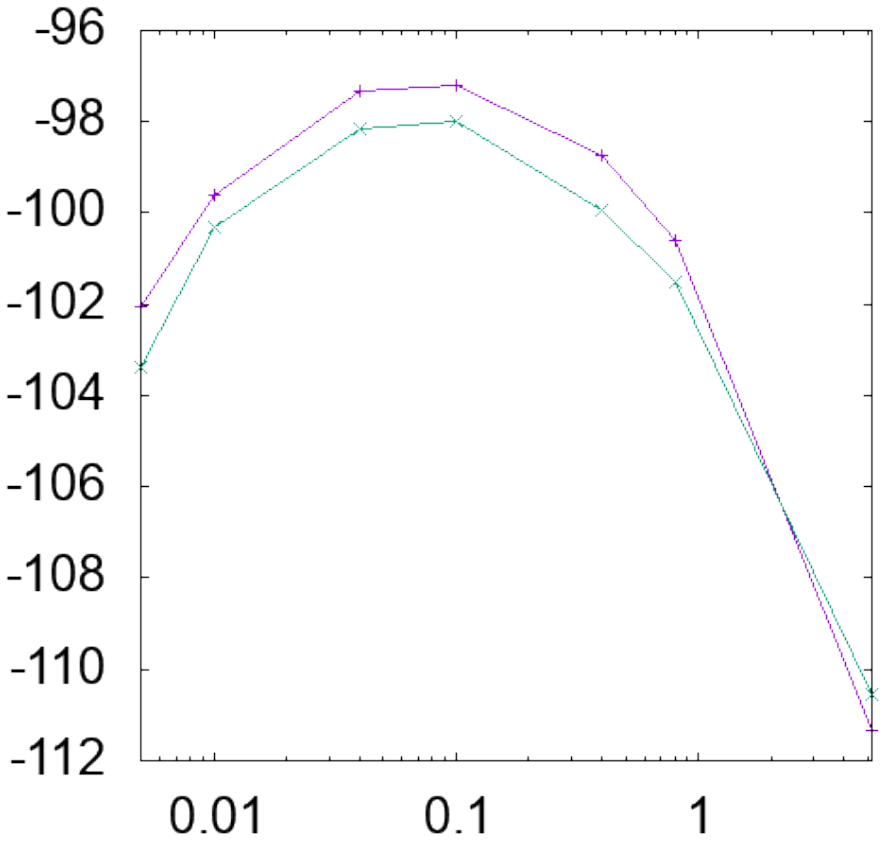

Figure 16. Evolution of the log likelyhood with the width of the RBF.

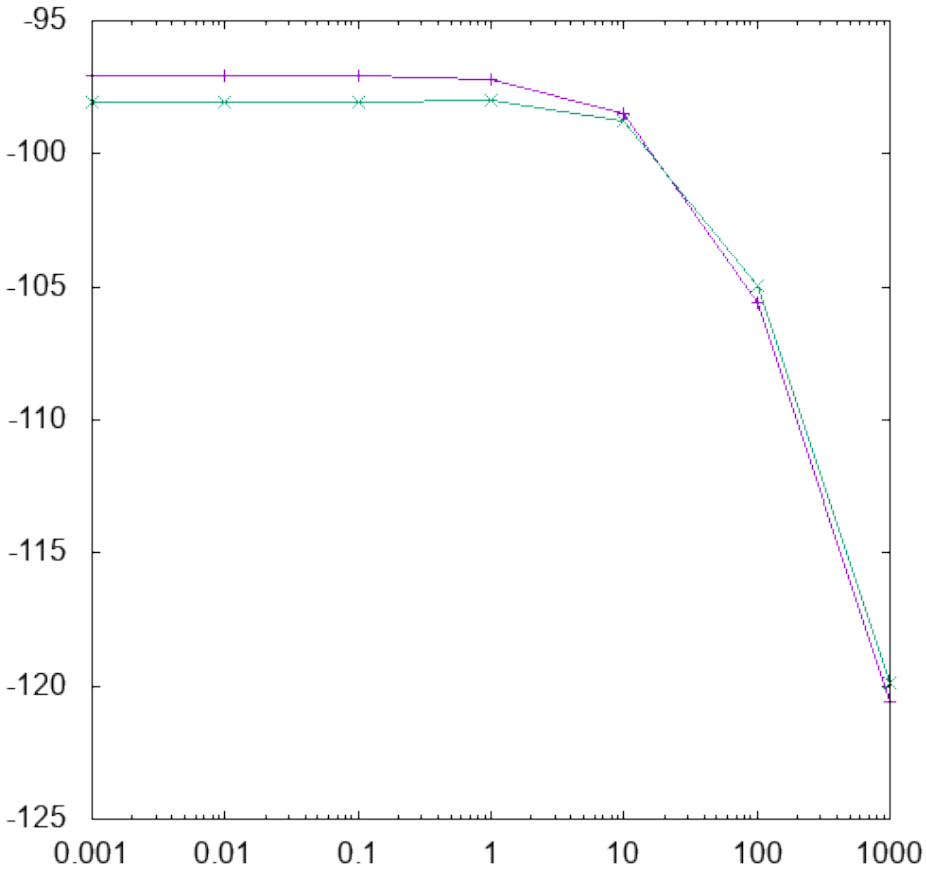

Figure 17. Evolution of the likelyhood with various values of the regularization.

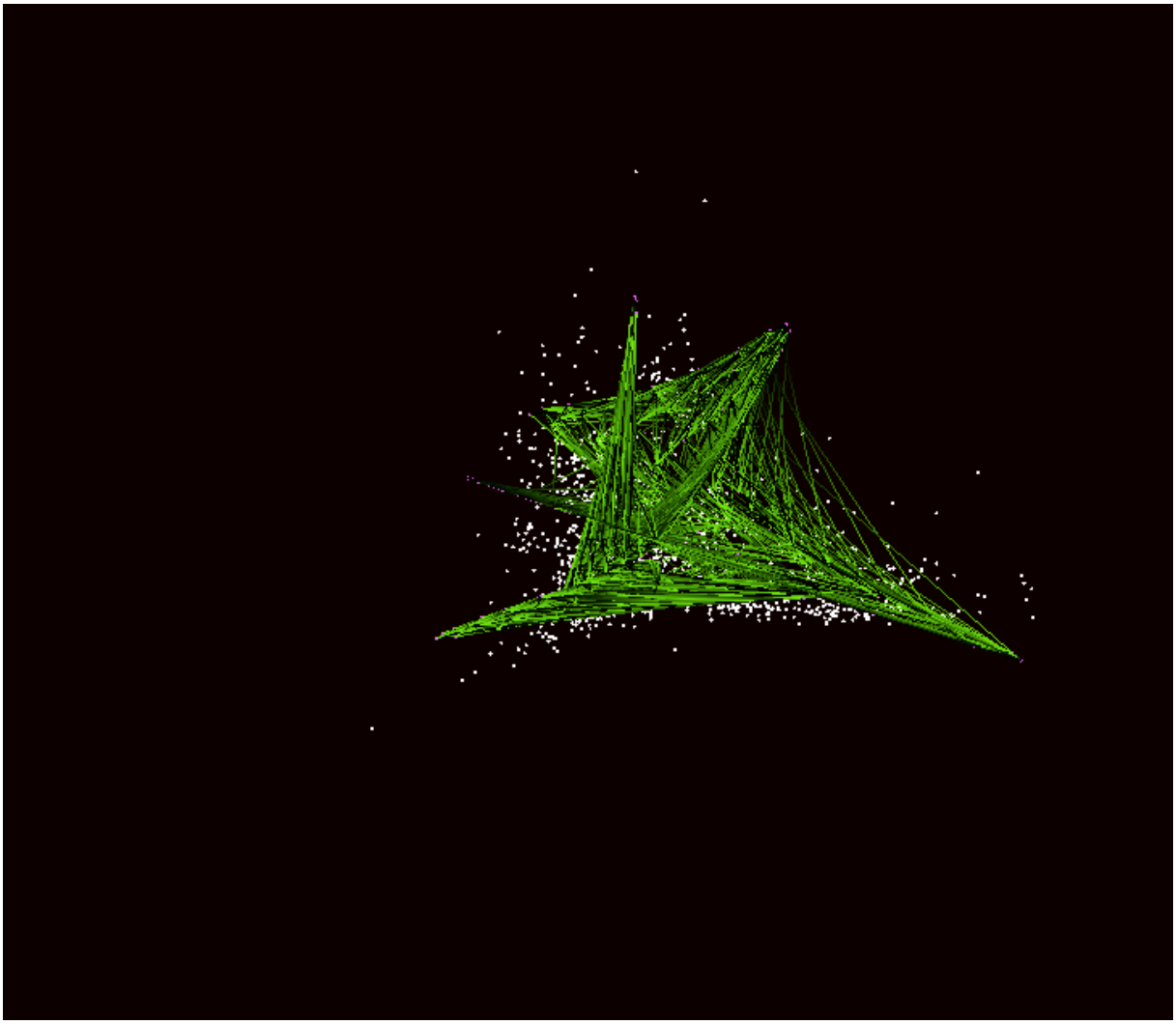

Figure 18. Optimized manifold with 9 RBFs of width 0.1, 500 nodes and a regularization coefficient of 1.

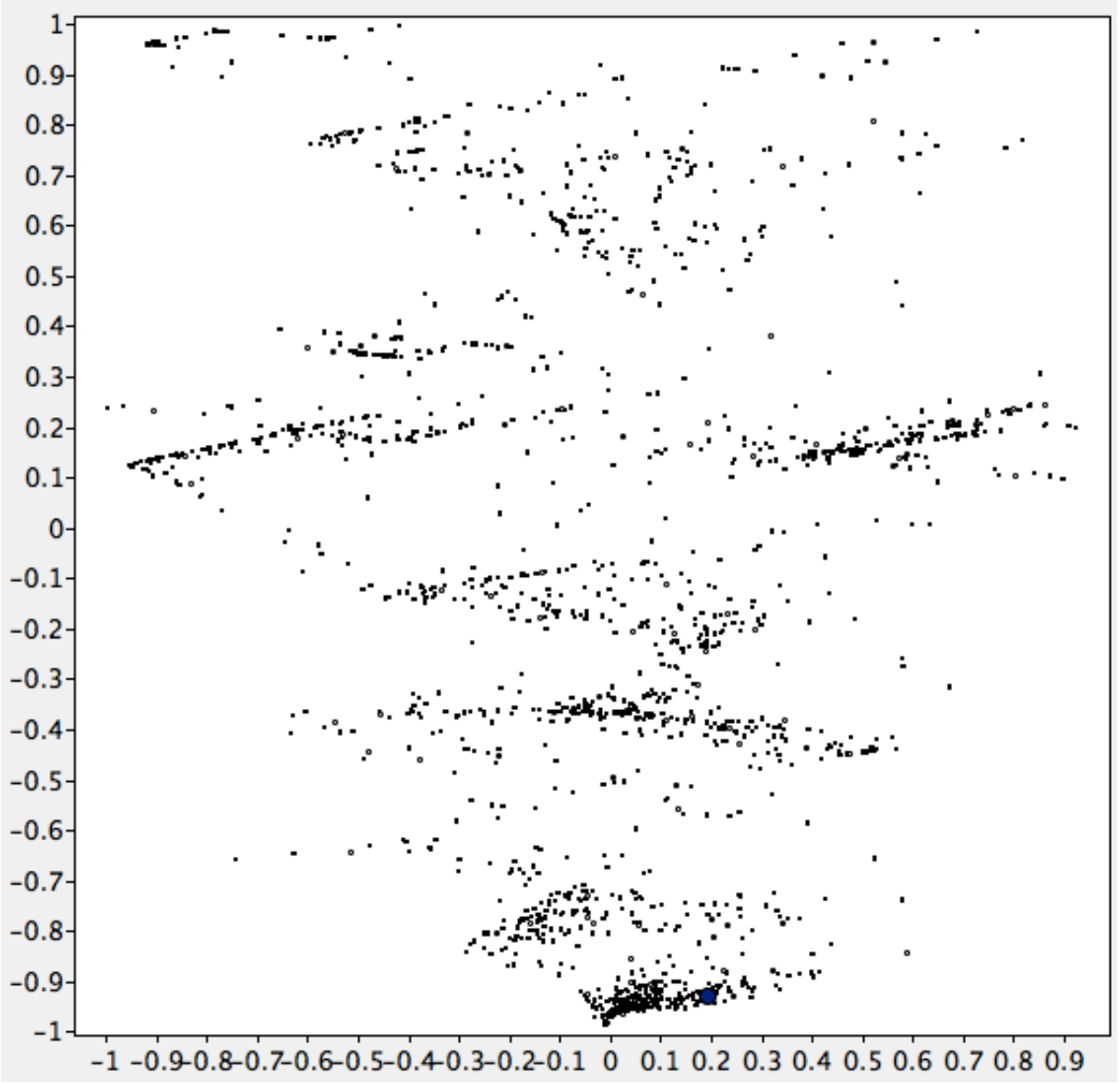

Figure 19. Projection of test compounds on the optimized GTM

3. Conclusion

The study of flavors is certainly a very fuzzy challenge. The flavor descriptions are for a large part subjective and therefore, the flavors labelling tend to be very noisy. Nonetheless, the rationality behind flavors has been demonstrated several times, illustrated by successful QSPR studies able to discriminate the sweet or the bitter taste with high performances.

However, unsupervised methods can be very relevant in this context, because they are not affected by the labels of the compounds and by the difficulties of curation of flavor labels. For this reason, the GTM approach is particularly suited. This illustrated in the beginning of the tutorial.

However, getting a meaningful picture of the chemical space of flavors requires some investigation about the algorithm itself. The exercises illustrated the generation and analysis of GTM and propose an optimization procedure. Although the procedure is tedious, the main results is that the number of RBF is the most important parameter to set in a GTM. For all others, the heuristics implemented make sense. Typically, the width of the RBF is set to cover

two times the typical distance between two RBF centers on the manifold, thus ensuring a reasonable stiffness of the manifold and reducing the chances of overfitting. At the same time, the regularization parameters can be set to 1 corresponding to a situation where each element of the GLR describing the manifold follows a standard distribution. Finally, the number of node, is rather a modification of the resolution of the map. It can be set to low value in an exploratory phase, then to large values, in order to produce better quality visualizations.
Thus, with only one important parameter to set, the GTM can be considered as rather simple method to visualize the chemical space.
Finally, an important aspect of the GTM through visualization, is the freedom of representation of the data. All steps of the calculations are generating files that are easy to read and to plot. The end user shall have the choice of the software and the tools to create custom representation, emphasizing the features of the map to support its observations.

4. Bibliography

[1] C. M. Bishop, M. Svensén, C. K. Williams, *Neural computation* **1998**, *10*, 215-234.
[2] N. Garg, A. Sethupathy, R. Tuwani, S. Dokania, A. Iyer, A. Gupta, S. Agrawal, N. Singh, S. Shukla, K. Kathuria, *Nucleic acids research* **2017**, *46*, D1210-D1216.
[3] A. Scalbert, C. Andres-Lacueva, M. Arita, P. Kroon, C. Manach, M. Urpi-Sarda, D. Wishart, *Journal of agricultural and food chemistry* **2011**, *59*, 4331-4348.
[4] A. Wiener, M. Shudler, A. Levit, M. Y. Niv, *Nucleic acids research* **2011**, *40*, D413-D419.
[5] J. Ahmed, S. Preissner, M. Dunkel, C. L. Worth, A. Eckert, R. Preissner, *Nucleic acids research* **2010**, *39*, D377-D382.
[6] M. Dunkel, U. Schmidt, S. Struck, L. Berger, B. Gruening, J. Hossbach, I. S. Jaeger, U. Effmert, B. Piechulla, R. Eriksson, *Nucleic acids research* **2008**, *37*, D291-D294.
[7] H. Arn, T. Acree, *Developments in Food Science* **1998**, *40*, 27-28.
[8] G. A. Burdock, *Fenaroli's handbook of flavor ingredients*, CRC press, **2016**.
[9] C. Rojas, R. Todeschini, D. Ballabio, A. Mauri, V. Consonni, P. Tripaldi, F. Grisoni, *Frontiers in chemistry* **2017**, *5*, 53.
[10] J. Leffingwell, D. Leffingwell, 10 ed., Leffingwell and associate, http://www.leffingwell.com/flavbase.htm, **2011**.
[11] K. Martínez-Mayorga, T. L. Peppard, A. B. Yongye, R. Santos, M. Giulianotti, J. L. Medina-Franco, *Journal of Chemometrics* **2011**, *25*, 550-560.
[12] H. Hotelling, *Journal of educational psychology* **1933**, *24*, 417.