Tutorial on Generative Topographic Mapping Landscapes

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

1. Introduction

This tutorial will explain how to build a GTM and use it to visualize the density distribution of a compound library in the chemical space, then use the map to draw a property landscape (for regression) or an activity landscape (for classification). It is illustrated on an environmental fate property, the bioconcentration factor.

Software and files

The tutorial uses the following software:

- **xGTMapTool**: a graphical user interface frontend for the preparation of a GTM
- **xGTMView**: a software for visualization of the GTM and for their chemical interpretation.
- **xGTMReSample**: a software to change the resolution of the map: its number of nodes.
- **xGTMClass**: a software that prepares an activity landscape and estimate the classification performances of the model.
- **xGTMReg**: a software the prepare a property landscape and estimate the regression performances of the model.
- **xGTMLandscape**: a software that allows a visualization of the landscape and its chemical interpretation.

In the frame of this tutorial, two concepts are used in a specific way:

- *Property landscape*: a map which is color coded according to a property value. The map results from a responsibility weighted average of the contribution of training dataset compounds. It estimates the likely value of the property at this location.
- *Activity landscape*: a map which is color coded according to a class. There are two possible colorations for a given class. The first one monitors the density of those compounds from this class. The second one refers to the probability of compounds in a location of the map to be a member of this class.

The following files are provided:

- `train.sdf` and `test.sdf`: The chemical structures in SDF format. The SD fields are:
  - *MolNr*: molecule number in the dataset;
  - *CAS*: Chemical Abstract Service identifier;
  - *STDmols (InChi)*: Standardized structure in InChi format;
  - *STDmols (InChiKey)*: Standardized structure InChi Key;
  - *STDmols (canonical SMILES)*: Chemical structure in SMILES format;
  - *PHTYP*: Pharmacophoric labels of the atoms;
  - *FFTYP*: Amber force field labels of the atoms;

    - *logBCF*: Experimental log value of the bioconcentration factor (in L/Kg);
    - *BCFcl*: The class of the compound either as bioconcentrating or not (BCF/notBCF).
- `train.hdr`: the header file describing the molecular descriptors used.
- `train.svm` and `test.svm`: the actual molecular descriptors matrices corresponding to the training and test sets, respectively.
- `trainBCFlog.prp` and `testBCFlog.prp`: the property files that store the activity of the compound as the logarithm of experimental bioconcentration factor, in the same order as the corresponding SDF files.
- `trainBCFcl.prp` and `testBCFcl.prp`: the property files that store the activity of the compound as bioconcentrating (BC) or not bioconcentrating (notBC), in the same order as the corresponding SDF files.

The tutorial provides files that are pre-generated but are outputs of the instructions:

- `train.xml`: the GTM trained on the training set data;
- `trainPrj.mat` and `testPrj.mat`: the projections of the training and test set compounds on the map.
- `trainR.svm` and `testR.svm`: the responsibilities of the training and test set compounds on the map.
- `trainBCFlog_reg.xml`: the property landscape of the training set as logarithm of the BCF.
- `trainBCFcl_cls.xml`: the activity landscape of the training set as BC/notBC classes.
- `*Dens.mat`: Three column files locating the GTM nodes and the local density.
- `*LS.mat`: Multi-column files, the first two being the (x,y) location of the GTM nodes and the others are the landscape values.
- `*k8000*`: Files generated after resampling the GTM using 8000 nodes.

License

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

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

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

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

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

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

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

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

The Bioaccumulation Factor dataset

The determination of Bioconcentration Factor (BCF) is a mandatory parameter used for the PBT/vPvB (Persistent Bioaccumulative and Toxic/very Persistent very Bioaccumulative) substances assessment by the European Union Registration, Evaluation, Authorisation and Restriction of Chemical Substances Regulation (REACH, EC No 1907/2006). In Europe a substance is not considered to possess a significant bioaccumulation potential below a BCF value of 2000 L/Kg (or 3.3 log unit), then it is considered as "bioaccumulative" up to 5000 L/Kg (or 3.7 log unit) and "very bioaccumulative" above [1]. However, acquisition of BCF data is expensive, and requires the sacrifice of animal lives. This explains the attention that deserves alternative methods and in particular QSAR[2].

Bioconcentration experimental data was collected from multiple sources, including several publicly available databases and literature research: the Japanese National Institute of Technology and Evaluation (NITE)[3], the European Chemical Industry Council Long Range Initiative (CEFIC LRI)[4], the Canadian Domestic Substance List (DSL)[5] and the ECOTOXicology knowledgebase of the US Environmental Protection Agency (ECOTOX EPA)[6] (accessed through the OECD Toolbox[7]), and the database of ECHA (accessed through the eChem portal[8]). Additional values were retrieved from literature from the works of Arnot and Gobas[9], Dimitrov et al.[10] and Fu et al.[11]. It is publicly available on the Zenodo platform: https://doi.org/10.1080/1062936X.2019.1626278.

The following entries were excluded: inorganic, polymer, UVCBs (Unknown or Variable composition, Complex reaction products or Biological materials). When the BCF value was not reported in L/Kg of body weight, not calculated on a whole-body measurement-basis or the test was performed on a non-recommended OECD species, the value was excluded. Since these are important study conditions that have to be explicitly stated [3], entries which were missing such details were excluded as being of lower reliability. Chemical structures were standardized and duplicates were removed. When multiple BCF values were available for a given compound, the median was taken as representative value. For some substances the range of BCF values could reach two log units.

The classes have been determined using the thresholds mentioned above. The label notBC is attributed to compounds with a logBCF (logarithm of the bioconcentration factor, expressed in L/Kg) value lower or equal to 3.3. The label BC is attributed to compounds with a logBCF value larger than 3.3.

The Generative Topographic Maps landscapes

Step by step instructions

The exercises are developed to introduce the concept of predictive landscapes based on the GTM approach. They start with the generation of a GTM (Exercise 1) that will be visualized (Exercise 2). In the following, the resolution of the map will be increased (Exercise 3). In the next step, you will be guided in building and validation of an activity landscape, for classification problem (Exercise 4). Then, you will be guided on the building and validation of a property landscape, for regression problem (Exercise 5). Finally, the tutorial will introduce

the visualization tool to analyze the obtained landscapes and discuss its chemical content (Exercise 6).

1.1. Exercise 1. Train a GTM.

The aim of this exercise is to train a GTM on a training dataset, then use the built GTM to analyze a test set.

Inputs:

- `train.svm`
- `test.svm`

Outputs:

- `train.xml`
- `trainR.svm`, `trainPrj.svm`, `trainPC123.mat`, `trainZ.mat`, `trainZ3D.mat`, `trainPhi.mat`, `trainPhi3D.mat`
- `testR.svm`, `testPrj.svm`

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.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 **center** option. In the **`Initial scaling`** list, select the item **`Standard`**.	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. The **`Standard`** initialization of the manifold is to extend it according to the loadings of the first two principal components. This is a reasonable choice to avoid a bias from the largest compounds of the dataset. However if that population should critically be represented by the GTM, then it is better to scale the manifold in order that it covers the whole dataset, using the command **`Extended`**. Finally, it is possible to tune this initialization with the **`Custom`** command, requiring as input a scaling factor from the default, **`Standard`**, setup.

	Note: if the principal components have been already computed, it is possible to load them using the **Precomputed PCA** element of the interface.
Set the **Number of traits** value to 110 (Figure 1, area 3), the interface should look as on Figure 2. then click on the button **OK** (Figure 1, area 6). *Caution*: the calculation takes about 15 minutes.	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 4). 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 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 (Figure 5): • 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 6) informs that the process terminated successfully and the last iteration is informative about the log likelihood of the studied dataset.
Click the **Use model** radio button	This action switches the interface to project a dataset on the GTM.
Click the button to the right of the **Model (XML)** label (Figure 1, area 1) and select the file `train.xml`. The **Input** should be `train.svm` and **Output** should be `train`. Tick the **Save full information** box.	This command project the training dataset on the GTM manifold. It will generate detailed information in a set of files. • `trainPrj.mat`: the coordinates of the compounds on the map.

The interface should look like Figure 3. Click the **OK** button.	• `trainR.svm`: are the responsibilities of the corresponding compounds. • `trainZ.mat`: the pre-processed dataset • `trainPC123.mat`: the first three principal components, they can be reused to bypass PCA calculations for training a GTM (if the pre-processing is the same). • `trainWPhi.mat`: The GTM nodes coordinates in the initial space • `trainWPhi3D.mat`: the GTM nodes projections on the first 3 principal components. • `trainZ3D.mat`: the dataset projection on the first 3 principal components. The log likelihood is -208.86.
Click the button to the right of the **`Input`** label (Figure 1, area 1) and select the file `test.svm`. The file name in the **`Output`** box should update to `test`. Untick the **`Save full information box`**. Click the **OK** button.	This operation project the test dataset on the GTM manifold. By default, only the projections (`testPrj.mat`) and the responsibilities (`testR.mat`) are saved. The loglikelihood of the test set is -205.52. Thus the test set is explained equivalently to the training set. A variation of a few log likelihood unit is not significative in this case. This can be evidenced by repeating the process with a different composition of the training and test set. slightly less explained by the model. To get a scale of variations of the loglikelihood, it is useful to compare to how the initial flat state of the manifold explains the data. This information is the located at the top of the training log: `First LLt=-1427.36` (Figure 4). Actually the number of nodes have been optimized to this end.

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.svm` is used to train a GTM model. The number of traits is set so that the model generalizes to a test set taken from the same distribution. The software is used to output additional information on the GTM model of the training set and to project a the test set on the manifold. The good generalization of the model on the test data is illustrated by the correspondence of the loglikelihood score of both dataset. This difference can be compared to the scale of loglikelihood values explored when training the manifold from a flat geometry to the final optimized one.

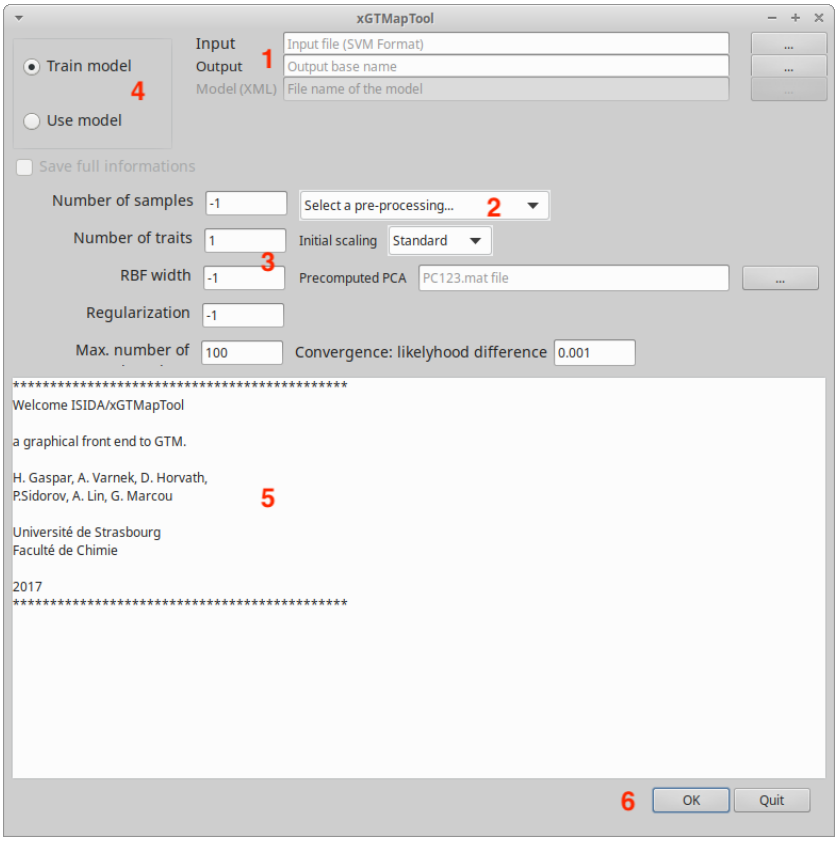

Figure 1. The interface of the xGTMapTool application. The file management is operated in the region (1) of the interface. The preprocessing is taken care of in (2) and the parameterization of the model is performed in (3). The use of the interface to train or apply a GTM model is controlled in (4).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

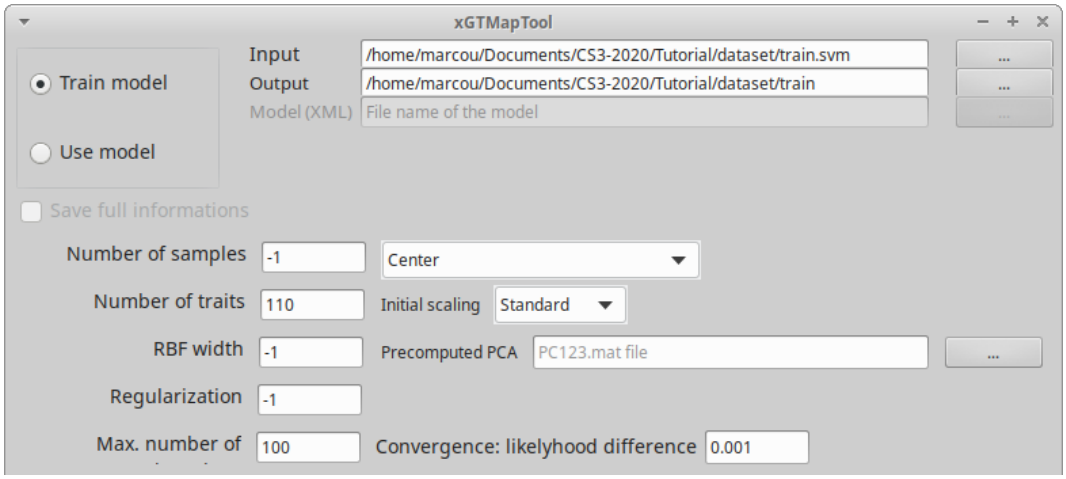

Figure 2: State of the interface to train the GTM on BCF data.

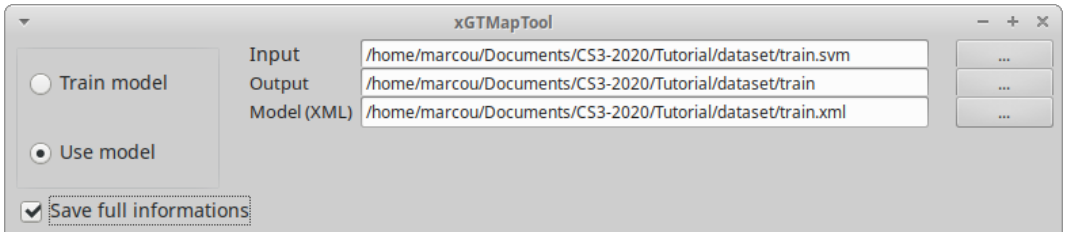

Figure 3: State of the interface to project the BCF training data on the GTM.

Figure 4. State of the log when starting the calculations. The parameter values of the model are explicited.

Figure 5. 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.

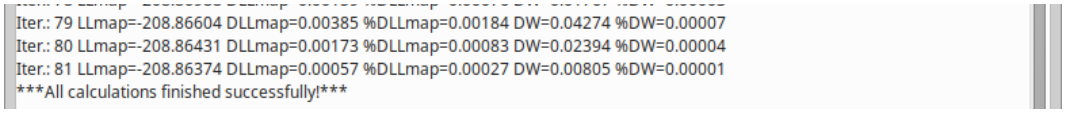

Figure 6. Last iteration of the training of the GTM.

1.2. Exercise 2. Visualize the projected data

The aim of this exercise is to analyze the datasets in an unsupervised way using the GTM. The training sets and test sets are scrutinized, order to get an understanding of their chemical content and localization of chemotypes in the chemical space.

Inputs:

- `train.xml`, `trainR.svm`, `trainPrj.mat`, `train.sdf`
- `testR.svm`, `testPrj.mat`, `test.sdf`

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 is 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.xml`. • If needed, click the **Projection coordinates (MAT format)** button and chose the file `trainPrj.mat`. • Check also that the corresponding `trainR.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.	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 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). • Untick the **Traits** box, then tick the **Samples** box. • Untick the **Samples** box, then tick the **Projections** box. • Tick the **Responsibility** box. *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 (Figure 7, area 3) changes according to the state of the tick boxes. It shall display, in the order: • the localization of the RBF on the latent space (Figure 8); • the positions of the nodes of the GTM (Figure 9); • the location of each molecule on the map (Figure 10); • the responsibility pattern of the selected compound. The dots of the plot are clickable. On click, the corresponding compound is selected and displayed (Figure 7, area 5).

Change the input files setup (Figure 7, area 1). • Click the **Projection coordinates (MAT format)** button and chose the file testPrj.mat. • Check that the corresponding testR.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 test.sdf. • Click the **OK** button. • Click the **Projections** box.	These operations will load the projections of the test set. The distribution of test compounds shall overlap nicely the one observed for the training set. The same chemotypes should be found in the test set distribution as those observed for training set. The test set distribution is illustrated on Figure 11.
Click on the SDF field selector ((Figure 7, area 2) and select the field BCFclass.	This operation color the dots of the plot according to the SDF field value. This field indicates two classes: not bioaccumulating (notBCF; logBCF<=3.3) and bioaccumulating (BCF; logBCF>3.3). The property should already concentrated, being indicative of the regions of the chemical space that are a concern for bioaccumulation.

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 and localization of a property of interest.

The latent space distribution is sampled at 2700 nodes. This a bit too low, thus several compounds are overlapped on the very location of these nodes. A simple workaround is to increase the number of nodes.

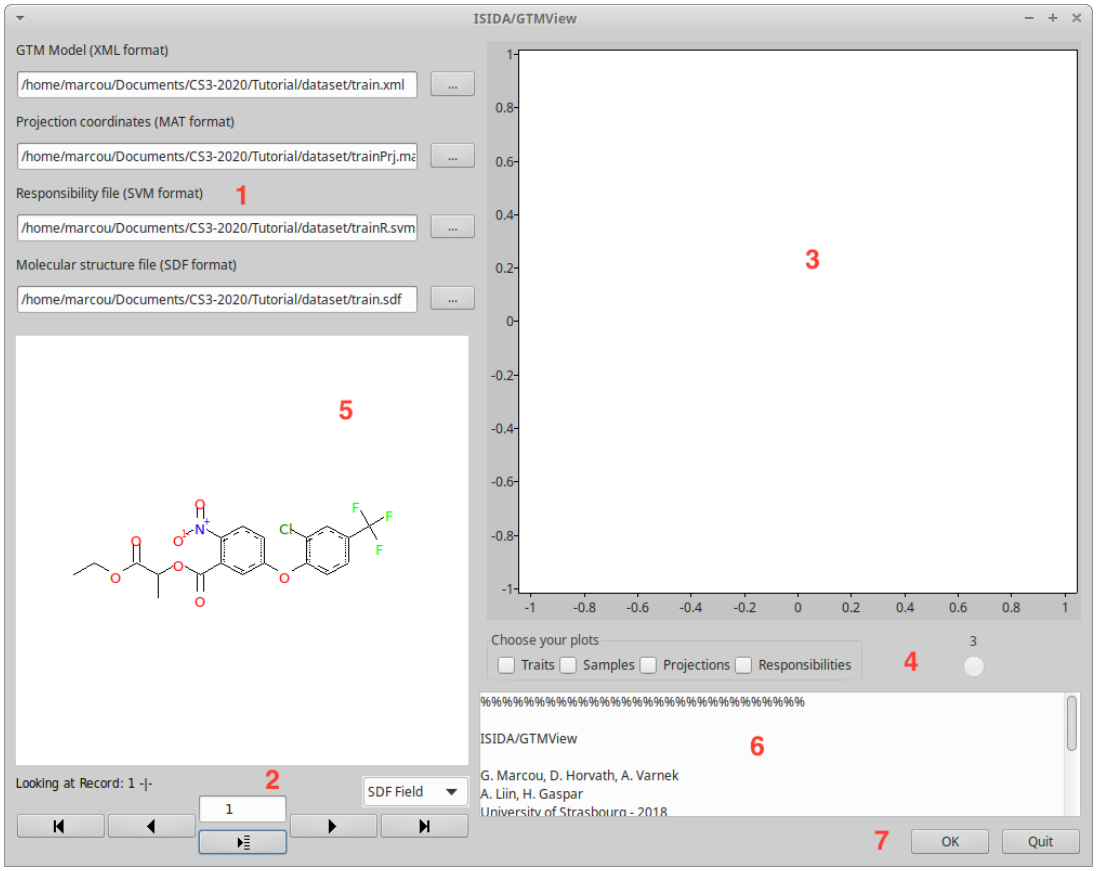

Figure 7. Interface of the xGTMView software. Input management is take care in (1). Navigation in the chemical structure file is performed in (2) and chemical structures are displayed in (5). The GTM data are plotted in (3) and controlled in (4). The log are written in (6) and the calculation are launched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

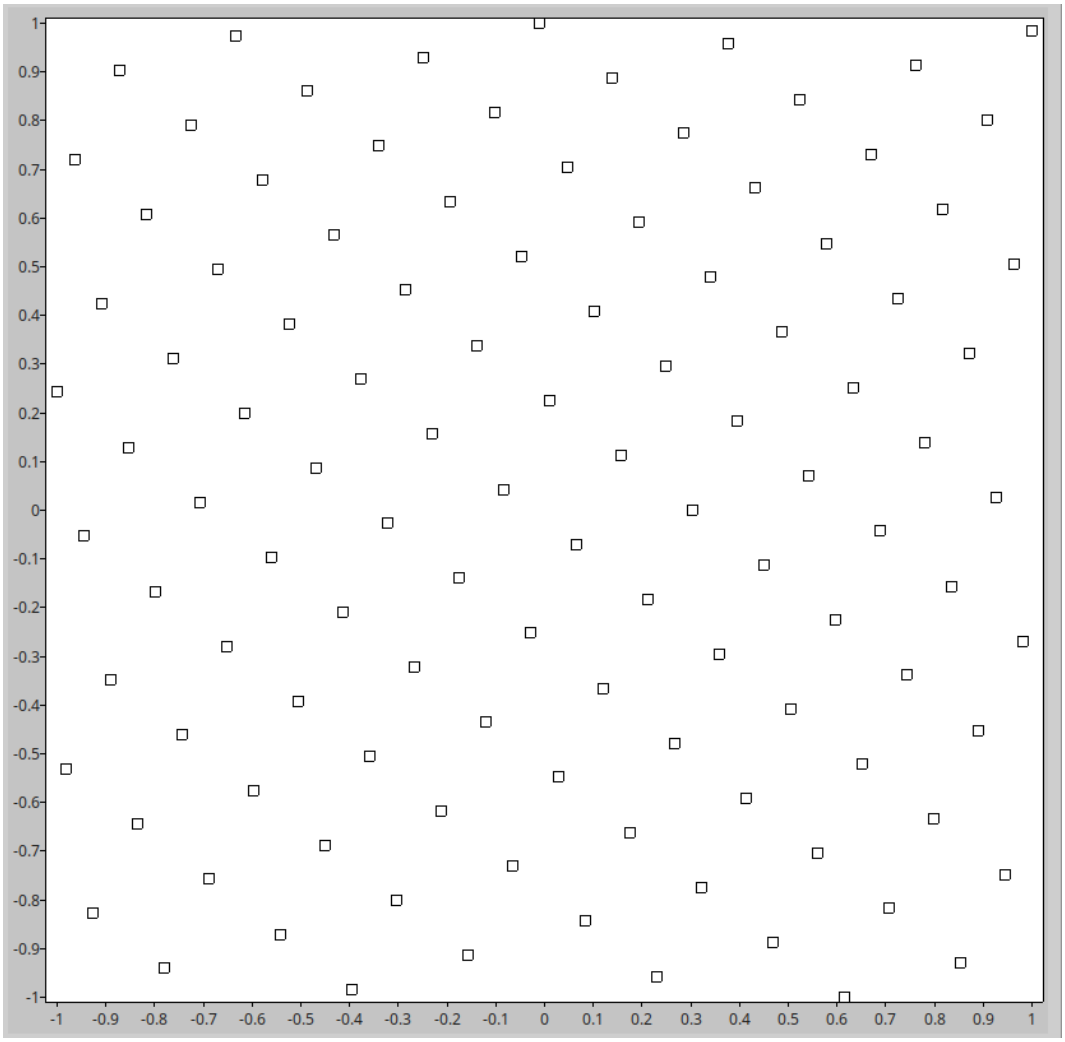

Figure 8. Position of the RBF centers (the traits) on the 2D manifold. The traits are positioned in a pseudo-regular way.

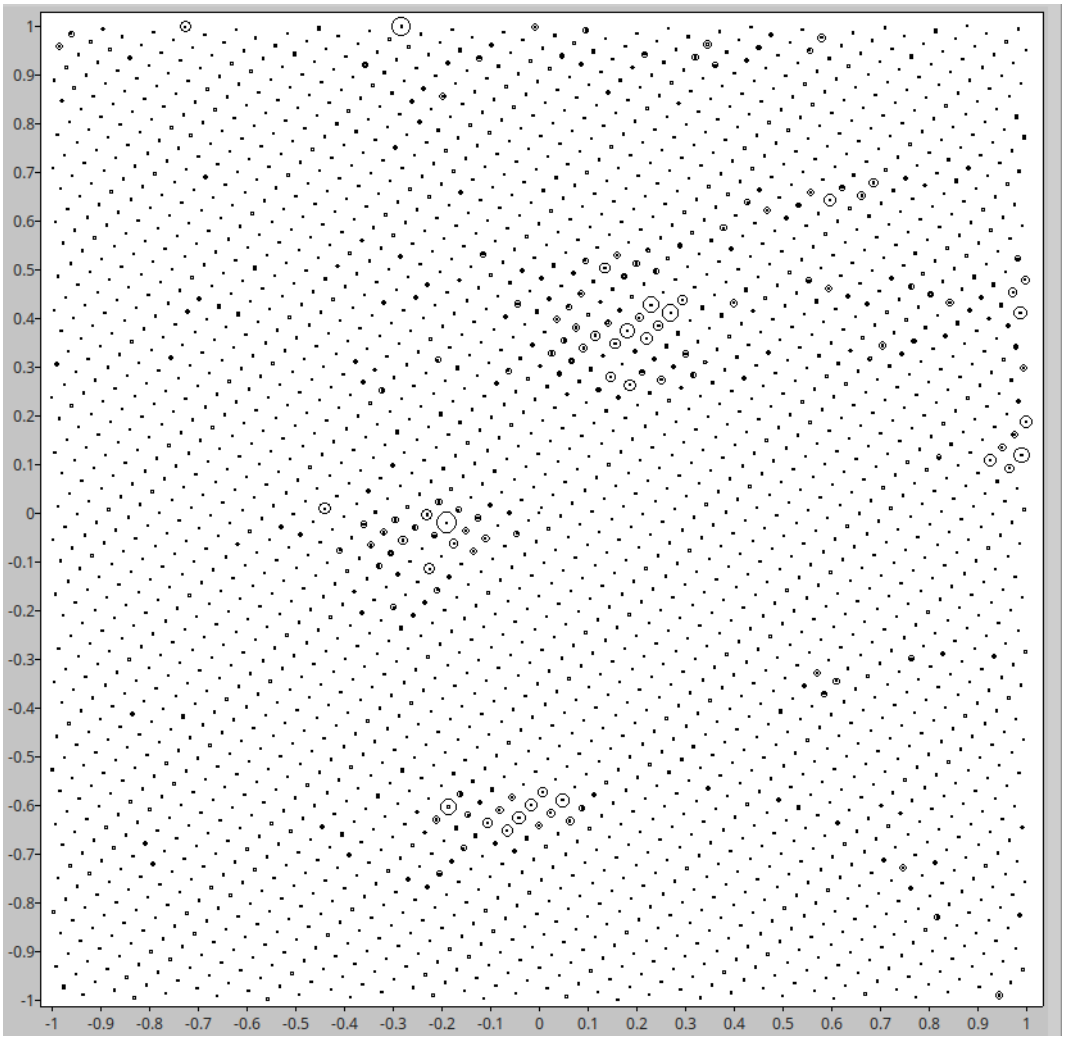

Figure 9. Positions of the sampling points of the manifold. These are the points were the density probability are estimated. The size of the circle around a sample point is proportional to the density of the chemical space region it is located in.

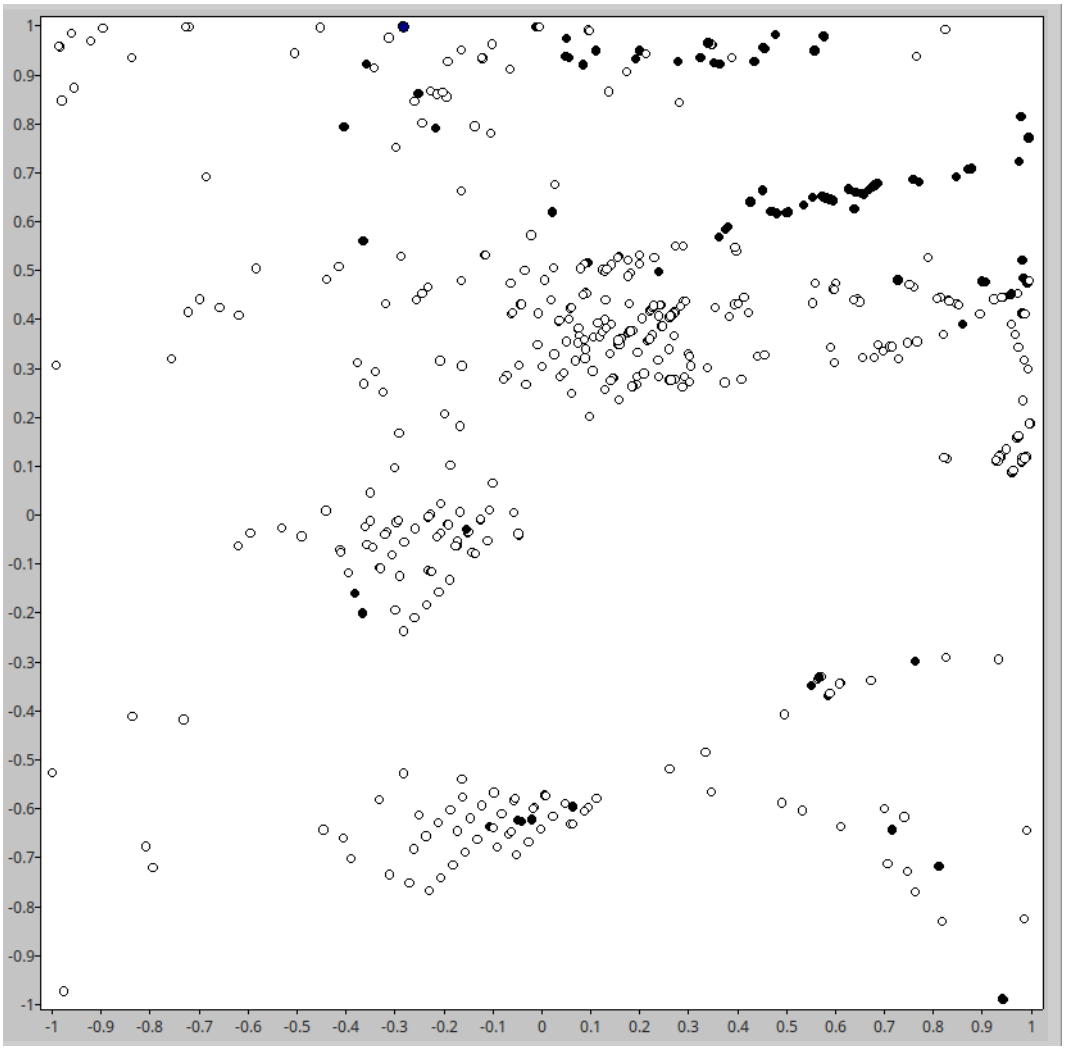

Figure 10. Projection of the training dataset on the GTM. Each point corresponds to a molecule. The black dots are those compounds associated to bioaccumulation.

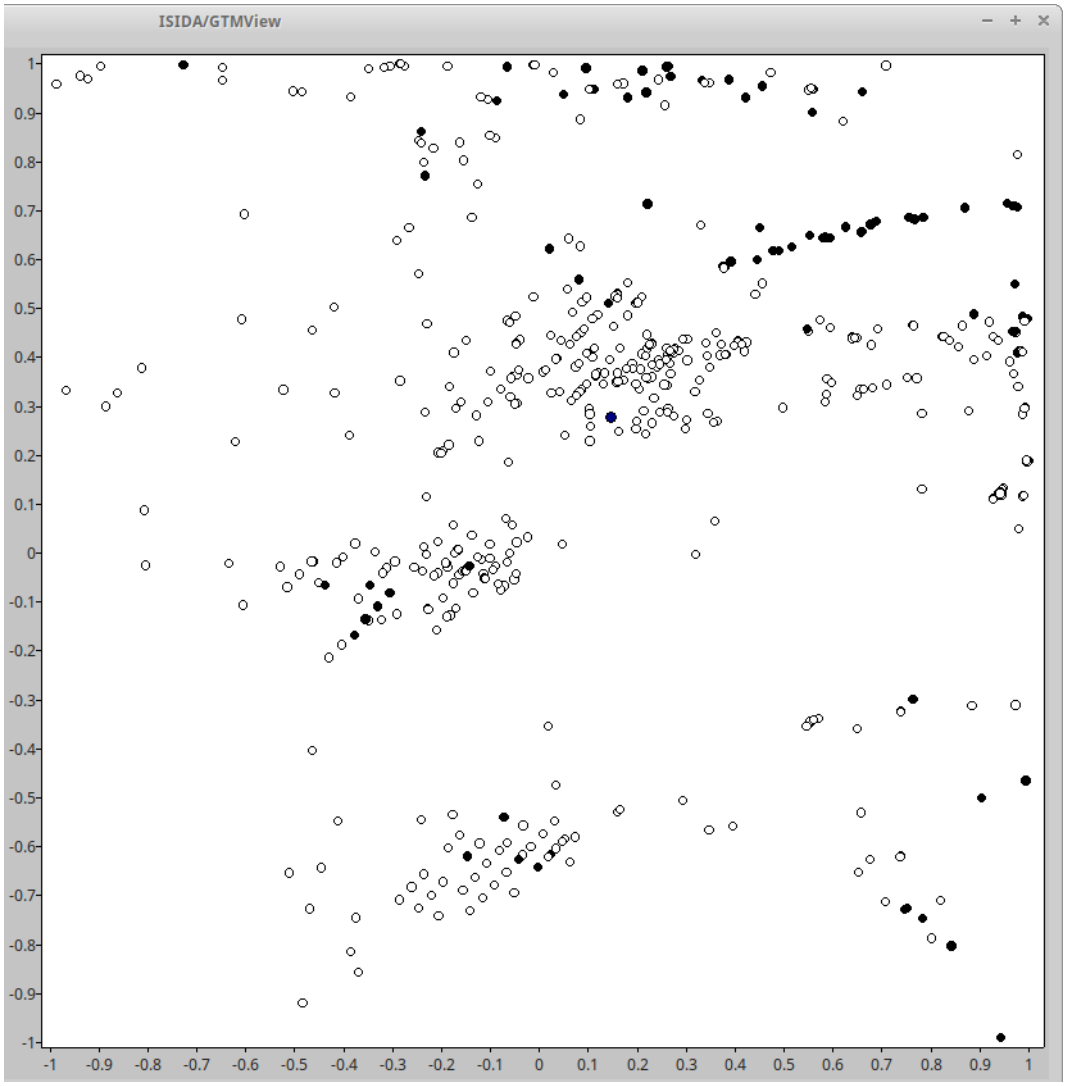

Figure 11. Projection of the test dataset on the GTM. Each point corresponds to a molecule. The black dots are those compounds associated to bioaccumulation..

1.3. Exercise 3. Increase the number of nodes

The aim of this exercise is to increase the resolution of the model to get a more precise picture of the chemical space.

Input:

- `train.xml`, `train.svm`
- `test.svm`, `test.sdf`

Output:

- `traink8000.xml`, `traink8000R.svm`, `traink8000Prj.mat`
- `testk8000R.svm`, `testk8000Prj.svm`

Instructions	***Comments***
Open the application **`xGTMReSample`**	The interface should look as illustrated in the Figure 12. Input management is located in (1). The new sampling is configured in (2). The log are written in (3) and the calculation are launched in (4).
Setup the input files to process (Figure 12, area 1).	This setup will load the GTM model build during the Exercise 1. The number of nodes

• Click the **GTM Model** button and chose the file `train.xml`. • Optionally, click the **Data file** button and chose the file `train.svm`. • Name the output as `traink8000.svm`. • Select the mode Pseudo-regular and set Nodes value to 8000. • Click the **OK** button.	of this model will be set to 8000, increasing the resolution of the latent space distribution. Using the **Rectangular** mode, the nodes are organized on a rectangular grid rather than distributed in pseudo-regular way. In that case the geometry of the grid must be specified. This feature can be convenient for manipulating the data with other plotting tools. The new model can be immediately applied to a dataset, here to the training set, to obtain responsibilities and projections.
Use the **xGTMapTool** software (Figure 1).	The improved resolution model will be used to project the test set data.
Select the **Use model** mode. • As Input select the file `test.svm`. • Name the **Output** as `testk8000` • Chose the file `traink8000.xml` as **Model (XML)**. Click the **OK** button.	With this setup the 8000 node GTM is used to compute the projections and responsibilities of the test set compounds. The former are saved in the file `testk8000Prj.mat` and the later in the file `testk8000R.mat`. The new estimated likelihood value should be close to the one obtained during the exercise 1, about -205.25. The value of 8000 has been selected as being small enough to speed up calculations and being illustrative of the improved resolution.
Use the **xGTMView** software (Figure 7).	This software is then used to monitor the changes in the GTM analysis of the test data.
Setup the interface so that: • The **GTM Model (XML format)** is the file `traink8000.xml` • The **Projection coordinates (MAT format)** is the file `testk8000Prj.mat` • The **Responsibility file (SVM format)** is the file `testk8000R.svm` • The **Molecular structure file (SDF format)** is the file `test.sdf` • Push the slide bar to the value 5 (Figure 7, area 4) • Click the button OK At this stage, the data are loaded. • Tick the **Projections box** (Figure 7, area 4)	The figure is very similar to the one initially obtained (Figure 13). However each compound is better localized and there are less overlapping points.

• Click on the SDF field selector (Figure 7, area 2) and select the field `BCFclass`.	

Conclusion

In this exercise, the number of nodes used to sample the GTM manifold is modified. The new model is used to project the test set. The effect is then monitored.

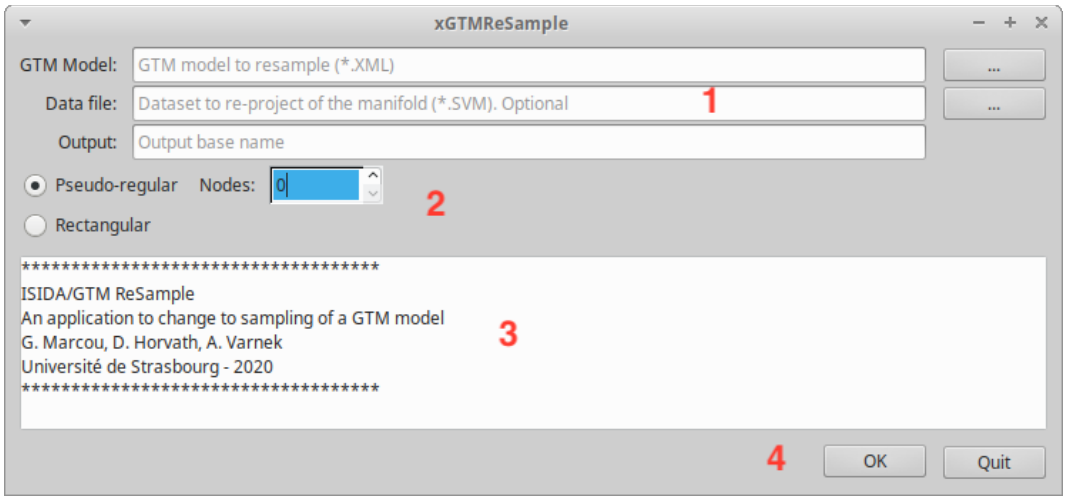

Figure 12: Interface of the xGTMReSample software. Input management is located in (1). The new sampling is configured in (2). The log are written in (3) and the calculation are launched in (4).

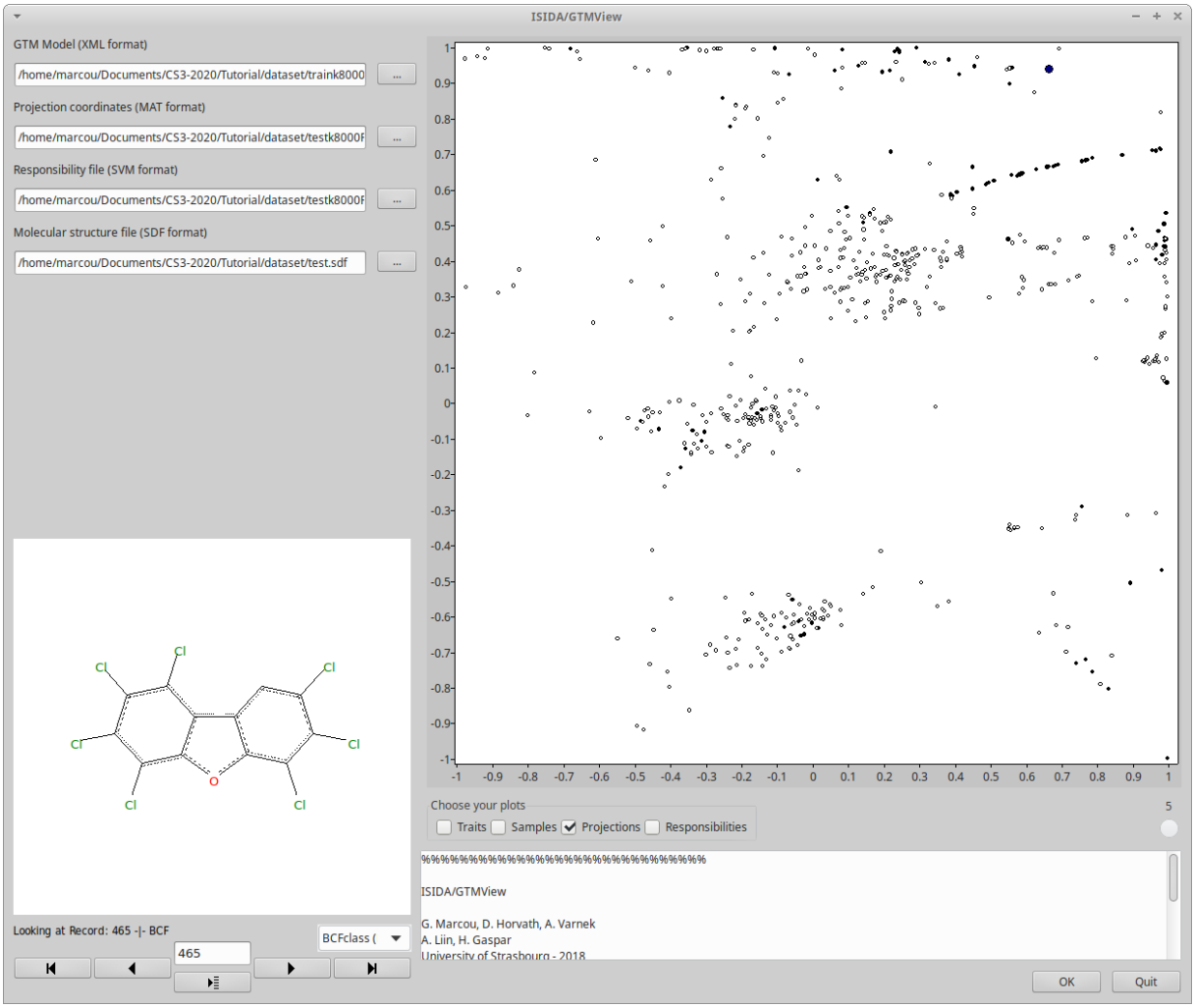

Figure 13: State of the xGTMView software visualizing the test set using a GTM using 8000 nodes.

1.4. Exercise 4. Classification models

This exercise will use the GTM to build a classification model discriminating compounds that are not bioaccumulating (labeled "notBCF") from those that are bioaccumulating (labeled "BCF").

Input:

- `train.xml`, `trainR.svm`, `trainBCFcl.prp`
- `testR.svm`, `testBCFcl.prp`

Output:

- `trainBCFcl_cls.xml`, `trainBCFcl_clsDens.mat`, `trainBCFcl_clsLS.mat`, `trainBCFcl*tsv`
- `testBCFcl*tsv`, `testBCFcl_clsDens.mat`, `testBCFcl_clsLS.mat`

The interface and approach is homologous to the later regression exercise. The instructions are therefore almost identical, so the two exercises are independent and self-consistent. It is Yet, both exercises 4 and 5 are required for the next exercise on visualization of the models.

Instructions	***Comments***
Open the application **`xGTMClass.`**	The interface should look as illustrated in the Figure 14. Input management is located in (1). The meta-data are configured in (2),

	they allow to define the author of the classification model, give a title, add comments and normalize classes. The region (3) give controls over the applicability domain. The region (4) allows to select between training a model, apply a model or perform a cross-validation. The model performances can be evaluated if the property file is provided in area (1). The logs generated during the calculations are recorded in the region (5) and the calculations are launched in the region (6).
Check that the software is configured on **Train** mode (Figure 14, area 4). Setup the input files to process (Figure 14, area 1). • Click the **GTM file** button and chose the file `train.xml`. • Click the **Responsibilities file** button and chose the file `trainR.svm`. • Click the **Property file** button and chose the file `trainBCFcl.prp`. • Name the output as `trainBCFcl_cls`.	The aim of this setup is to train a classification model based on the GTM. To this end, the `train.xml` file is loaded, providing information on the geometry of the manifold and details on the probability density modelled by the GTM. Then, information from the training set is loaded. Each compound is described by a vector of responsibilities; the whole training set is stored in the file `trainR.svm`. The property values associated to each compound, in the same order, are provided through the file `trainBCFcl.prp`. The first line of this file is always a comment, and is interpreted as a description of the property. Note that the software looks in the working directory to propose relevant input and output file names. Of course, it is possible to generate the models using the high resolution manifold, using as input the files `traink8000.xml` and `traink8000R.svm` as GTM and responsibilities files, respectively. However, it will not improve the predictive performances of the models
Complete the following fields: - **Title**: "Bioaccumulation Factor class"; - **Author**: enter your name; - **Comments**: "doesn't bioaccumulate (notBCF) is logBCF<=3.3 else does bioaccumulate (BCF), *today's date*"; - **Property name**: "BCFcl".	The meta-data are stored in the XML files of the model. They should not be disregarded for management of the models. The property name is a short name that can be used for referencing the model in an external system. Below, the tick-box **Normalize classes** applies a correction to the classes estimates to compensate for class imbalance. In the

	current situation, it is relevant, but for this exercise, it is not used.
Set the **Min. Responsibility**, **Min. Density** values to 0 and **Prevalent class ratio** to 1 (Figure 14, area 3).	The minimal responsibility is a threshold applied to the responsibility of each molecule on each node. If this value is below or equal the threshold, it is ignored in the preparation of the model. The minimal density is a threshold applied to each node, when using the model. If the density of the training set on a node is below or equal to the threshold, the node is associated to an out of applicability domain label. This means that the responsibilities of a compound on such node will contribute to an "OutOfAD" score and if this score dominates the class scores, then the compound is considered out of applicability domain. With this setup, the applicability domain is neutralized, except for those compounds covered by empty nodes (density equal to 0) for which the model cannot compute a prediction. This is because the concept of applicability domain is out of the scope of this tutorial. The Prevalent class ratio compute the ratio of the largest computed class probability over any other, for a given compound. Using this setting, if this ratio is 1, this means that at least two classes are equiprobable and no decision can be taken: the compound is out of applicability domain. The value can be decreased for a more stringent applicability domain. The current setting is the most neutral.
The setup should look like on Figure 15. Click the **OK** button (Figure 14, area 6)	This starts the calculations. They take are very fast since, they only need to accumulate the contributions of each compounds to each class. A message appears in the log to summarize the calculations. Three files are also created: - `trainBCFcl_cls.xml`: this file records the classification model - `trainBCFcl_clsDens.mat`: this is a matrix file that stores the *x* and *y* coordinates of each node and the density value on the node of the training data

	- `trainBCFcl_clsLS.mat`: this is a matrix file that stores the *x* and *y* coordinates of each node and the score for each class in the same order as they appear in the `trainBCFcl_cls.xml` file value on the node of the training data except the out of applicability domain which is always the last column. In this situation the order is `notBCF`, `BCF`, `OutOfAD`.
Click the **`Cross-validate`** radio-button. Check that: • The **`GTM file`** is the file `train.xml` • The **`Responsibilities file`** is the file `trainR.svm` • The **`Property file`** is the file `trainBCFcl.prp` • The **`Output`** is set to `trainBCFcl_cls_CV`. • The number of folds is set to the value 10. The setup should look as in Figure 16. Click the button **OK**.	This operation generates a cross-validation procedure to evaluate the predictive property of the map. The compounds are divided in 10 non-overlapping subsets, each being recursively allocated as a test set while the others are merged into a training set. The manifold is not modified, only the content of the dataset used for training the landscape and for estimating predictive performances are changing. The models and predictions generated at each fold are saved using the **`Output`** as base name. On average across folds, the reported balanced accuracy should be 0.80 (Figure 17).
Click the **`Predict`** radio-button. Check that: • The **`Landscape model file`** is the file `trainBCFcl_cls.xml` • The **`Responsibilities file`** is the file `trainR.svm` • The **`Property file`** is the file `trainBCFcl.prp` • The **`Output`** is set to `trainBCFcl_cls_pred`. The setup should look as in Figure 18. Click the button **OK**.	This setup applies uses the landscape to estimate the BCF class to the training data. Thus the performances are overestimated, with a balanced accuracy value of 0.89. (Figure 19).
Click the **`Predict`** radio-button. Check that: • The **`Landscape model file`** is the file `trainBCFcl_cls.xml` • The **`Responsibilities file`** is the file `testR.svm`	This setup applies uses the landscape to estimate the BCF class to the test data. The performances are comparable to those observed in cross-validation (Figure 21). If the **`Properties file`** field in not filled, the model is applied and the software returns the predicted classes in the `*.tsv`

• The **Property file** is the file `testBCFcl.prp` • The **Output** is set to `testBCFcl_cls_pred`. The setup should look as in Figure 20. Click the button **OK**.	file. But the predictive performances cannot be estimated.

Conclusion

In this exercise, the GTM is used to prepare an activity landscape discriminating the bioconcentrating (BCF) from non-bioconcentrating (notBCF) chemical species. The performances of this classification model are estimated on the training set, in cross-validation and on an external test set. The performances are comparable to the state of the art models (cross-validated balanced accuracy 0.84) regarding this property.

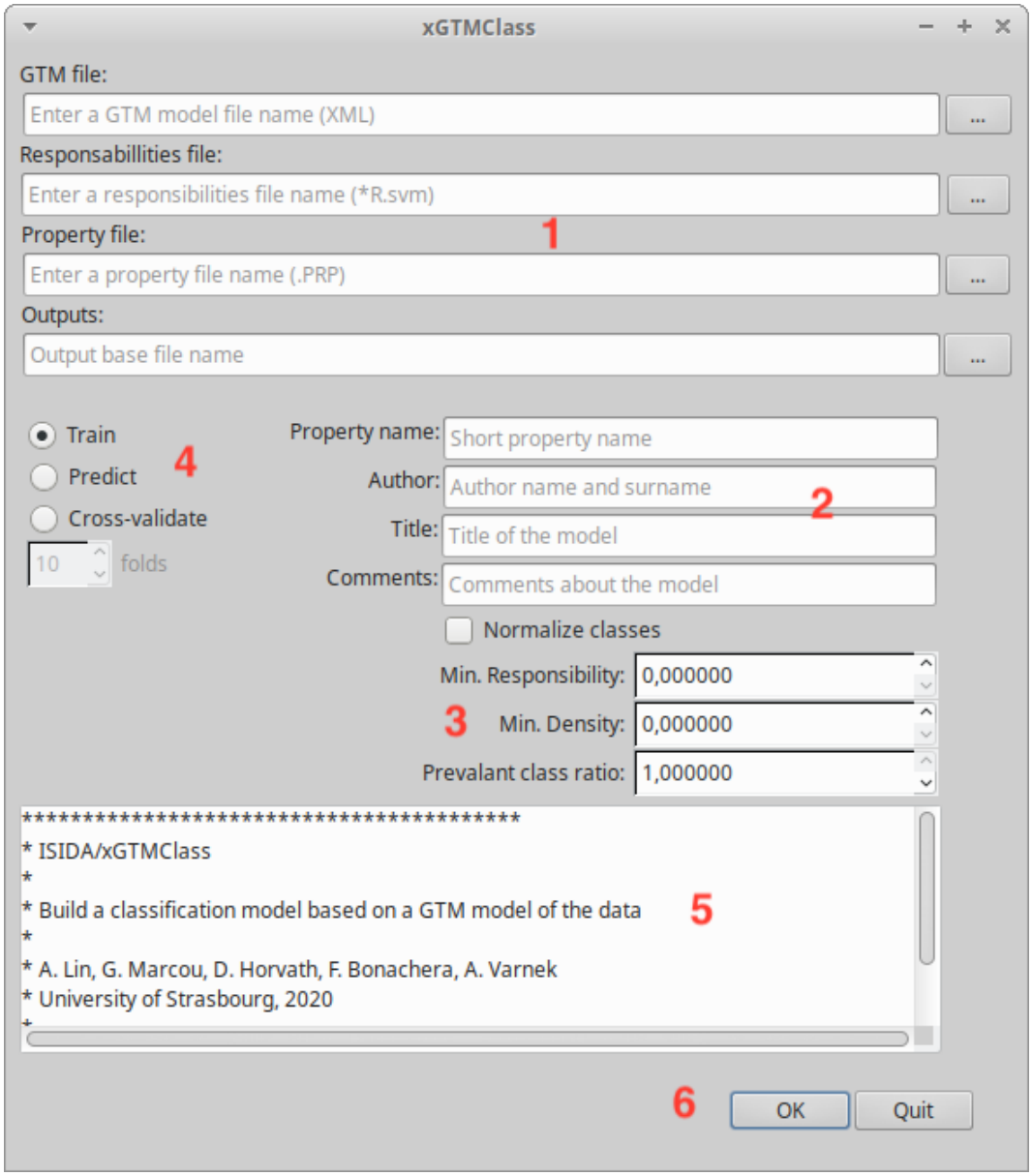

Figure 14: Interface of the xGTMClass software. The inputs and outputs are setup in area 1. The meta-data are provided in aread 2. The control of applicability domain is provided through the region 3. The model building and validation is chosen in area 4. The log are written in 5 and the calculations are launch with buttons in region 6.

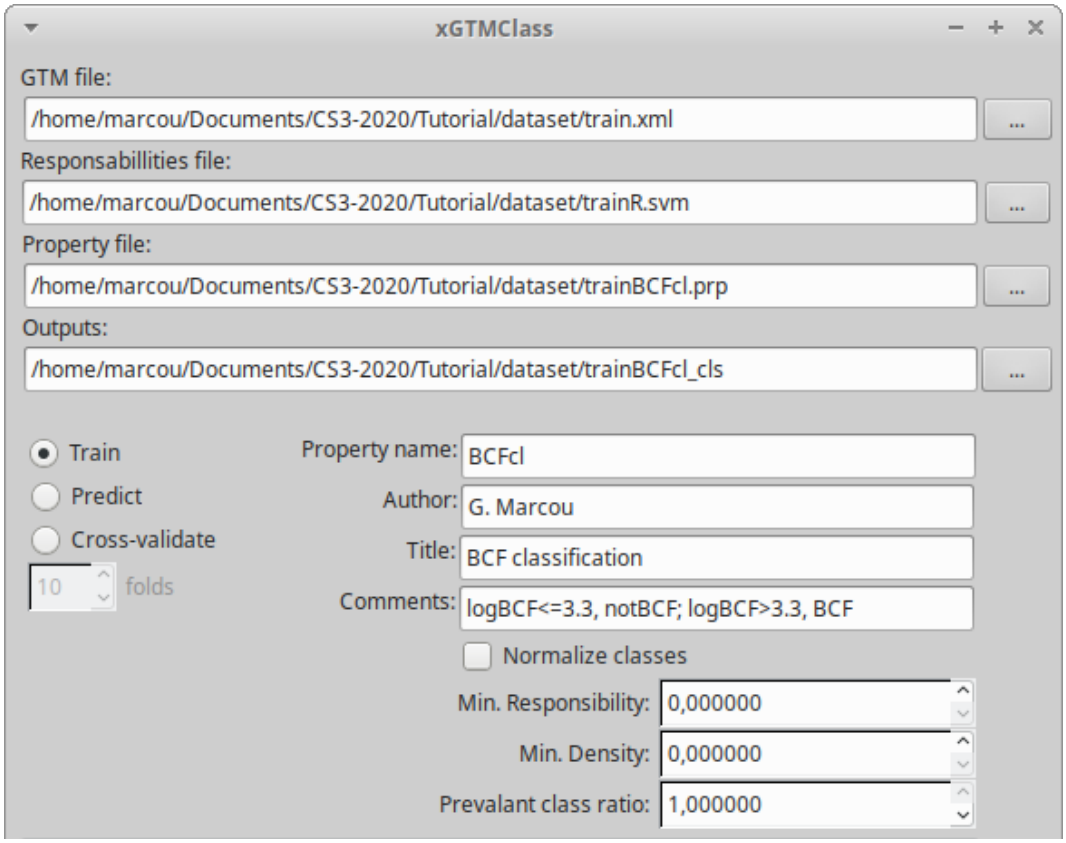

Figure 15: Preparation of a classification model discriminating bioaccumulating from non-bioaccumulating compounds.

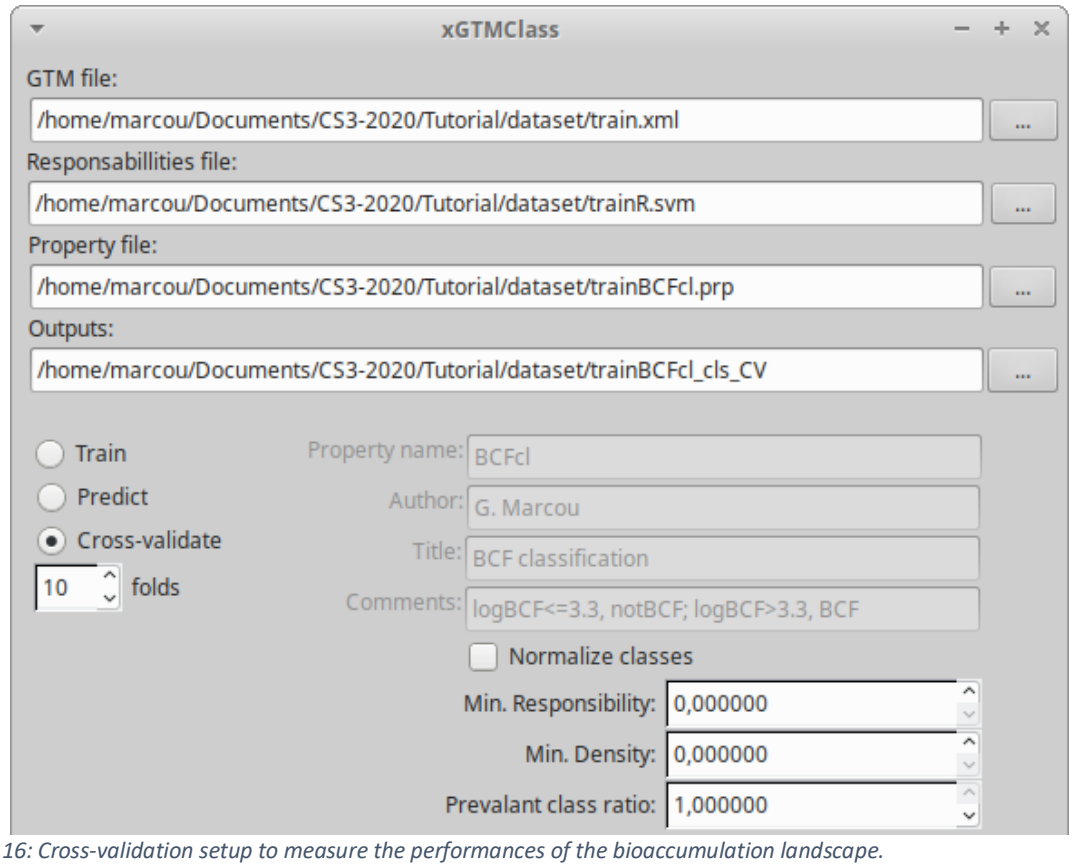

Figure 16: Cross-validation setup to measure the performances of the bioaccumulation landscape.

```
Processing responsibilities...
Number of molecules processed 567
Number of molecules used for landscape 567
---------------------------------------
Landscape saved in /home/marcou/Documents/CS3-2020/Tutorial/dataset/
trainBCFcl_cls_CV_train-fold10.xml
WARNING: File /home/marcou/Documents/CS3-2020/Tutorial/dataset/
trainBCFcl_cls_CV_test-fold10.tsv is overwritten.
---------------------------------------
Number of molecules processed: 65
Number of molecules in AD: 50
Number of molecules out AD: 15
---------------------------------------
*---------- fold 10 ----------*
Accuracy        =0.8200
Balanced Accuracy =0.6250
********** Averaged performances **********
Accuracy        =0.8858
Balanced Accuracy =0.8003
..... Calculations complete .....
```

Figure 17: Performances measures on the last fold and average performance accross all folds of the BCF landscape.

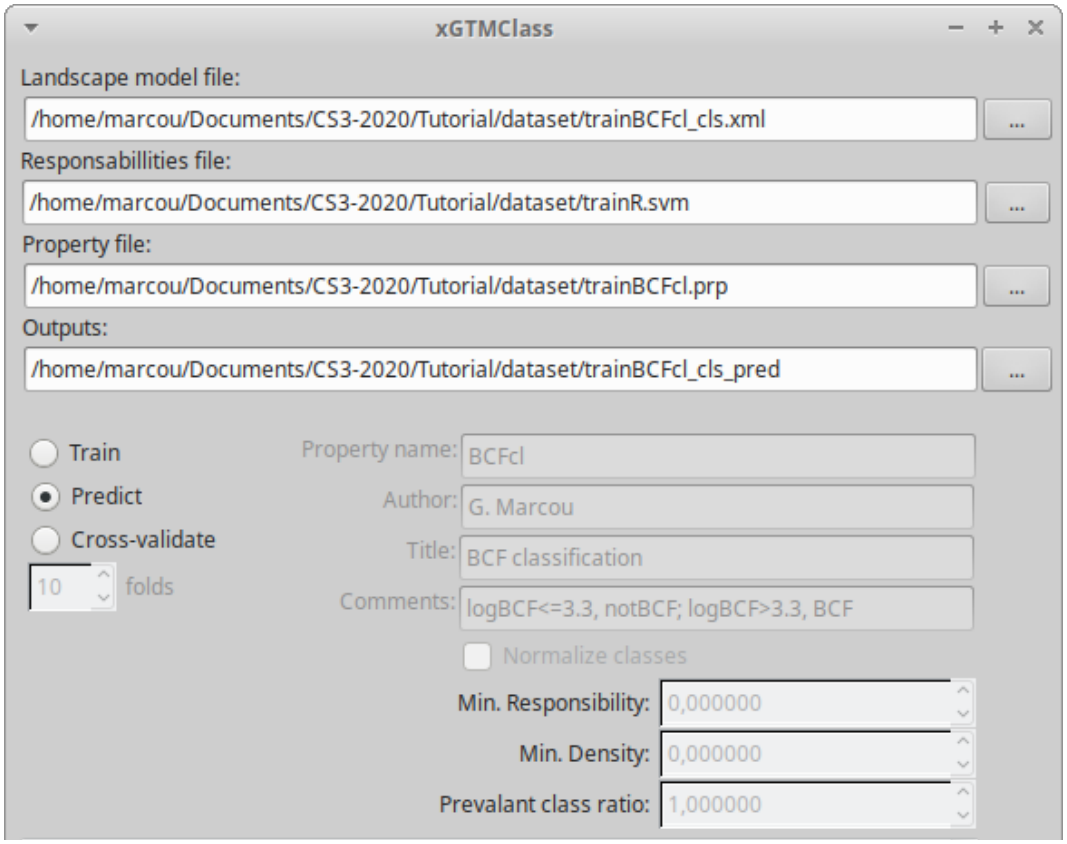

Figure 18: Configuration of the xGTMClass software to apply the model to training set data and estimate the performances.

```
/------------ Performances ------------\
Accuracy         =0.9525
Balanced Accuracy =0.8869
Precision(notBCF)=0.9514
Recall(notBCF)   =0.9922
F(notBCF)        =0.9714
MCC(notBCF)      =0.8392
Precision(BCF)=0.9588
Recall(BCF)   =0.7815
F(BCF)        =0.8611
MCC(BCF)      =0.8392
.............. Calculations complete ..............
```

Figure 19: Performances of the BCF classification landscape on the training dataset.

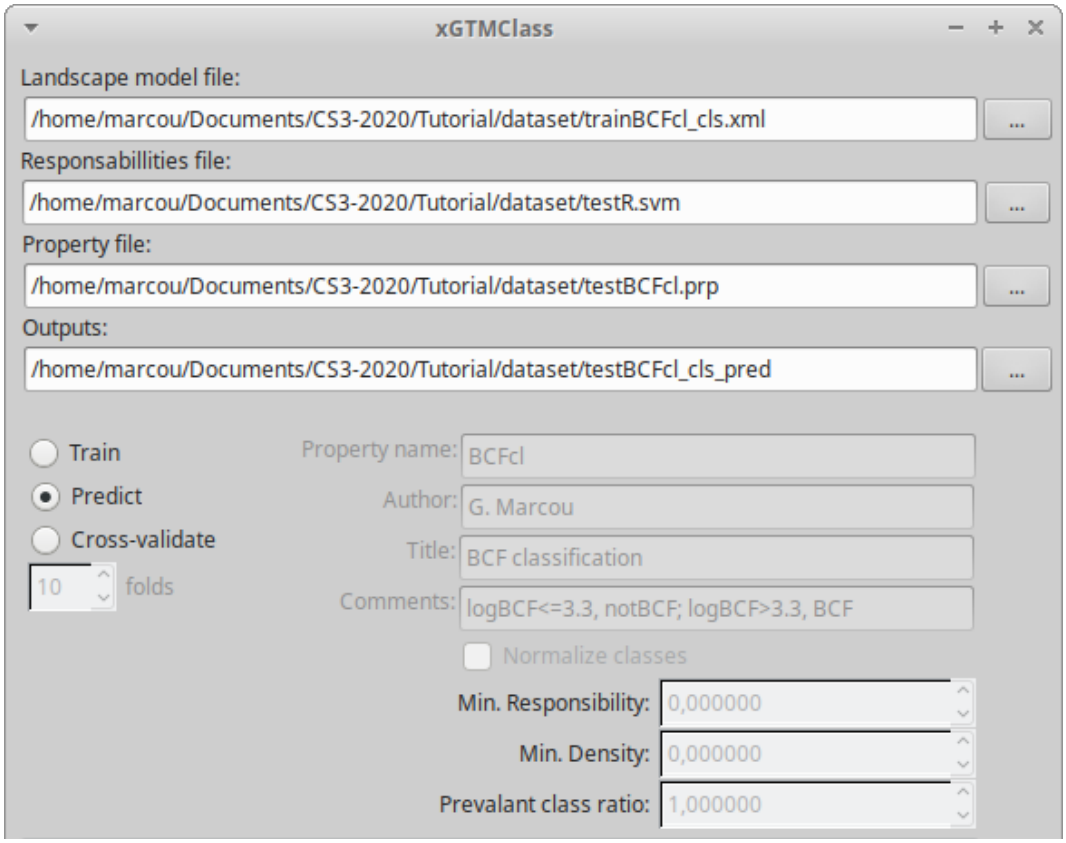

Figure 20: Configuration of the xGTMClass software to apply the model to test set data and estimate the performances.

```
/------------ Performances ------------\
Accuracy         =0.8670
Balanced Accuracy =0.7474
Precision(notBCF)=0.9073
Recall(notBCF)   =0.9335
F(notBCF)        =0.9202
MCC(notBCF)      =0.5237
Precision(BCF)=0.6471
Recall(BCF)   =0.5612
F(BCF)        =0.6011
MCC(BCF)      =0.5237
.............. Calculations complete ..............
```

Figure 21: Performances of the BCF classification landscape on the test dataset.

1.5. Exercise 5. Regression models

This exercise will use the GTM to build a regression model to predict the logarithm value the bioaccumulation factor expressed in L/Kg, given the chemical structure of new compounds.
Input:

- `train.xml`, `trainR.svm`, `trainBCFlog.prp`
- `testR.svm`, `testBCFlog.prp`

Output:

- `trainBCFlog_reg.xml`, trainBCFlog_regDens.mat, trainBCFlog_regLS.mat, `trainBCFlog*tsv`
- `testBCFlog*tsv`, `testBCFlog_regDens.mat`, `testBCFlog_regLS.mat`

The interface and approach is homologous to the later regression exercise. The instructions are therefore almost identical, so the two exercises are independent and self-consistent. It is Yet, both exercises 4 and 5 are required for the next exercise on visualization of the models.

Instructions	***Comments***
Open the application **`xGTMReg.`**	The interface should look as illustrated in the Figure 14. Input management is located in (1). The meta-data are configured in (2), they allow to define the author of the classification model, give a title and add comments. The region (3) give controls over the applicability domain. The region (4) allows to select between training a model, apply a model or perform a cross-validation. The model performances can be evaluated if the property file is provided in area (1). The logs generated during the calculations are recorded in the region (5) and the calculations are launched in the region (6).
Check that the software is configured on **`Train`** mode (Figure 14, area 4). Setup the input files to process (Figure 14, area 1). • Click the **`GTM file`** button and chose the file `train.xml`. • Click the **`Responsibilities file`** button and chose the file `trainR.svm`. • Click the **`Property file`** button and chose the file `trainBCFlog.prp`. • Name the output as `trainBCFlog_reg.`	The aim of this setup is to train a regression model based on the GTM. To this end, the `train.xml` file is loaded, providing information on the geometry of the manifold and details on the probability density modelled by the GTM. Then, information from the training set is loaded. Each compound is described by the vector of responsibilities on each node, the whole training set is stored in the file `trainR.svm`. The value of the property associated to each compound, in the same order, are provided through the file `trainBCFlog.prp`. The first line of this file is always a comment and

	is interpreted as a description of the property. Note that the software looks in the working directory to propose relevant input and output file names. Of course, it is possible to generate the models using the high resolution manifold, using as input the files `traink8000.xml` and `traink8000R.svm` as GTM and responsibilities files, respectively.
Complete the following fields: - **`Title`**: "Logarithm Of Bioaccumulation Factor"; - **`Author`**: enter your name; - **`Comments`**: "Bioconcentration factor (in L/Kg), *today's date*"; - **`Property name`**: "BCFlog".	The metadata are stored in the XML files of the model. They should not be disregarded for management of the models. The property name is a short name that can be used for referencing the model in an external system.
Set the **`Min. Responsibility`** and **`Min. Density`** values to 0 (Figure 14, area 3).	The minimal responsibility is a threshold applied to the responsibility of each molecule on each node. If this value is below or equal the threshold, it is ignored in the preparation of the model. The minimal density is a threshold applied to each node, when using the model. If the density of the training set on a node is below or equal to the threshold, then the node is associated to an out of applicability domain label (OutAD). This means that the responsibilities of a compound on such node will contribute to an "OutAD" score and if this score is large, then the compound is considered out of applicability domain. With this setup, the applicability domain is neutralized, except for those compounds covered by empty nodes (density equal to 0) for which the model cannot compute a prediction. The concept of applicability domain is out of the scope of this tutorial.
The setup should look like on Figure 15. Click the **OK** button (Figure 14, area 6)	This starts the calculations. They are very fast since, they only need to accumulate the contributions of each compounds to each class. A message appears in the log to summarize the calculations. Three files are also created: - `trainBCFlog_reg.xml`: this file records the regression model

	- `trainBCFlog_regDens.mat`: this is a matrix file that stores the *x* and *y* coordinates of each node and the density value on the node of the training data - `trainBCFlog_regLS.mat`: this is a matrix file that stores the *x* and *y* coordinates of each node and the weighted average value the bioconcentration factor logarithm from compounds contributing to this node. The out of applicability domain is the last column.
Click the **`Cross-validate`** radio-button. Check that: • The **`GTM file`** is the file `train.xml` • The **`Responsibilities file`** is the file `trainR.svm` • The **`Property file`** is the file `trainBCFlog.prp` • The **`Output`** is set to `trainBCFlog_reg_CV`. • The number of folds is set to the value 10. The setup should look as in Figure 16. Click the button **OK**.	This operation generates a cross-validation procedure to evaluate the predictive property of the map. The compounds are divided in 10 non-overlapping subsets, each being recursively allocated as a test set while the others are merged into a training set. The manifold is not modified, only the content of the dataset used for training the landscape and for estimating predictive performances are changing. The models and predictions generated at each fold are saved using the **`Output`** as base name. On average across folds, the reported RMSE should be 1.1 (Figure 17). This value is fairly large, although the model is indicative of the trend as illustrated by the determination coefficient measured about 0.54.
Click the **`Predict`** radio-button. Check that: • The **`Landscape model file`** is the file `trainBCFlog_reg.xml` • The **`Responsibilities file`** is the file `trainR.svm` • The **`Property file`** is the file `trainBCFlog.prp` • The **`Output`** is set to `trainBCFlog_reg_pred`. The setup should look as in Figure 18. Click the button **OK**.	This setup applies the landscape to estimate the BCF logarithm to the training data. Thus the performances are overestimated, with a balanced accuracy value of 0.6. (Figure 19).
Click the **`Predict`** radio-button. Check that:	This setup applies uses the landscape to estimate the logarithm of BCF value to the test data. The performances are close to

• The **Landscape model file** is the file `train_cls.xml` • The **Responsibilities file** is the file `testR.svm` • The **Property file** is the file `testBCFlog.prp` • The **Output** is set to `testBCFlog_reg_pred`. The setup should look as in Figure 20. Click the button **OK**.	those observed in cross-validation (Figure 21). If the **Properties file** field in not filled, the model is applied and the software returns the predicted classes in the `*.tsv` file. But the predictive performances cannot be estimated.

Conclusion

In this exercise, the GTM is used to prepare a property landscape to predict the logarithm value of the bioaccumulation factor expressed in L/Kg. The performances on the test set are weak (RMSE on test set about 1.05) compared to published models (RMSE about 0.6). However, it is sufficient to provide with a trend that can be visually translated in landscapes as in the next exercise.

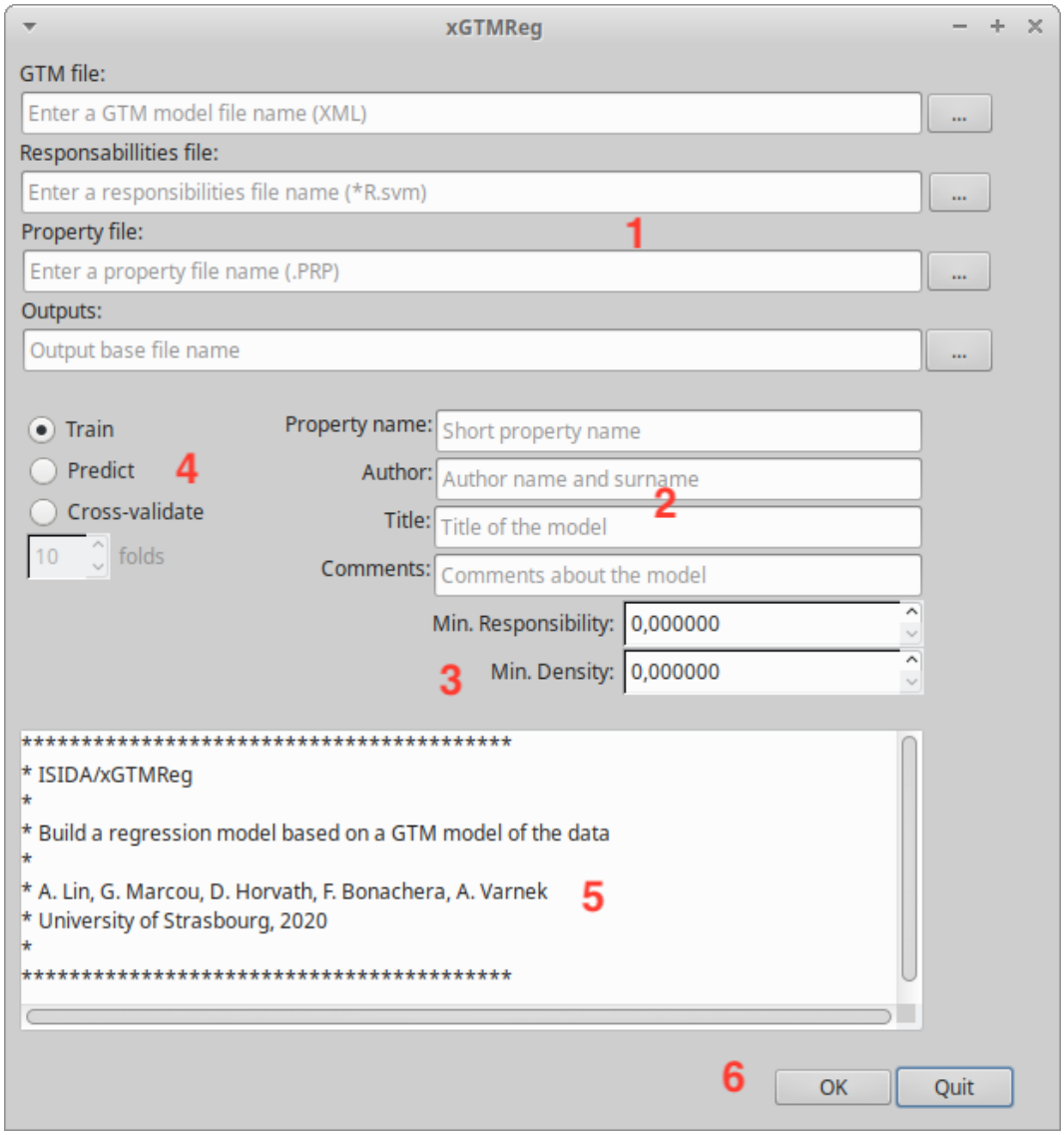

Figure 22: Interface of the xGTMReg software. The inputs and outputs are setup in area 1. The meta-data are provided in aread 2. The control of applicability domain is provided through the region 3. The model building and validation is chosen in area 4. The log are written in 5 and the calculations are launch with buttons in region 6.

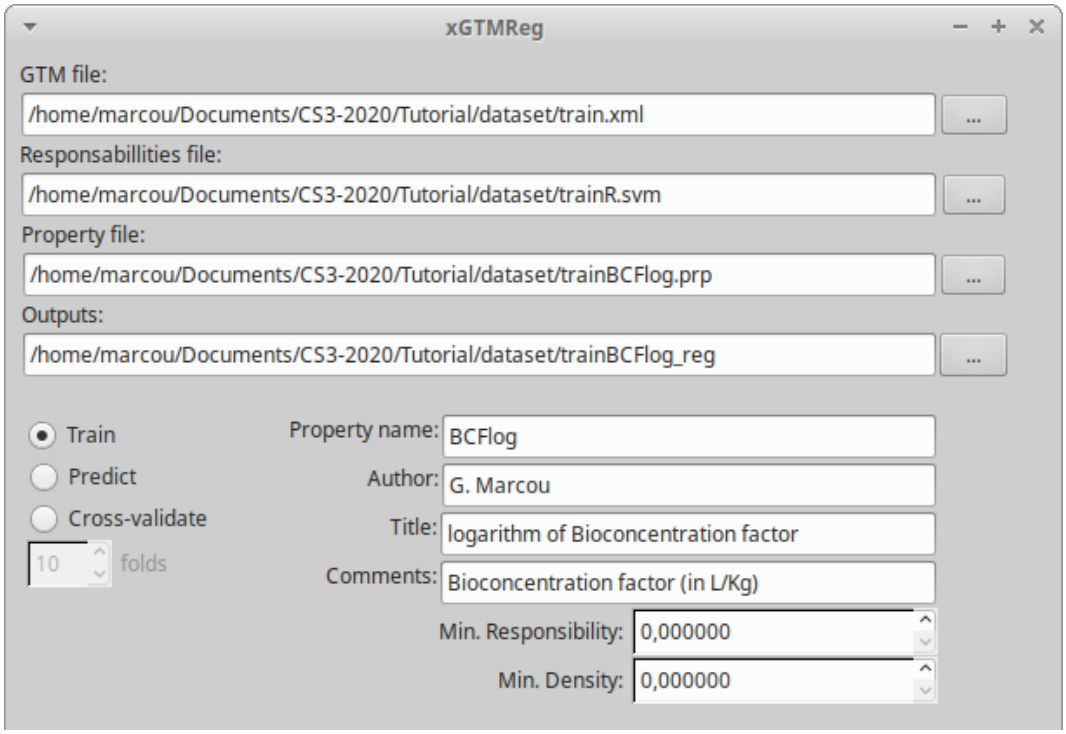

Figure 23: Preparation of a regression predicting the bioaccumulation factor logarithm.

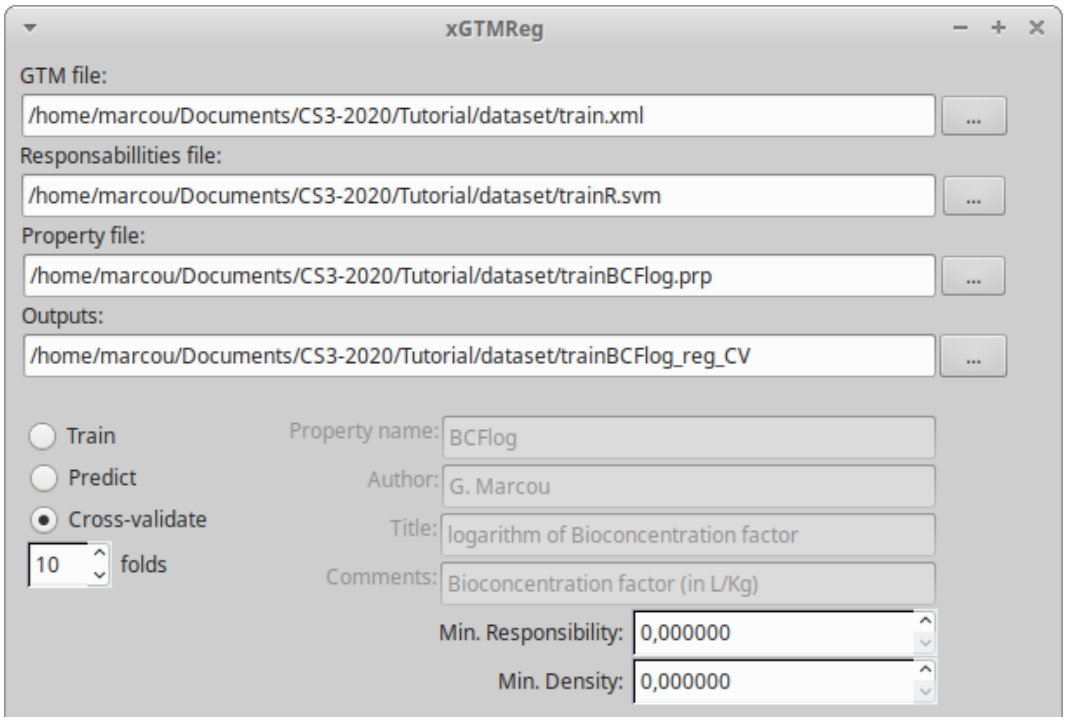

Figure 24: Cross-validation setup to measure the performances of the bioaccumulation factor logarithm landscape.

```
*------------ fold 10 ------------*
RMSE       =1.0181
MAE        =0.7570
Relative RMSE=0.8481
Relative MAE =0.7724
R2det      =0.4964
CCC        =0.2808
R2cor      =0.2941
************** Averaged performances **************
RMSE       =1.0997
MAE        =0.8461
Relative RMSE=0.8067
Relative MAE =0.7393
R2det      =0.5468
CCC        =0.3412
R2cor      =0.3760
.............. Calculations complete ..............
```

Figure 25: Performances measures on the last fold and average performance across all folds of the log(BCF) landscape.

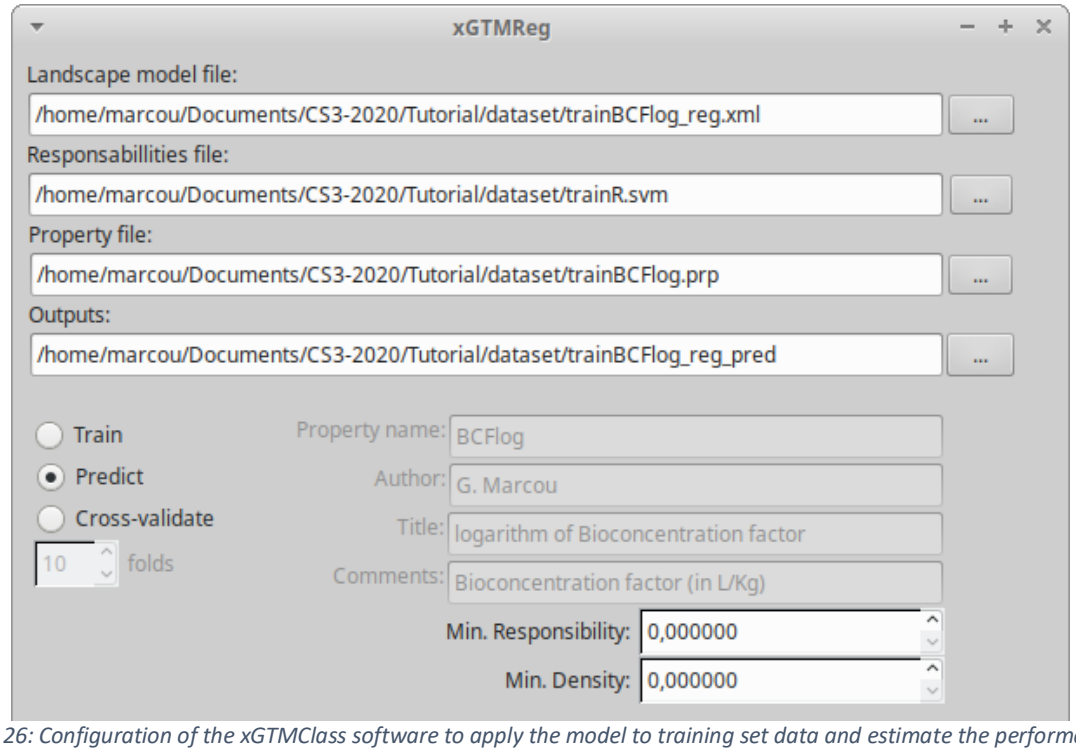

Figure 26: Configuration of the xGTMClass software to apply the model to training set data and estimate the performances.

```
----------------------------------------
/------------ Performances ------------\
RMSE       =0.5993
MAE        =0.3926
Relative RMSE=0.4380
Relative MAE =0.3436
R2det      =0.8082
CCC        =0.8921
R2cor      =0.8085
.............. Calculations complete ..............
```

Figure 27: Performances of the logarithm of BCF landscape on the training dataset.

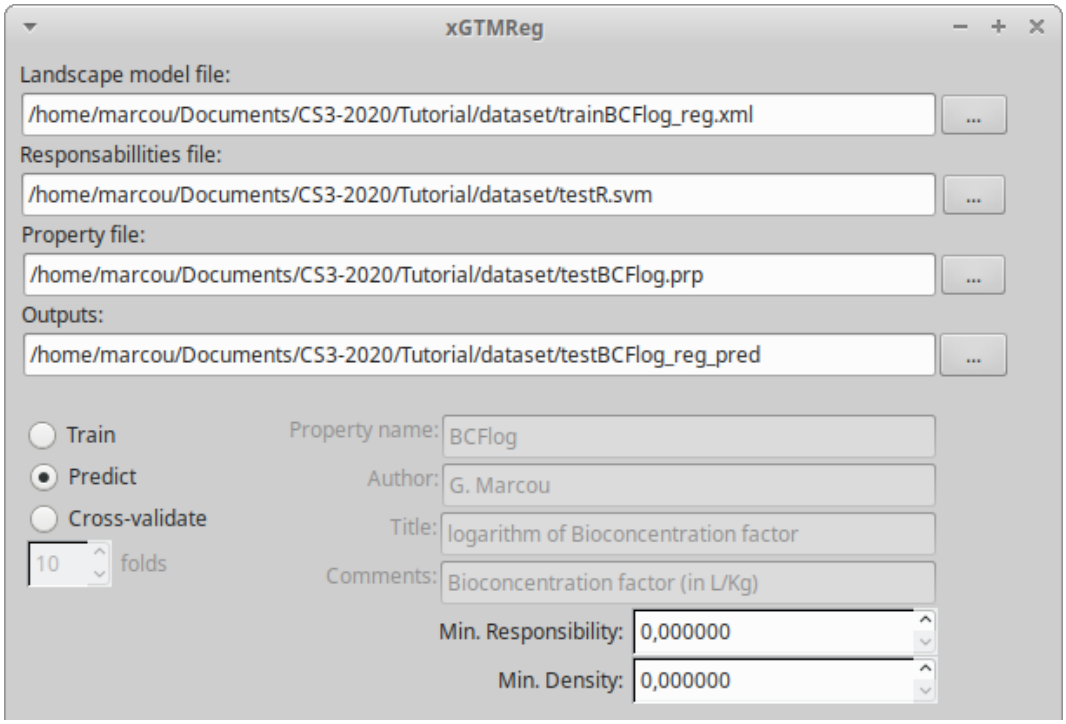

Figure 28: Configuration of the xGTMClass software to apply the model to test set data and estimate the performances.

```
/------------ Performances ------------\
RMSE       =1.0527
MAE        =0.8164
Relative RMSE=0.7624
Relative MAE =0.7018
R2det      =0.4188
CCC        =0.6346
R2cor      =0.4360
.............. Calculations complete ..............
```

Figure 29: Performances of the BCF classification landscape on the test dataset.

1.6. Exercise 6. Property and activity landscapes

This GTM can be used to build predictive models for classification and regression as illustrated in the previous exercises. These models can be transferred to the map, providing a z-axis used to color the map. When a quantitative estimation is transferred to the map, the result is termed a property landscape; when it is a score estimating the population of a class on the map, it is termed and activity landscape.

Input:

- `trainBCFlog_reg.xml`, `trainBCFcl_cls.xml`, `trainPrj.mat`, `train.sdf`
- `testPrj.mat`, `test.sdf`

Instructions	***Comments***
Open the application **`xGTMLandscape.`**	The interface should look as illustrated in the Figure 30. Input management is located in (1). The landscape is displayed in (2). The displayed activity or property, as well as some controls over the display are provided in region (3). The area (4) of the interface

	controls the navigation through the loaded chemical structures. The logs generated are recorded in the region (5) and the calculations are launched in the region (6).
Setup the input files to process (Figure 30, area 1 and Figure 31). • Click the **GTM landscape model file (.xml)** button and chose the file `trainBCFlog_reg.xml`. • Click the **Projection file (Prj.mat)** button and chose the file `trainPrj.mat`. • Click the **Chemical structures file (SDF)** button and chose the file `train.sdf`. Click the **OK** button (Figure 30, area 6).	This setup loads the property landscape representing the logarithm of the bioconcentration factor (`trainBCFlog_reg.xml`), the projections of the training dataset (`trainPrj.mat`) and the corresponding chemical structure file (`train.sdf`). After loading the projections of the chemical compounds are displayed and the chemical structures can be navigated (Figure 32). Besides, the property selector in the area 3 of Figure 30 is updated and becomes useable.
In the property selector, select the item **Density**.	The density map is displayed in gray scale (Figure 33), the dark regions being the most populated. The colors are located on the GTM nodes, so the maps based the 8000 nodes version of the map can be used to generated more resolved visuals.
In the property selector, select the item **BCFlog**.	The property map of the bioconcentration factor is displayed (Figure 34). The dark colored regions are those with the lower logBCF value. The white areas are unpopulated regions and are out of applicability domain. Therefore, they are not drawn.
Setup the input files to process (Figure 30, area 1 and Figure 35). • Click the **GTM landscape model file (.xml)** button and chose the file `trainBCFcl_cls.xml`. • Click the **Projection file (Prj.mat)** button and chose the file `trainPrj.mat`. • Click the **Chemical structures file (SDF)** button and chose the file `train.sdf`. Click the **OK** button (Figure 30, area 6).	This setup loads the activity landscape locating the regions of the chemical space that are populated by compounds that are likey to bioconcentrating or not (`trainBCFcl_cls.xml`). After loading the projections of the chemical compounds are displayed and the chemical structures can be navigated (Figure 32). The property selector in the area 3 of Figure 30 is updated. It is more complicated because for activity landscapes, two marginal probabilities distribution per class can be plotted (termed in the interface as “Likelihood” and “Class”). Additionally, the applicability domain appears as an additional class.

In the property selector, select the item **Likelihood class=BCF**.	The activity landscape is displayed in blue scale (Figure 36), the dark regions are low probability density value and the more light regions are high probability density value. The map is obtained by summing up the responsibilities of compounds labeled as bioconcentrating. It is interpreted as a measure of the marginal probability of the nodes to be activated by bioconcentrating compounds. Here the most visible region are PCBs.
In the property selector, select the item **Likelihood class=notBCF**.	This map represents the activated nodes, but for the non-bioconcentrating chemical structures. This is the major class of the dataset, so it tends to cover a larger part of the map. They cover simple benzene derivatives, various thiophosphates and silicates (Figure 37).
In the property selector, select the item **Class class=BCF**. Then select the item **Class class=notBCF**.	This time, the landscape represents the probability of a compound located at a given node, to belong to the BCF class (Figure 38) or notBCF class (Figure 39). It is related to the "Likelihood" maps by a Bayes formula. The score population is also much more concentrated over extreme values. In contrast to the property landscape, the applicability domain appears as regions of zero probability (dark) for both BCF and notBCF classes.
In the property selector, select the item **Class class=OutOfAD**	The applicability domain is more visible when displayed using the item **Class class=OutOfAD** (Figure 40). Typically, there are no compounds from the training set that are considered as out of applicability domain. Therefore, the Likelihood landscape is flat and almost null. Therefore, each class is equiprobable on these regions of the map. However, the decision is to set the probability of both classes (BCF/notBCF) to 0 and the probability of OutOfAD to 1. Therefore, the Class landscape actually represents the null density regions of the map. This picture become more complicated when modifying the parameters of the model defining the applicability domain.

Conclusion

In this exercise, the regression and classification GTM models are visualized, leading to property and activity landscapes, respectively. The projections of the training set compounds are used interpret the map chemically. The applicability domain and density maps are main information from the landscape analysis. They locate those regions where the chemical space has not been explored yet concerning the bioconcentration property.
In the classification exercise, the classes BCF and notBCF are mutually exclusive. It translates on the map through the shape complementarity between the classes. This is true as long as the classes are not normalized, of course. Another aspect of activity landscapes is that they can represent two kinds of marginal probabilities: probability to of a node considering a class (Likelihood) and probability of class considering a node (Class). The former quantity is more convenient to compare the populations of classes, as for instance when comparing chemical libraries. The latter is better suited to illustrate the classification model. However, they ultimately are convertible one into the other through a Bayes formula so they basically encode the same information.

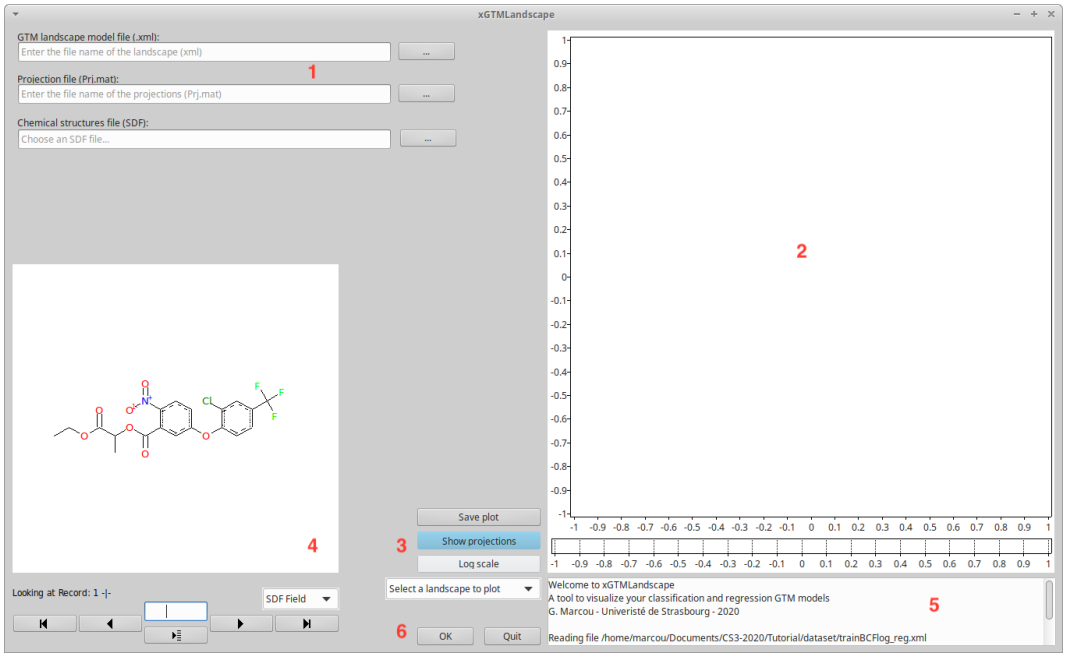

Figure 30: Interface of the xGTMLandscape software. The inputs and outputs are setup in area 1. The landscape is displayed in area 2. The choice of the landscape to display and some rendering control are provided through the region 3. The navigation through the chemical structures is located in area 4. The log are written in 5 and the calculations are launch with buttons in region 6.

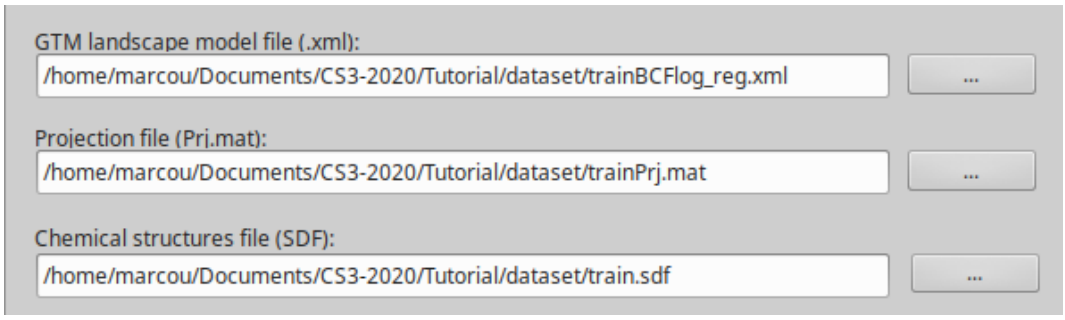

Figure 31: Preparation for loading the logarithm of bioconcentration factor property landscape.

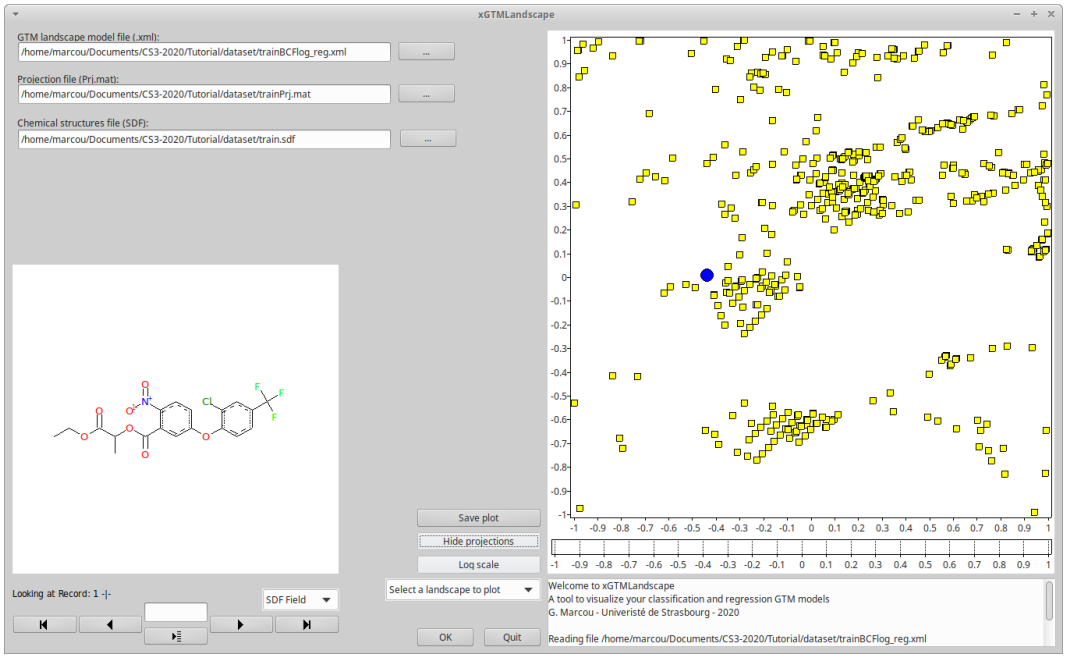

Figure 32: State of the xGTMLandscape software interface after loading a property or activity landscape.

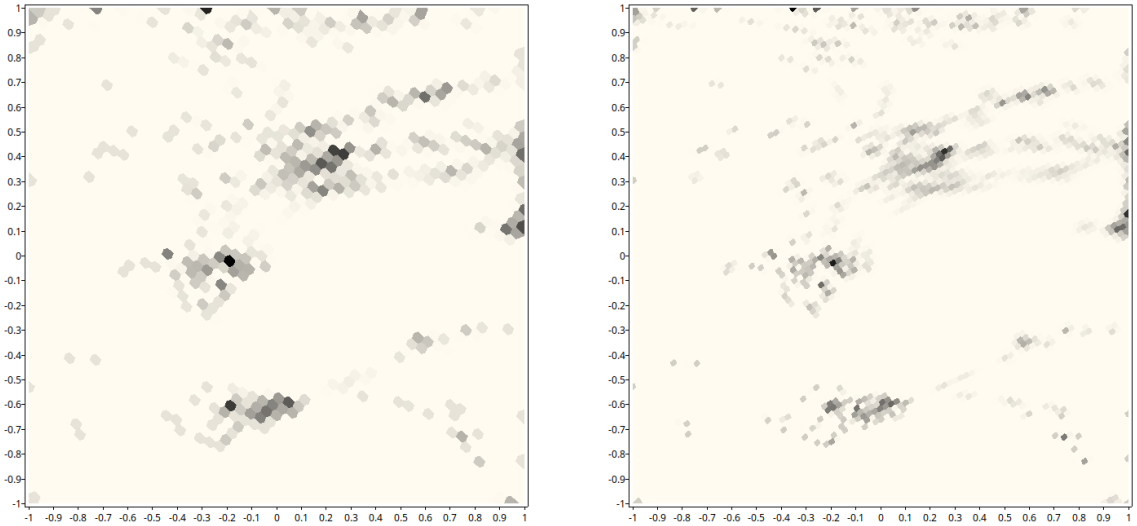

Figure 33: Density landscape of the training dataset using the standard map, or the improved map using 8000 nodes.

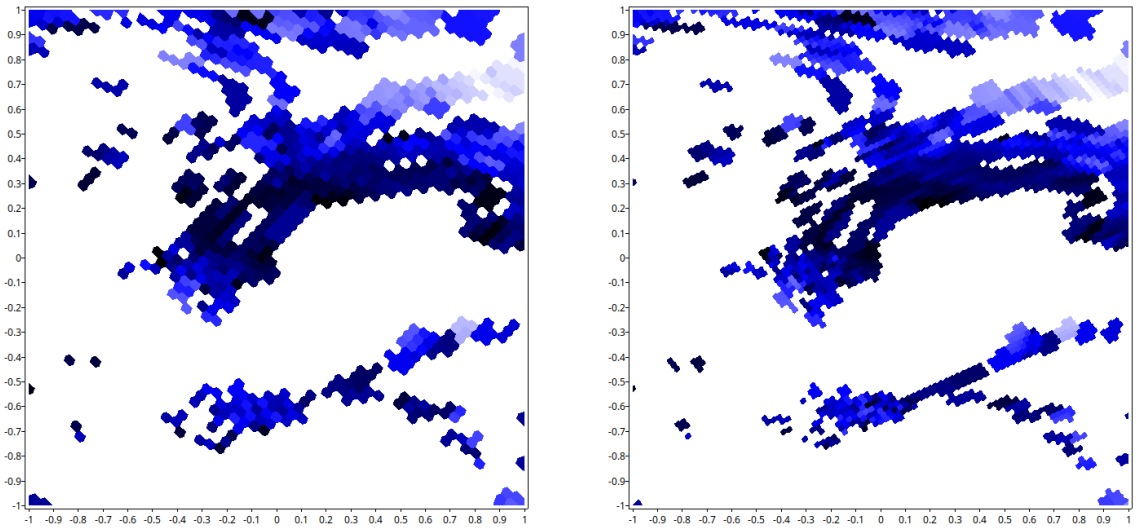

Figure 34: Property landscape of the training dataset of the logarithm of the bioconcentration factor, using the standard map, or the improved map using 8000 nodes.

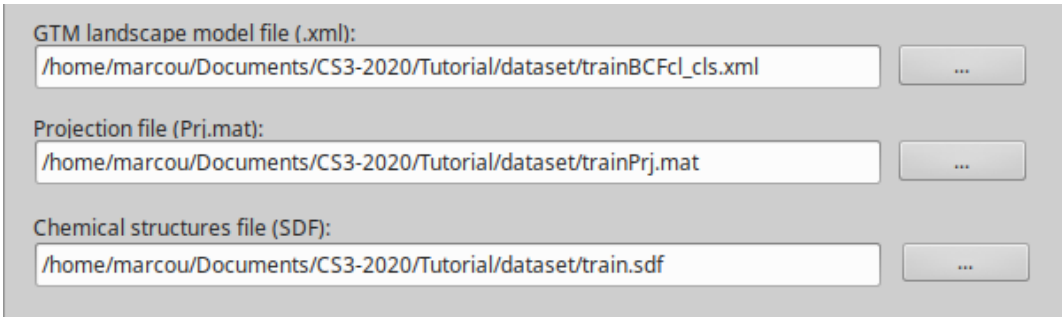

Figure 35: Preparation for loading the bioconcentration factor activity landscape.

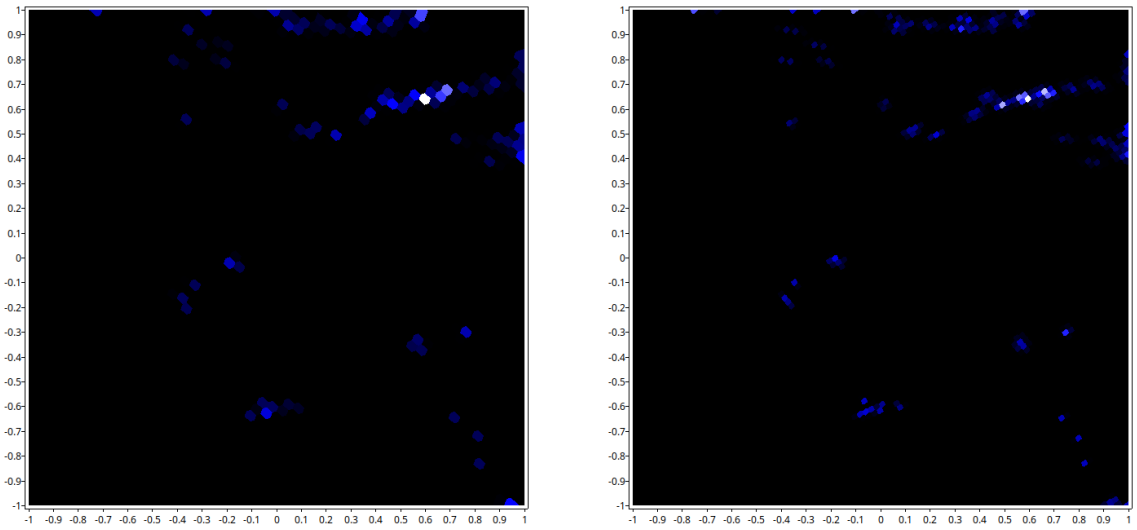

Figure 36: Activity landscape as the density of compounds labeled as bioconcentrating, using the standard map or the improved map using 8000 nodes.

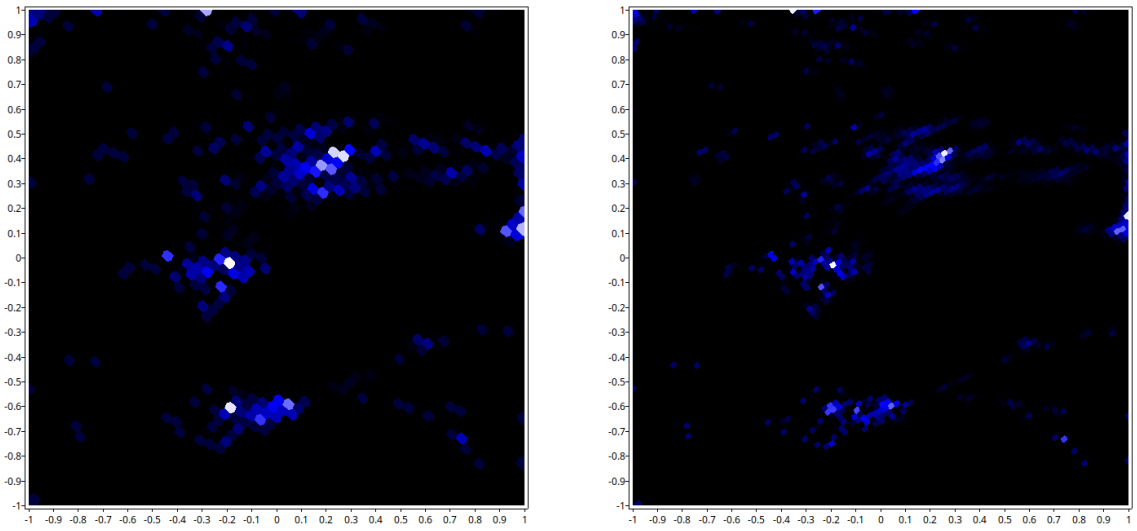

Figure 37: Activity landscape as the density of compounds labeled as not bioconcentrating, using the standard map or the improved map using 8000 nodes.

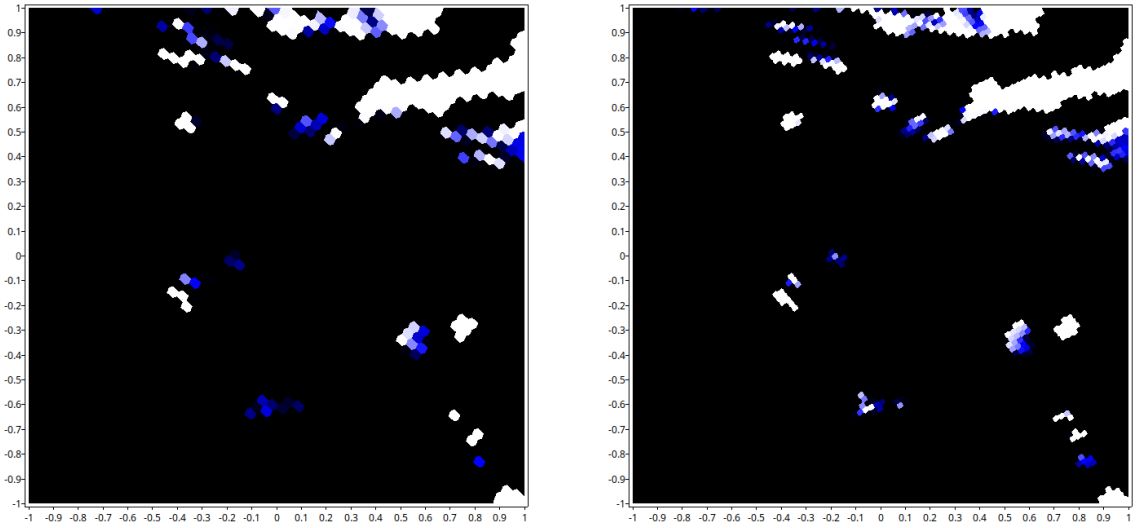

Figure 38: Activity landscape as a probability of compounds to be labeled as bioconcentrating, using the standard map or the improved map using 8000 nodes.

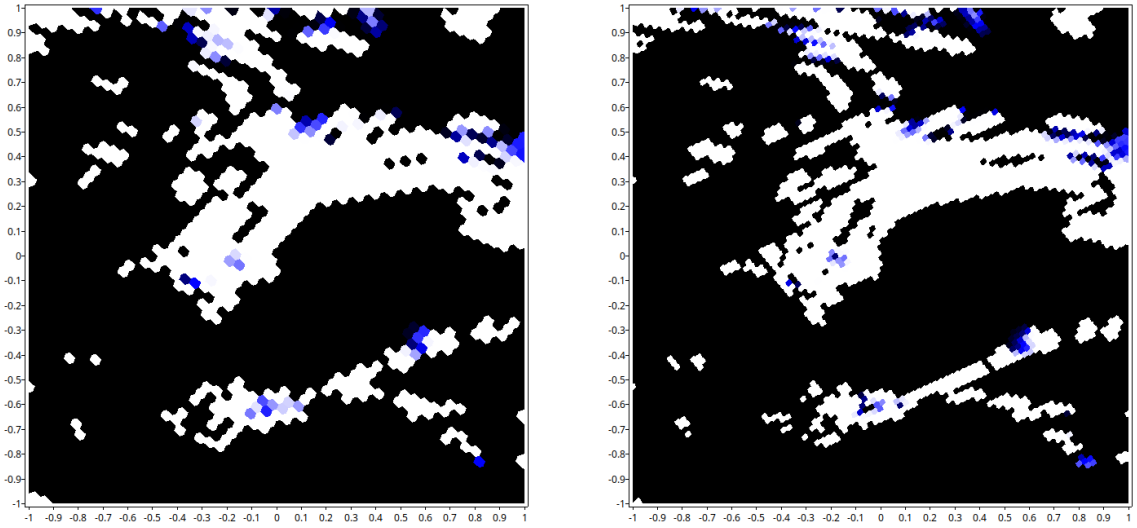

Figure 39: Activity landscape as a probability of compounds to be labeled as not bioconcentrating, using the standard map or the improved map using 8000 nodes.

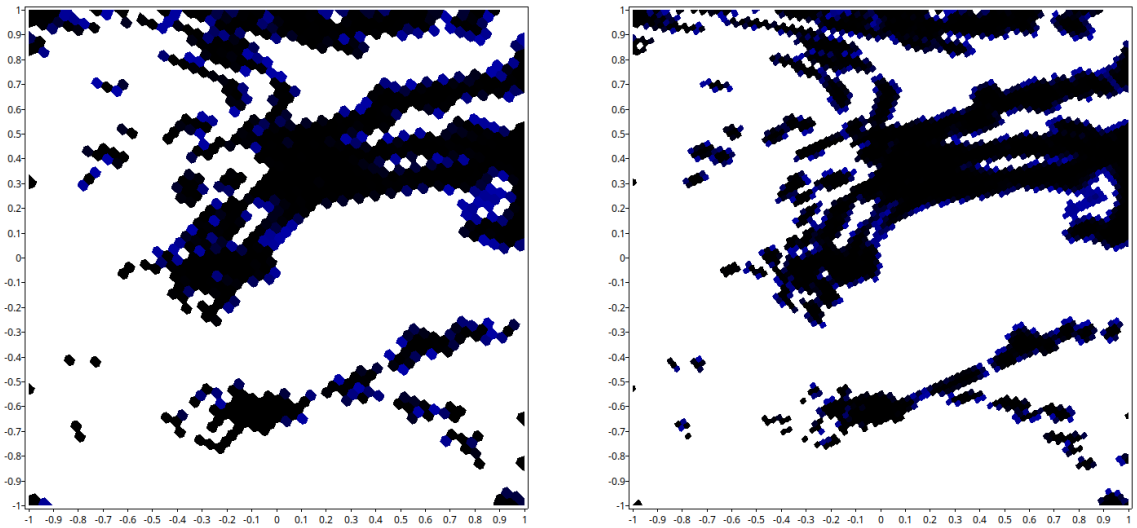

Figure 40: Activity landscape of the out of applicability domain class, standard map or improved map using 8000 nodes.

2. Conclusion

Bibliography

[1] ECHA, (Ed.: ECHA), **2017**.

[2] E. Commission, in *R. (EC) N. 1907/2006, R. (EC) N. 1907/2006* (Ed.: EC), **2006**.

[3] N.-N. I. o. T. a. Evaluation.

[4] CEFIC.

[5] E. Canada.

[6] U. EPA.

[7] OASIS.

[8] OECD.

[9] J. A. Arnot, F. A. P. C. Gobas, *Environ. Rev.* **2006**, *14*, 257–297.

[10] S. Dimitrov, N. Dimitrova, T. Parkerton, M. Comber, M. Bonnell, O. Mekenyan, *SAR QSAR Environ. Res.* **2005**, *16*, 531–554.

[11] W. Fu, A. Franco, S. Trapp, *Environ. Toxicol. Chem.* **2009**, *28*, 1372–1379.