Automatic quantification of the subcellular localization of chimeric GFP protein supported by a two-level Naive Bayes classifier

Sara Sáez-Atienzar\textsuperscript{a}, Jesús Martínez-Gómez\textsuperscript{b}, Juan Ignacio Alonso-Barba\textsuperscript{b}, José Miguel Puerta\textsuperscript{b}, María F. Galindo\textsuperscript{c}, Joaquín Jordán\textsuperscript{a}, Luis de la Ossa\textsuperscript{b}\textsuperscript{c},

\textsuperscript{a}Grupo de Neurofarmacología, Departamento Ciencias Médicas, Facultad de Medicina de Albacete, Universidad Castilla-La Mancha, IDINE, Spain
\textsuperscript{b}Grupo de investigación en Sistemas Inteligentes y Minería de Datos, Departamento de Sistemas Informáticos, I\textsuperscript{3}A Universidad Castilla-La Mancha, Spain
\textsuperscript{c}Unidad de Neuropsicofarmacología Traslacional, Complejo Hospitalario Universitario de Albacete, Área de Tecnología Farmacéutica, Facultad de Farmacia, Universidad Castilla-La Mancha, Spain

Article info

Article history:
Available online 8 October 2014

Keywords:
Confocal imaging
Protein translocation
Artificial vision
Supervised classification

Abstract

Herein we report a method supported by a two-level Naive Bayes classifier to help and improve the automatic detection and counting of cells overexpressing GFP-chimeric proteins. This toll is frequently used as a reporter for the localization and the distribution pattern of a protein in a cell. This approximation requires, besides confocal microscopy, the participation of a qualified and blind counting supervisor to avoid subjective appreciations of the imaging interpretation of the data. Indeed, this counting required specific staff training, and the interpretation of the data is inevitably subjective. In order to avoid this, we have designed an automatic detection cell counting software. We have used as a model SH-SY5Y cells overexpressing GFP-Bax protein, after 6-hydroxydopamine addition. Our proposed method learns the counting criteria after a short training stage, and uses the resulting classifier to process new images and obtaining both the number of transfected cells and the proportion of these cells that present a translocated protein. The software achieves an accuracy over 97% when detecting transfected cells, and over 93% when detecting cells with GFP-Bax translocated. Besides the hours of qualified work that can be saved, the models learnt can be stored and reused (without training) so as to homogenize criteria among different researchers.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

With relative frequency, proteins change their intracellular localization to develop specific functions. This also seems to be the case for the pro-apoptotic member of the Bcl-2 family Bax. In healthy living cells, Bax is found to reside predominantly in the cytosol, where it participates in diverse physiological events such as neuronal development and spermatogenesis (Knudson, Tung, Tourtellotte, Brown, & Korsmeyer, 1995; Rodríguez, Ody, Araki, García, & Vassalli, 1997; Shindlerand, Latham, & Roth, 1997). However, upon cell death stimuli, Bax migrates from the cytosol to organelle membranes (Gómez-Lázaro et al., 2007, 2008; Misteli & Spector, 1997; Pérez-Álvarez, Solesio, Manzanares, Jordán, & Galindo, 2009). It has been proposed that insertion of Bax into mitochondria may result in alteration of mitochondrial membrane permeability, promoting cell death in various cellular systems. To study the subcellular trafficking of a specific protein in a time-dependent manner, we and others have developed green fluorescent protein (GFP)-tagged versions of specific proteins (Gómez-Lazaro et al., 2008, 2007; Pérez-Álvarez et al., 2009). As an example we have used GFP-Bax to directly visualize the subcellular localization of Bax and to quantitatively assess the effects of different inducers on regulating its cellular distribution (Gómez-Lazaro et al., 2008, 2007; Pérez-Álvarez et al., 2009). The total number of transfected cells and those displaying a spotted pattern must be quantified. Unfortunately, these experiments require specific training of laboratory personnel and still the interpretation of the images is very subjective.

In order to automatically carry out the complete process, there are two main steps that need to be performed: cells detection and protein aggregation recognition. While there are some proposals performing similar stages (Schüffler et al., 2013; Sommer, Fiaschi, Hamprecht, & Gerlich, 2012) (specially with histopathological images), the problem we are facing presents specific requirements...
that make it hard to solve with those techniques. Firstly, the number of available labeled images is too low due to the difficult acquisition process. This reason avoids us from using proposals requiring a large set of training samples (Muthu Rama Krishnan, Chakraborty, Ranjan Rashmi, & Ray, 2012). Moreover, the acquired images present high variability due to the 24 h exposition of the cells to 6-OHDA. All these reasons forced us to propose a new technique based on an intense integration of the experts knowledge with machine learning methods. Concretely, most of the features used to detect and characterize the cells have been explicitly selected by the laboratory experts. These features differ from those used in other proposals (Arteta, Lempitsky, Noble, & Zisserman, 2012), just like PHOG (Bosch, Zisserman, & Munoz, 2007) or local binary patterns (Ojala, Pietikainen, & Maenpaa, 2002).

In this article, we present a proposal that automatically carries out the process of quantification of GFP-chimeric protein localization. The algorithm is based on supervised classification (Duda, Hart, & Stork, 2001; Mitchel, 1997), and builds a classifier model from a (small) subset of images previously processed with the supervision of the expert. Then, such a model is used to process the rest of the images in a fully automatic way.

2. Evaluation of the mitochondrial translocation of GFP-Bax

The transient transfection of SH-SYSY cells with a fluorescent GFP-Bax construct and confocal microscopy is based on a previously described procedure (Gómez-Lazaro et al., 2008). In brief, SH-SYSY cells were plated on poly (L-lysine)-coated glass slides for 24 h before the transfection at a density of 5.3 x 10^4 cells/cm^2. Transfection was performed with a plasmid containing GFP-Bax (Solesio, Sáez-Atienzar, Jordán, & Galindo, 2013) and the Lipofectamine™ reagent (Invitrogen, Barcelona, Spain). After 4 h of co-incubation, the transfection mixture was removed and replaced with fresh complete medium. Cells were then incubated for 24 h to allow for sufficient GFP-Bax expression, and finally treated for 12 h with 25 μM of 6-hydroxydopamine. Cells were then washed in fresh HEPES buffer and mounted in a chamber for confocal microscopy. The excitation/emission wavelengths for GFP 488/509 nm. Images were captured with a Zeiss LSM 710 confocal microscope, using a 40X objective.

Mitochondrial translocation of the construct resulted in a change from a diffusely green fluorescent cytoplasm to a brighter spotted one. Fig. 1 shows two sample images. White arrows indicate mitochondrial translocation, whereas the black ones point to the rest of cells.

3. Identification and classification of cells by means of supervised classification

As mentioned above, the purpose of the developed method is the quantification of the proportion of transfected cells that display a spotted pattern of protein. This involves both detecting the cells in the image, and classifying them according to the pattern. None of these processes is trivial, and therefore can not be approached by using simple processing rules. Fig. 1 shows a representative example of such circumstance. Although the main feature that distinguishes cells from the background is their color, the intensity of the objects surrounded by a rectangle in Fig. 1(A), which have been discarded by the expert, is higher than the intensity of those in Fig. 1(B), which have been accepted. Hence, not only does the criteria for acceptance depend on the particular image of the object, but also on some features of the whole image. Similar problems arise when trying to determine whether the protein of a cell is translocated. Thus, in Fig. 1(A), the surface of the spotted pattern of protein in the cell labeled with “D” is greater than the surface of the pattern in the cell labeled with “T”. However, mitochondrial translocation has been only detected in cell “T”. Due to problems as the mentioned above, we aimed to develop a method capable to mimic, to some extent, the criterion of the human expert when detecting transfected cells and translocation of mitochondria. This can be done by means of supervised classification techniques.

Supervised classification (Duda et al., 2001; Mitchel, 1997) is based on the use of a model, Cn, which relates the class of an object with some of its known features. It uses as input a set of example objects whose class is already known -namely training dataset- and uses it to estimate the model, which is then used to determine the class of new objects whose class is not previously known. (See A for further details on supervised classification).

In this context, each of the example cases corresponds to a detected object, which may or may not be a transfected cell. In order to characterize each object, we have considered a set of features (listed in Table 1) that can be extracted automatically from the image, and are related to aspects such as color, size, etc. However, its class must be specified by the expert during a training...
Therefore, from the point of view of the user, the software proceeds in two sequential stages:

1. **Training**: The software processes a subset of the images (e.g. 15). It shows each detected object to the expert, which must label it as discarded, cell with diffused aggregation of protein, or cell with translocated aggregation of protein. Fig. 2(A) shows a screenshot of the software during the training process. Once the training set of images has been manually classified, the software induces the classifier from the training dataset, which contains the features describing each object.

2. **Automatic quantization**: The software uses the classifier to process the rest of images and obtains both the number of transfected cells and the proportion of them presenting a translocation of the protein. Then, the results obtained can be shown, as in Fig. 2(B), and statistics are stored.

### Table 1
Features used to characterize each detected object. From the subimage containing each object, which can be either a shade or a transfected cell, the software obtains these numeric values.

<table>
<thead>
<tr>
<th>Feature name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>Size of the object/cell (Number of pixels)</td>
</tr>
<tr>
<td>$X_2$</td>
<td>Number of pixels over the upper background color threshold</td>
</tr>
<tr>
<td>$X_3$</td>
<td>Proportion of pixels over the upper background color threshold</td>
</tr>
<tr>
<td>$X_4$</td>
<td>Number of pixels over the aggregation color threshold</td>
</tr>
<tr>
<td>$X_5$</td>
<td>Proportion of pixels over the aggregation color threshold</td>
</tr>
<tr>
<td>$X_6$</td>
<td>Length of the major axis in the ellipse which surrounds the object/cell</td>
</tr>
<tr>
<td>$X_7$</td>
<td>Length of the minor axis in the ellipse which surrounds the object/cell</td>
</tr>
<tr>
<td>$X_8$</td>
<td>Average color of the object/cell</td>
</tr>
<tr>
<td>$X_9$</td>
<td>Number of pixels (in the whole image) over the upper background color threshold</td>
</tr>
<tr>
<td>$X_{10}$</td>
<td>Proportion of pixels (in the whole image) over the upper background color threshold</td>
</tr>
<tr>
<td>$X_{11}$</td>
<td>Number of pixels (in the whole image) over the aggregation color threshold</td>
</tr>
<tr>
<td>$X_{12}$</td>
<td>Proportion of pixels (in the whole image) over the aggregation color threshold</td>
</tr>
<tr>
<td>$X_{13}$</td>
<td>Average color of the image</td>
</tr>
<tr>
<td>$X_{14}$</td>
<td>Proportion of pixels of the object over the upper background color threshold in relation with the same proportion for the whole image ($X_{1}/X_{10}$)</td>
</tr>
<tr>
<td>$X_{15}$</td>
<td>Proportion of pixels of the object over the aggregation color threshold in relation with the same proportion for the whole image ($X_{4}/X_{12}$)</td>
</tr>
<tr>
<td>$X_{16}$</td>
<td>Average color of the object in relation with the average color of the image ($X_{8}/X_{13}$)</td>
</tr>
<tr>
<td>$X_{17}$</td>
<td>Histogram (relative frequencies) obtained from the image of the object. Each image is transformed to gray scale, and each pixel takes a value between 0 and 255. Each one of these features contains the proportion of pixels with the corresponding value (from $X_{17}$, which corresponds to 0, to $X_{272}$ which corresponds to 255)</td>
</tr>
<tr>
<td>$Y_k$</td>
<td>(k = 0: \text{Discarded Object}, k = 2: \text{Cell with diffuse aggregation}, k = 3: \text{Cell with translocated aggregation})</td>
</tr>
</tbody>
</table>

**Fig. 2.** Screenshots of the software. (A) corresponds to the training phase. The object to be processed is surrounded by a white rectangle, and the expert must provide its class. (B) Result of the process: Objects crossed out are discarded; objects delimited by gray circles are cells of interest where the protein aggregation is diffuse; and objects delimited by a dotted-white circle are cells where the protein aggregation is translocated. Statistics are shown and stored. (The image was obtained as those in Fig. 1.)
In the following lines, both steps are described in detail. Fig. 3 depicts a graphical description of the whole process.

3.1. Training

From the algorithmic point of view, the process of quantization of translocated cells can be decomposed into three well-differentiated tasks:

1. Detecting objects of interest in the image.
2. Determining which of these objects are transfected cells and discard the rest.
3. Classifying the cells depending on whether they present a translocation of the protein or not.

These tasks are considered separately. We used a method to detect objects, and then applied two classifiers: $C_{h}^{(1)}$, which determines whether an object is a transfected cell or must be discarded; and $C_{h}^{(2)}$, which considers only cells as input and classifies them according to the protein translocation. In both cases we have used the Naive Bayes classifier (Domingos & Pazzani, 1996; Minsky, 1961), extensively described in Appendix A. We choose this model because its learning requires few data and time, classification is fast, and it is comparable, in terms of accuracy, with more complex models.

In the training process, the expert labels each object as either 0 (discarded), 2 (diffuse aggregation of protein) or 3 (translocated aggregation of protein). The resulting training dataset is used to estimate the first classifier, $C_{h}^{(1)}$, which determines which objects must be discarded. Only those objects labeled with 2 or 3 (transfected cells) are used to build the training dataset used to estimate $C_{h}^{(2)}$.

In the following, we explain each one of the three steps that are also represented in Fig. 3.

3.1.1. Object detection

In contrast to other problems of cell counting, the cells obtained from images in our context can be easily distinguished from the background which, theoretically, is black (see Fig. 1). Therefore, objects can be detected by using an intensity threshold and binarizing the image. However, the minimum color threshold necessary to detect the darkest objects produces some close and brighter cells to appear as only one object. This fact must be taken into account in the design of the method.

As in subsequent stages the objects can be discarded (when classifying them with $C_{h}^{(1)}$), it is not important to detect objects (shades) that may not correspond to transfected cells (false positives). Therefore, the detection can be permissive. Namely, we developed a heuristic method based on the observation and the experts knowledge. It is described in detail in Fig. 4, and is based on the parameters also shown in the figure. Basically, it uses an intensity color threshold to distinguish objects from the background. Tiny objects are directly discarded, and so are small cells, unless they have an aggregation of protein. Lastly, as different cells may appear as a unique accumulation, each object bigger than a certain size ($\text{minCumSize}$) is segmented by using a higher color threshold to determine whether it is a single-big cell or several small cells.

3.1.2. Recognition of transfected cells

In order to generate the first training dataset, we used features $X_{1}$ to $X_{16}$ in Table 1, as preliminary experiments shown that it is better to discard the rest (which correspond to the histogram). In relation with the value of the class, provided by the expert during the training stage, the model only distinguishes two possibilities: “0” and “> 0” depending on whether the object must be discarded or considered a transfected cell respectively.

Once the training dataset has been built, it is used to estimate the Naive Bayes model $C_{h}^{(1)}$.

3.1.3. Classification of cells

The procedure for cell classification is similar to that used for cell detection. However, only those objects classified by the user as transfected cells in the previous stage (see Fig. 3) are considered to build the second training dataset. In this case, we have used the whole set of features described in Table 1, since the histogram becomes important for detecting translocations. In relation with the class, we have considered now two different labels: “2”, for cells with a diffuse aggregation of proteins; and “3”, for cells where the aggregation is translocated.

Once the second training dataset has been built, it is used to estimate the second Naive Bayes model $C_{h}^{(2)}$.

3.2. Automatic quantization

From the training stage, the software obtains both $C_{h}^{(1)}$ and $C_{h}^{(2)}$, which are then used to automatically quantify the rest of images.
Object detection

Parameters:
- minCellSize: Minimum size of a cell (number of pixels). Default Value = 200.
- smallCellMaxSize: Maximum size of a “small” cell (number of pixels). Default Value = 4000.
- minCumSize: Minimum size of an accumulation of cells (number of pixels). Default Value = 1500.
- aggregationThreshold: Threshold which denotes an aggregation (intensity of color). Default Value = 40.
- maxNumCellsCum: Maximum number of cells in an accumulation. Default Value = 2.

Procedure:
1. The image is segmented into a set of objects, being each object a connected pixel such that the intensity of their color is over lowerBackGroundThreshold. All objects with a pixel smaller than minCellSize are considered noise, and therefore discarded.
2. Those objects with size (number of pixels) smaller than smallCellMaxSize are discarded, unless a number of pixels over minCellSize corresponds to a potential aggregation of protein, i.e., the intensity of their color is over aggregationThreshold.
3. All objects whose size (number of pixels) is smaller than minCumSize are considered as a potential cell.
4. The remaining objects (those greater than minCumSize) may be an accumulation of cells, and are processed individually:
   a. The own object is segmented in different sub-objects such that their color is over upperBackgroundThreshold.
   b. All sub-objects with a size (number of pixels) smaller than minCellSize are removed.
   c. If there is no sub-objects left, the whole object is considered as a potential (dark) cell.
   d. If there is only one part left, the whole object is considered as a potential cell.
   e. If there are several sub-objects, the bigger ones are considered potential cells. The parameter maxNumCellsCum determines the maximum number of cells that can appear in an accumulation.

This process is also depicted in Fig. 3. For each one of the images, objects are detected again as described in Fig. 4. Then, given the values for the input features of each object, it is classified with the first model \( C_1 \). If it is labeled with 0, it must be discarded since it is not considered a transfected cell. Otherwise, it is considered a transfected cell, and is processed with the second model \( C_2 \), which predicts whether it presents a translocation or not.

4. Results

4.1. Accuracy of the model

We used a set of 8 different images (containing 84 objects) to test the model for transfected cell detection \( C_1 \). In order to obtain concluding results, we carried out a 5-fold cross validation, i.e., we divided the set of objects into 5-folds of equal size (20% of the dataset), and we classified each fold with a different model, which is estimated using the remaining cases (80%) as training dataset. Table 2(A) shows the confusion matrix with the results obtained for the detection of transfected cells. As can be seen, 80 out of 84 objects (95.24%) have been correctly detected. 58 out of 59 of the objects classified as cells are indeed (precision = \( \frac{58}{59} = 0.983 \)), and 58 out of 61 cells have been detected (recall = \( \frac{58}{61} = 0.9508 \)).

In order to improve the results, we also carried out a process of feature subset selection, trying to eliminate several input variables that may not contain useful information or are redundant. Concretely, we carried out a Greedy Forward Selection procedure using the own Naive Bayes classifier for evaluation in a wrapper fashion (Kohavi & John, 1997). The result of this procedure is a set of 5 input features, \( \{X_2, X_4, X_6, X_7, X_{14}\} \) that have been used to build an alternative model, \( C_{1_{fss}} \) instead of using \( X_1-X_{16} \). In this case, as can be seen in the confusion matrix shown in Table 2(B) only 2 objects were wrongly classified, and the accuracy has reached 97.62%. Precision has risen up to 1, whereas recall is also slightly higher, 0.967, as only two cells were not detected.

In order to evaluate the second model for classification of cells, \( C_2 \), we used just those objects corresponding to transfected cells, and then we proceeded in a similar way by carrying out a 5-fold cross validation. The confusion matrix shown in Table 3(A) shows that, in that case, 57 out of 61 cells have been correctly classified, i.e., the system has an accuracy of 93.44%. In this case (considering Translocated as the class of interest), the precision of the classifier is 0.9624, whereas recall is also slightly higher, 0.967, as only two cells were not detected.

Table 2

<table>
<thead>
<tr>
<th>Cell</th>
<th>Discarded</th>
<th>Classifier/Actual class</th>
</tr>
</thead>
<tbody>
<tr>
<td>58</td>
<td>3</td>
<td>Transfected Cell (61)</td>
</tr>
<tr>
<td>1</td>
<td>22</td>
<td>Discarded (23)</td>
</tr>
<tr>
<td>Total: 59</td>
<td>25</td>
<td>Objects (84)</td>
</tr>
<tr>
<td>59</td>
<td>2</td>
<td>Transfected Cell (61)</td>
</tr>
<tr>
<td>0</td>
<td>23</td>
<td>Discarded (23)</td>
</tr>
<tr>
<td>Total: 59</td>
<td>25</td>
<td>Objects (84)</td>
</tr>
</tbody>
</table>

(A) Accuracy: 95.24% – Precision: 0.983 – Recall: 0.95.
(B) Accuracy: 97.62% – Precision: 1 – Recall: 0.967.
translocated cells is 29/61 = 0.475, the classifier would report 27/61 = 0.442 (instead of 27/61 = 0.426).

We also carried out the same procedure of feature subset selection described above, obtaining \( \{X_5, X_7, X_{12}, X_{23}, X_{32}, X_{41}, X_{25}, X_{102}\} \). Table 3(B) shows the results obtained with this second model, \( C_{(2-5)} \).

This time the algorithm obtains 100% of accuracy, and both learned with a training dataset which only contains the set of features \( \{X_5, X_7, X_{12}, X_{23}, X_{32}, X_{41}, X_{25}, X_{102}\} \), obtained through feature subset selection over the whole set of features.

### Table 3
Confusion matrices obtained using the Naive Bayes classifier to determine in which cells the aggregation of the protein is translocated. The training dataset is composed by the information relative to 61 different cells. The expert detected a translocated aggregation of protein in 29 of them. (A) shows the confusion matrix obtained when using the model \( C_{(5-2)} \), learned from the training dataset containing the features \( \{X_1 - X_{12}\} \). (B) shows the confusion matrix obtained with the classifier \( C_{(2-5)} \), learned with a training dataset which only contains the set of features \( \{X_5, X_7, X_{12}, X_{23}, X_{32}, X_{41}, X_{25}, X_{102}\} \), obtained through feature subset selection over the whole set of features.

<table>
<thead>
<tr>
<th>Classifier/Actual class</th>
<th>Diffuse</th>
<th>Translocated</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>31</td>
<td>1</td>
<td>Diffuse (32)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>26</td>
<td>Translocated (29)</td>
</tr>
<tr>
<td>Total: 34</td>
<td>27</td>
<td></td>
<td>Cells (61)</td>
</tr>
<tr>
<td>(B)</td>
<td>32</td>
<td>0</td>
<td>Diffuse (32)</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>29</td>
<td>Translocated (29)</td>
</tr>
<tr>
<td>Total: 32</td>
<td>29</td>
<td></td>
<td>Cells (61)</td>
</tr>
</tbody>
</table>

(A) Accuracy: 93.44% – Precision: 0.963 – Recall: 0.897.

(B) Accuracy: 100% – Precision: 1 – Recall: 1.

4.2. Usage and time

The software\(^1\) automatically opens all images\(^2\) contained in the folder named “training”, and uses them for the training process as described in Section 3. Afterwards, it processes, without any intervention, all images contained in the folder named “new”.

In relation with the time required for the training process, it depends on the user, since labels for each image must be introduced manually, although the program shows each object almost instantly. It is important pointing out that the classifiers obtained in the training stage can be saved and reused later –skipping the training stage and directly doing the automatic quantization– even for different researchers, allowing an homogeneous criteria to evaluate images.

Once all the training images have been processed, the model is learnt in tenths of a second. Afterwards, the time required to process each image -detect and classify cells- depends on the number of objects. Nevertheless, we have measured a time 31.02 s to classify all new images\(^3\).

5. Conclusions and future work

In this paper, we have presented a novel proposal for the automatic quantification of the subcellular localization of GFP-chimeric proteins. This method is based on machine learning, and aims to follow the criteria set out by the expert research team. More specifically, we have implemented a classifier that uses two different Naive Bayes models for cell detection and cell classification. The results in this paper show that our method is able to detect 96.7% of the cells transfected with GFP-Bax protein. In 93.4% of these cases, the subcellular localization of Bax could be correctly identified (100% if using feature subset selection). The new method was validated through independent visual assessments by two experienced researchers.

The proposed method has been properly used in laboratory experiments, where researches have saved a lot of time when it has been necessary to analyze dozens of images. The quantifier is released as a tool easy to use and understand. This tool can be directly used by any user without technical skills by placing the images used for training and test in the appropriate folders. Moreover, the classifiers generated after its use can be stored and applied in future experiments, which allows independent researchers to carry out posterior quantification using the same criteria. In spite of the system has been only used to quantify cellular distribution of GFP-Bax, we believe that this tool can be applied widely to quantify subcellular localizations of proteins that are fused to fluorescent marker proteins.

The main contribution of the proposed system is the capability of properly working with a tiny set of training images (6 in the experimentation). While other proposals require an off-line annotation of the training set (due to its large volume), our approach allows experts to perform this stage before using the quantification and classification system. Such capability is very important for coping with environments that need to face with dynamic changes. This is our case, where confocal images of cell cultures are captured 24 h after 6-OHDA exposure, and different experiments can present significant variations in the type of acquired images.

As early future work, we plan to automatically select the images used for training when a whole set of images to classify has been provided. Using the current system, the subset of images used for training needs to be explicitly provided by the end user. This can lead into two types of unsuccessful experiences: a.- the user selects very few training images and obtains poor classification results, and b.- the user select a large training set and expends too much time unnecessarily. The automatic selection of the training images will be performed by discarding redundant images and therefore recognizing the most representative ones. We also have in mind the integration of matching techniques with the classification system. This will be done by using visual invariant features (e.g. Yu & Morel (2011)), and it would provide a second way to process challenging input images.

Acknowledgements

We thank Carlos Garrido and Luis Bonet for technical help. This work was partially supported by FIS-FEDER (PI080693; PI-2008/21) to M.F. Galindo and José R. Marín-Tébar, and by FEDER funds and the Spanish Government (MINECO) through projects TIN2010-20900-C04-03 and TIN2013-46638-C3-3-P.

Appendix A. The Naïve Bayes classifier

In the context of supervised classification (Duda et al., 2001; Mitchell, 1997), the information is represented by means of a set of \( n \) input features \( X = (X_1, \ldots, X_n) \), and the class \( Y \in \{c_1, \ldots, c_k\} \). Therefore, each individual object is represented by a tuple \((x, y)\), where \( x = (x_1, \ldots, x_n) \) is the vector with the values of the features for this object, and \( y \) is its class. The process takes as input a

---

2. It accepts all standard image formats.
3. We have implemented our method with language Matlab (version 2012b). Experiments have been done in a PC with an Intel i7 processor and 4 GB of RAM running an operating system Ubuntu Linux 12.04.
training dataset \( X \). \( Y \) with known outputs, and use it to estimate a model

\[
c_{\theta}(X) = \{c_1, \ldots, c_k\},
\]

which provides the class for each new example given only the values for the input features \( x \).

There is a large number of classifier models, such as Neural Networks, Decision Trees, Support Vector Machines (Han, Kamber, & Pei, 2011, chap. 6), etc. Whereas some of them are hard to train but more powerful, some other are simpler. The most representative of the latter is the Naive Bayes classifier (Domingos & Pazzani, 1996; Minsky, 1961). Despite its simplicity, Naive Bayes has been successfully used in many domains, including classification of medical images (Moralesa et al., 1998), and keeps on being one of the most used algorithms in the field of machine learning and data mining.

Given the values of the input variables for an object \( X = (x_1, \ldots, x_n) \), classifiers based on probabilistic models usually return the class \( y \in \{c_1, \ldots, c_k\} \) with the highest probability, i.e.,

\[
c_{\theta}(X) = \arg\max_{y \in \{c_1, \ldots, c_k\}} P(y|X_1, \ldots, x_n). \tag{A.1}
\]

According to Bayes’ Theorem, the conditional distribution of the output variable \( (\text{the class}) \) given the inputs, can be obtained as:

\[
P(Y|X_1, \ldots, X_n) = \frac{P(X_1, \ldots, X_n|Y)P(Y)}{P(X_1, \ldots, X_n)}. \tag{A.2}
\]

If substituting the right side of expression (A.1) by using Bayes’s rule:

\[
c_{\theta}(X) = \arg\max_{y \in \{c_1, \ldots, c_k\}} \frac{P(X_1, \ldots, x_n|y)P(y)}{P(X_1, \ldots, x_n)}. \tag{A.3}
\]

As \( P(x_1, \ldots, x_n) \) takes the same value for all \( K \) classes, this expression can be rewritten as:

\[
c_{\theta}(X) = \arg\max_{y \in \{c_1, \ldots, c_k\}} P(X_1, \ldots, x_n, y). \tag{A.4}
\]

Therefore, we use the joint probability distribution of all variables to obtain the class with the highest probability given a certain input.

Managing the joint probability distribution of all the involved variables, \( P(X_1, \ldots, x_n, Y) \), becomes unfeasible even for small datasets (with few variables). Not only for the amount of parameters needed (exponential in the number of variables), but also for the amount of data required so that the distribution is representative. Because of that there are models – so called probabilistic graphical models (Pearl, 1988) – which can approximate the joint probability distribution in a very efficient way. In order to do that, they support in marginal and conditional independencies among variables.

In the context of classification, probabilistic models assume some of these independences so that the factorization of the joint probability distribution is simpler. That is the case of Naive Bayes, which is the simplest of the probabilistic classifiers. It assumes independence among input variables given the class. Because of that, the joint probability distribution can be factorized as:

\[
P(x_1, \ldots, x_n, y) = \prod_{i=1}^{n} P(x_i|y) \cdot P(y). \tag{A.5}
\]

Therefore, the Naive Bayes classifier can be formulated as:

\[
c_{\theta}(X) = \arg\max_{y \in \{c_1, \ldots, c_k\}} \prod_{i=1}^{n} P(x_i|y) \cdot P(y). \tag{A.6}
\]

Notice that building a Naive Bayes model only requires to calculate the marginal distribution for \( y, P(Y) \), and the probability distribution of each variable conditioned to the class \( P(X_i|Y) \).

In case that \( x_i \) is a discrete variable, its conditional probability is computed by counting frequencies and using Laplace estimation. When \( X_i \) is numeric, the model assumes that its values follow a Gaussian distribution in the instances of each class. Therefore, the probability of the variable \( X_i = x_i \) given the class \( c_k \) is expressed as:

\[
p(X_i = x_i|c_k) = \mathcal{N}(x_i; \mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}}.
\]

References


