OXFORD

Briefings in Bioinformatics, 17(6), 2016, 912–925

doi: 10.1093/bib/bbv102
Advance Access Publication Date: 3 December 2015
Paper

Surveying and benchmarking techniques to analyse DNA gel fingerprint images

Jónathan Heras, César Domínguez, Eloy Mata and Vico Pascual

Corresponding author: Jónathan Heras. Department of Mathematics and Computer Science, Universidad de La Rioja, Ed. Vives, C/ Luis de Ulloa 2, 26004, Logroño, La Rioja, Spain. Tel.: +34-941299461; Fax: +34-941299460; E-mail: jonathan.heras@unirioja.es

Abstract

DNA fingerprinting is a genetic typing technique that allows the analysis of the genomic relatedness between samples, and the comparison of DNA patterns. The analysis of DNA gel fingerprint images usually consists of five consecutive steps: image pre-processing, lane segmentation, band detection, normalization and fingerprint comparison. In this article, we firstly survey the main methods that have been applied in the literature in each of these stages. Secondly, we focus on lane-segmentation and band-detection algorithms—as they are the steps that usually require user-intervention—and detect the seven core algorithms used for both tasks. Subsequently, we present a benchmark that includes a data set of images, the gold standards associated with those images and the tools to measure the performance of lane-segmentation and band-detection algorithms. Finally, we implement the core algorithms used both for lane segmentation and band detection, and evaluate their performance using our benchmark. As a conclusion of that study, we obtain that the average profile algorithm is the best starting point for lane segmentation and band detection.

Key words: DNA fingerprinting; benchmarking; lane segmentation; band detection; survey

Introduction

DNA fingerprinting is a technique for comparing DNA patterns that allow the analysis of the genomic relatedness among different samples, as well as to type and classify them. There are multiple DNA fingerprinting techniques, and the choice of which of them we must use depends on their applications (medical diagnosis, forensic science, parentage testing, food industry, agriculture and many others) [1].

After capturing DNA gel fingerprint images (also known as gel images), the process to analyse them can be split into five steps (Figure 1). First, the image is pre-processed to remove noise and fix distortions. Subsequently, the lanes of the image are segmented, and for each of those lanes (lane images) the bands are detected. Because the band positions of a lane are influenced by experimental conditions, a normalization step is carried out to compare banding patterns within the same gel and also from different gels. Finally, the similarity among lanes (based on banding patterns) is computed and graphically represented by means of a dendrogram (a hierarchical tree).

Several software tools implement the workflow to analyse gel images [2]; hence, the processing of those images is highly automated. However, there are two steps that usually require user intervention: lane segmentation and band detection. In the former, the user might need to manually add lanes, remove some of the automatically segmented lanes or adjust the thickness and curvature of the detected lanes. In the latter, the user might need to add or remove bands. These two manual corrections are tedious and time consuming, and, therefore, algorithms that reduce this effort are desirable.

In the literature, several approaches have been studied to tackle lane segmentation and band detection; however, a

Jónathan Heras has a PhD in Computer Science from the Universidad de La Rioja, Spain. He is currently working as a post-doc researcher at the Department of Mathematics and Computer Science, Universidad de La Rioja, Spain.
César Domínguez has a PhD in Mathematics from the Universidad de La Rioja, Spain. He is currently working as a senior lecturer at the Department of Mathematics and Computer Science, Universidad de La Rioja, Spain.
Eloy Mata has a PhD in Computer Science from the Universidad de La Rioja, Spain. He is currently working as a senior lecturer at the Department of Mathematics and Computer Science, Universidad de La Rioja, Spain.
Vico Pascual has a PhD in Mathematics from the Universidad de La Rioja, Spain. She is currently working as a senior lecturer at the Department of Mathematics and Computer Science, Universidad de La Rioja, Spain.
Submitted: 17 September 2015; **Received (in revised form):** 20 October 2015

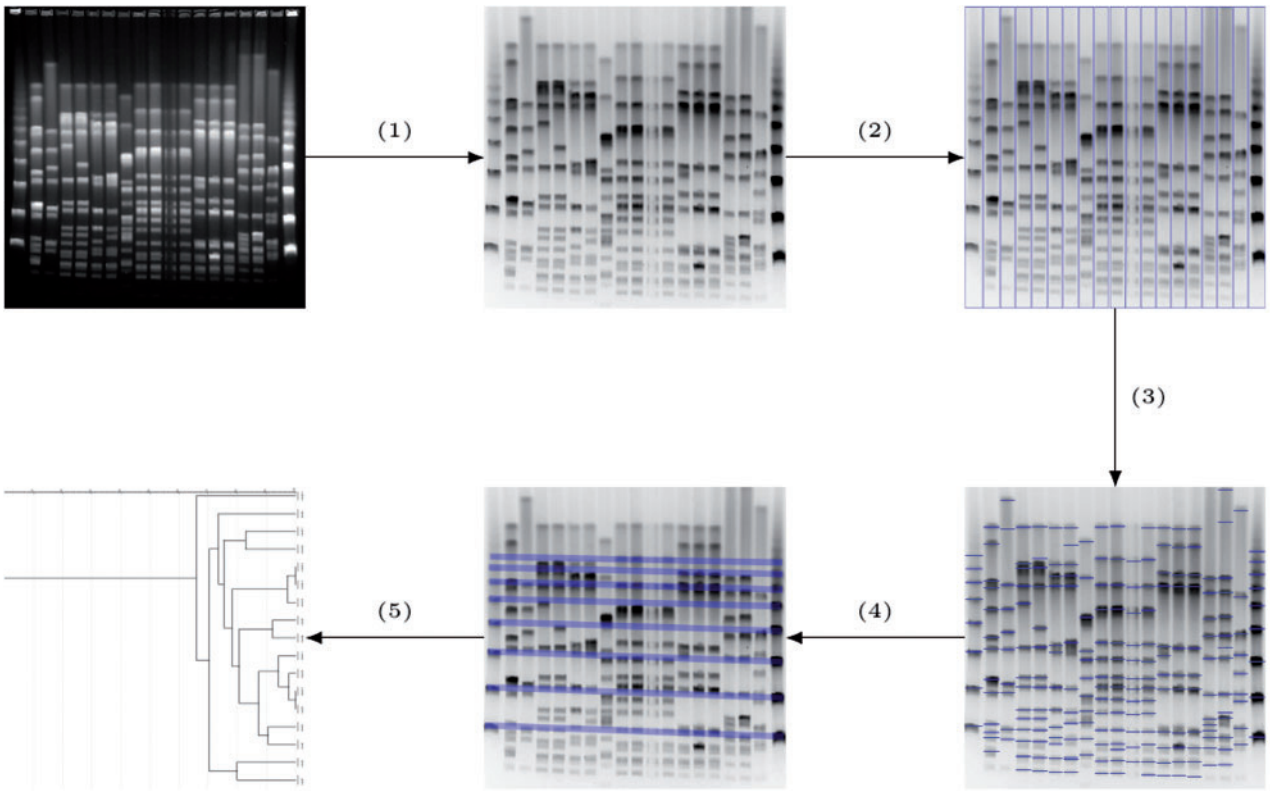

Figure 1. Workflow to analyse DNA gel fingerprint images. (1) The image is pre-processed. (2) The lanes of the image are segmented. (3) The bands of each lane are detected. (4) The gel is normalized. (5) The similarity across lanes is graphically represented by means of a dendrogram.

comparison of the different methods does not exist. The reason is 2-fold: the lack of a common benchmark to measure the performance of those methods, and the unavailability of the implementation of most algorithms. In this article, we deal with these two problems. The main contributions of this work are:

1. A survey of the fundamental techniques used in each stage of DNA gel fingerprint analysis (see 'Survey of methods' section)—special attention is paid to lane-segmentation and band-detection methods.
2. A benchmark for analysing gel and lane images, including a data set of images, the gold standards associated with those images and a set of tools that has been designed to obtain different quantitative measures from the data set (see 'Benchmarking: data set, gold standard, and analysis tools' section).
3. The implementation and comparison of the core algorithms used in the segmentation of lanes and the detection of bands (see 'Implementation and evaluation of lane-segmentation and band-detection algorithms' section).

Survey of methods

In this section, we provide a survey of the fundamental techniques used in the five stages of DNA gel fingerprint analysis. We screened PubMed Central and Google Scholar looking for corpora publications, and used the Google search engine to create a list of papers devoted to analyse gel images—the search strategy that we have followed is described in Supplementary Appendix A. This search produced 35 papers explaining different approaches.

As we have previously explained, we are mainly focused on the lane-segmentation and band-detection stages—because they are the two most time-consuming and tedious tasks. However, and for the sake of completeness, we also survey the main techniques used in the other steps of the procedure.

Image pre-processing

DNA gel fingerprint images might suffer from various types of distortions, including geometric distortion of the whole image, horizontal lane deformation (smiling), salt-and-pepper noise, non-uniform background or low/high contrast bands. Therefore, most approaches perform some safe image transformations before the actual analysis of gel images. Namely, the most usual techniques are filtering, background subtraction, smiling correction, morphological closing and deconvolution.

Filtering

The application of different filters to an image might smooth it, enhance the edges in the image and reduce its noise. Several filters have been used in the literature of gel-images analysis: the average filter [3–7] reduces the noise of the images, the notch filter is applied to smooth gel images [8, 9], the minimum filter reduces high-frequency noise [5], the Gaussian filter enhances the edges of the images [10, 11], the least-square filter reduces noise and smooths the images [12], the low pass homomorphic filter enhances the images [13], and the match filter also enhances the images [14].

Background subtraction

This technique removes local background differences. Several papers, see [12, 13], apply the rolling ball mechanism [15] to

remove the background from a gel image. In [16], the background fog is removed applying a maximum filter and, subsequently, a minimum filter. An automatic threshold is applied to equalize the grey values of the background in [14]. A polynomial function is used to model the background of images in [17] for latter subtraction, and a 'Top-Hat Transform' is used in [18] to subtract the background.

Smiling correction

This mechanism is used to correct gel distortion and smiling effects. In [12], they use a bounding box with distortion nodes to border the relevant part of the gel and to correct gel distortion and smiling effects. The method presented in [16] to correct the smiling effect consists in detecting and straightening a pair of bands common to most of the lanes. In [9], the smiling effect is fixed using a grid that captures the shape of distortions.

Morphological closing

This technique is used to remove noise. Different structuring elements can be used for morphological closing: a circular structure element of 5 pixel radius is used in [19], a square structure element is used in [20], a rectangular structure element is used in [6, 11] and a one-dimensional structural element parallel to the lanes is applied in [4].

Deconvolution

This method sharpens images and enhances the contrast of bands; however, this technique also increases the noise of the image. This mechanism has been used in [11, 12].

Lane segmentation and band detection

After a gel image has been pre-processed, the lanes of such an image are segmented, and afterwards, the bands of the segmented lanes are detected. The same intuitive idea is applied both in the segmentation of lanes and in the detection of bands. Because lane areas are covered with biological material, they appear lighter than the empty background areas between lanes; hence, strong intensity transitions between lanes and background are expected when moving horizontally across the image—analogously for bands when moving vertically across a lane. This idea is captured using a vertical (or horizontal in the case of bands) projection profile, and subsequently, obtaining the local peaks of such a profile (Figure 2).

In the literature, several projection profiles have been studied—we only consider here the definition of vertically projected profiles, and the definition of horizontally projected profiles is analogous. Given an image I with N columns and M rows of pixels, the vertical projection profile of I is an array of N elements that can be constructed using different methods—we will use $I_{i,j}$ to denote the intensity of the pixel located at column i and row j of I.

Average profile *[10, 12–14, 16, 20–27] The* i*-th element of the average profile—denoted by* $P_{AVG}(i)$—is computed using the formula:

$$P_{AVG}(i) = \frac{1}{M}\sum_{j=1}^{M} I_{i,j}$$

Derivative profile *[3, 7, 9, 17, 28–30] The* i*-th element of the derivative profile—denoted by* $P_{DER}(i)$—is computed using the formula:

$$P_{DER}(i) = \sum_{j=1}^{M} I_{i+1,j} - I_{i,j}$$

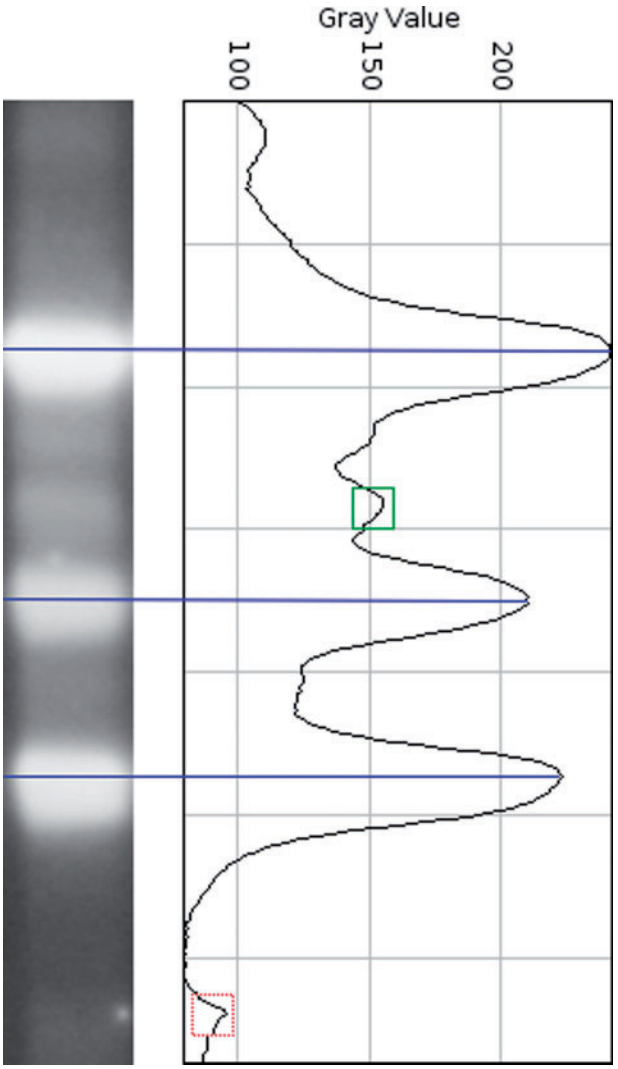

Figure 2. Average profile from a lane image. The horizontal lanes indicate the bands located from the peaks, the dotted square is a local peak coming from noise and the non-dotted square is a peak that comes from an uncertain band.

Sum profile *[4, 5, 8, 19, 28, 31, 32] The* i*-th element of the sum profile—denoted by* $P_{SUM}(i)$—is computed using the formula:

$$P_{SUM}(i) = \sum_{j=1}^{M} I_{i,j}$$

Binary *[6, 11, 18, 19, 33, 34] The* i*-th element of the binary profile—denoted by* $P_{BIN}(i)$—is computed after binarizing the image applying a threshold, and subsequently, using the formula:

$$P_{BIN}(i) = \sum_{j=1}^{M} I^{b}_{i,j}$$

where $I^{b}_{i,j}$ is the value of the pixel located at column i and row j of the binarized version of I. The computation of the threshold can be carried out using different methods like Otsu [11] or Kapur [6].

Maximum profile *[35, 36] The* i*-th element of the maximum profile—denoted by* $P_{MAX}(i)$—is computed using the formula:

$$P_{MAX}(i) = \max_{j=1}^{M} I_{i,j}$$

Standard derivation profile *[37] The* i*-th element of the standard derivation profile—denoted by* $P_{STD}(i)$—is computed using the formula:

$$P_{STD}(i) = \left(\frac{1}{N-1}\sum_{j=1}^{M}(I_{i,j} - \bar{I}_i)^2\right)^{1/2}$$

where $\bar{I}_i = \frac{1}{N}\sum_{j=1}^{M} I_{i,j}$.

Derivative of standard derivation profile *[37] The* i*-th element of the standard derivation derivative profile—denoted by* $P_{STDDER}(i)$—is computed using the formula:

$$P_{STDDER}(i) = P_{STD}(i+1) - P_{STD}(i)$$

These seven profiles are the core for the construction of more complex algorithms that refine the lane-segmentation and band-detection tasks. The enhancements introduced in the surveyed papers are summarized in Supplementary Appendix B.

A common improvement in the band-detection algorithms is the introduction of a height threshold [12, 16, 35]. As can be seen in Figure 2, some of the local peaks come from noise, and they are excluded by using a minimum height criterion: the value of the local peak must be higher than a fixed minimum to be considered as the location of a band. However, this criterion has two disadvantages: it can exclude low-intensity bands (see the non-dotted square in Figure 2), and the optimum height value varies from image to image. Therefore, the task of fixing the height threshold is usually left to the user.

Normalization

Owing to the fact that the band positions of a lane are influenced by experimental conditions, a normalization step is required to compare banding patterns within the same gel, and to compare patterns from different gels. Normalization among gels is achieved by introducing reference lanes that contain known DNA fingerprint patterns (reference markers). A reference marker consists of a set of band positions together with a physical property (mainly, the molecular weight) of each band of such a set. For example, in pulsed-field gel electrophoresis (PFGE), these reference lanes can consist in commercial molecular markers (such as Lambda Ladder PFG Marker, Middle Range PFG Marker or Low Range PFG Marker) or reference strains (e.g. *Salmonella enterica* Braenderup H9812). From the reference marker, the molecular weight of each band in the gel can be computed. This computation requires two interpolation stages: (i) a vertical interpolation within a reference lane serves to derive a migration model—this interpolation can be linear [38] or non-linear using cubic spline regression [12], and (ii) a horizontal interpolation is carried out to calculate the shift in each position of the non-reference lanes that fall between the reference lanes—in the same way that vertical interpolation, this horizontal interpolation can be carried out linearly [39] or by using cubic-spline regression [12].

Fingerprint comparison

The process to compare fingerprints (i.e. lanes) consists of two steps: the computation of similarity matrices, and the construction of dendrograms.

Similarity matrices

Given a list of n lanes, L, the similarity matrix of L is an $n \times n$ matrix where the element of row i and column j encodes the similarity between the i-th and j-th lanes of L. There are two approaches to calculate the similarity between lanes: band based and curve based [12]—a search in PubMed shows that both approaches are equally used in the literature (Supplementary Appendix A). In the former approach, the similarity between two lanes is calculated as a coefficient based on the number of matching and non-matching bands. In the latter approach, the similarity is determined using a correlation coefficient computed from the projection profiles (also known as densitometric curves) of the lanes.

Band-based methods. The comparison of lanes using band-based methods is a two-step mechanism: (i) matching is performed between the bands of two lanes, and (ii) the similarity of two lanes is computed based on the number of matching and/or non-matching bands.

In the first step, a tolerance value is introduced. This value indicates the maximum distance allowed between two bands to be considered as matching. Under this criterion, two (or more) bands on one lane might be eligible for matching with the same band on another lane (Figure 3). Two alternatives are considered to solve this problem: *closest band matching* or *first band matching*. In the former, the two bands that have the shortest distance are matched; in the latter, the first candidate that is encountered is matched (Figure 3).

Once the bands of two lanes are matched, the similarity between them can be computed using different coefficients. The most common band-based coefficients are provided in Table 1.

Curve-based methods. The curve-based coefficients work with the densitometric curves associated with the different lanes—as we have seen in 'Lane segmentation and band detection' section,

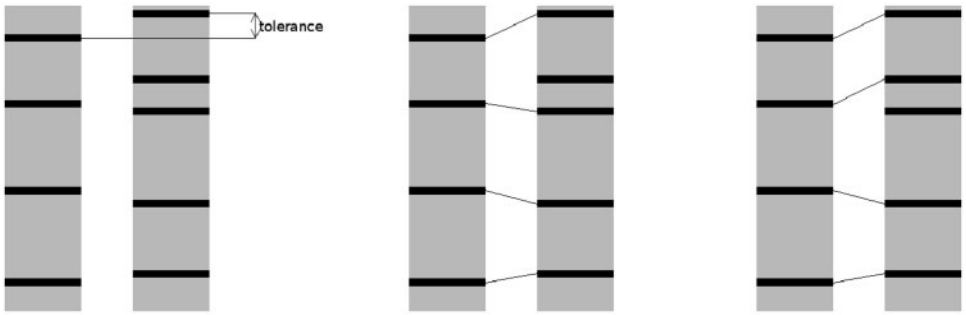

Figure 3. Matching bands in two lanes. *Left.* Example of tolerance, or maximum distance, that is defined to match two bands. *Centre.* Closest band matching. *Right.* First band matching.

different densitometric curves can be associated with a given lane. The most common curve-based coefficients used in the literature are summarized in Table 2.

Band-based versus curve-based methods. These two kinds of methods have their pros and cons. The advantage of curve-based coefficients is that they are less subjective than the band-based coefficients: band detection and tolerance fixation (two steps that require user intervention) are not required in curve-based methods, but they are necessary for band-based coefficients. However, the curve-based coefficients never show perfect matches—perfect matches are possible using the band-based coefficients. The advantage of band-based coefficients is that they provide a better control of the results (the bands selected from a lane can be manually modified by the user, but the densitometric curve cannot be altered).

Band- and curve-based methods are the traditional approach to compare fingerprints. Recently, a new approach has been proposed to classify 2D-gel images based on image texture [45]. Such a method could be extrapolated to classify fingerprints. The texture-based method can be seen as an improvement of the curve-based approach because it describes images using not only the information from the densitometric curve but also other features. Then, as in the case of the curve-based method, the texture-based approach will be less subjective than the band-based approach, but it will not show perfect matches. It remains as further work to compare the results that can be obtained with the three methods (band, curve and texture based).

Table 1. Band-based methods, their associated formula and the percentage of papers that use them

Method	Formula	Papers
Dice [40]	$\frac{2\times b_{i,j}}{b_i+b_j}$	72%
Jaccard [41]	$\frac{b_{ij}}{b_i+b_j-b_{ij}}$	12%
Different bands [12]	$1-(b_i+b_j-2b_{ij})$	8%
Ochiai [42]	$\frac{b_{ij}}{\sqrt{b_i b_j}}$	7%

Note. The following notation is used: given two lanes L_i and L_j, b_{ij} is the number of common bands (i.e. matched bands) that appear in the lanes L_i and L_j, b_i is the number of bands that appear in L_i and b_j is the number of bands that appear in L_j.

Table 2. Curve-based methods, their associated formula and the percentage of papers that use them

Method	Formula	Papers
Pearson coefficient [43]	$\frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2}\sqrt{\sum_{i=1}^{n} y_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} y_i\right)^2}}$	72%
Euclidean distance [44]	$\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$	18%
Cosine correlation [12]	$\frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$	7%

Note. The following notation is used: given two lanes L_i and L_j with height n, their densitometric curves are two arrays of n values where x_i and y_i are the i*th* values of the densitometric curves of L_i and L_j, respectively.

Clustering algorithms

The similarity matrices are fed as input to hierarchical clustering algorithms [46]. These algorithms are used to visualize the relations among fingerprints using a dendrogram—a kind of hierarchical tree. The construction of dendrograms follows an iterative process: at each step, the nearest two clusters (sets of fingerprints) are combined into a higher-level cluster. The difference among the methods relies on how the distance between the new clusters is recomputed. The main methods used in the literature are summarized in Table 3. Alternatively to hierarchical clustering, some papers apply the neighbour joining algorithm to construct phylogenetic trees [47].

Benchmarking: data set, gold standard, and analysis tools

The workflow to analyse gel images could be fully automated—just picking an algorithm for each stage—however, the results obtained automatically are unlikely to be precise. In particular, to obtain accurate results, it is usually necessary to carry out some adjustments in the lane-segmentation and band-detection steps. Therefore, it is relevant to know what are the lane-segmentation and band-detection algorithms that reduce the user interaction.

In general, the evaluation of segmentation/detection algorithms requires three ingredients: a *data set of images* (e.g. the face recognition data set [48], Berkeley segmentation data set for natural images [49], the data set of macrobiological structures [50] or the UCSB biosegmentation benchmark [51]), a *gold standard* that fixes the optimum segmentation (the gold standard is manually provided by experts) and a set of *metrics* to compare the results obtained with the segmentation/detection algorithms and the gold standards.

In this section, we present a benchmark to test algorithms for the segmentation of lanes in gel images, and the detection of bands in lane images. The benchmark can be downloaded from http://www.unirioja.es/cu/joheras/surveying/. The instrumental tool that we have used to create the benchmark is ImageJ [52]: a freely available Java platform for image processing that has been widely used in several contexts, and that can be easily extended by means of plug-ins. Using ImageJ, we have defined the gold standards of the data set. In addition, we have extended ImageJ with two plug-ins to measure the

Table 3. Linkage methods, their associated formula and the percentage of papers that use them

Method	Formula	Papers (%)
UPGMA	$d(X,Y)=\frac{1}{\lvert X\rvert\lvert Y\rvert}\sum_{x\in X}\sum_{y\in Y} d(x,y)$	27
Single (minimum) linkage	$d(X,Y)=\min(d(x,y))$ where $x\in X, y\in Y$	24
Complete (maximum) linkage	$d(X,Y)=\max(d(x,y))$ where $x\in X, y\in Y$	18
Ward	$d(X,Y)=\frac{n_X n_Y}{n_X+n_Y}\lvert m_X-m_Y\rvert^2$	8

Note. The following notation is used: X and Y are clusters, $d(X,Y)$ is the similarity between the two clusters, $d(x,y)$ is the similarity between two objects of different clusters, n_X is the number of elements of the cluster X and m_X is the centre of cluster X.

performance, using different metrics, of lane-segmentation and band-detection algorithms.

Data set

The data set consists of 50 gel images and 121 lane images saved using the tiff format. The data set of gel images consists of 24 images of good quality, 17 of intermediate quality and 9 of bad quality—the quality of the gel images was based on the straightness of lanes, and the contrast and noise of the images. In the case of lane images, the data set contains 42 good images, 66 intermediate images and 13 bad images—the quality of the lane images is based on the opinion of experts, the migration of the lane and the contrast of the bands.

The images of the data set were taken from agarose PFGE gels. These gels were prepared with agarosa D-5 (Pronadisa, Conda) in 0.5X Tris-Boric-EDTA (TBE). In each gel at least two lanes were placed with molecular weight markers (Lambda Ladder PFG Marker, Middle Range PFG Marker or Low Range PFG Marker) and the plugs of the test samples were placed in the remaining lanes. Electrophoresis was carried out in CHEF-DR II Drive Module (BioRad) machine with 2 L of 0.5X TBE with a spatula tip of thiourea. The conditions were different according to the enzyme and bacteria used. Gels were stained with an aqueous ethidium bromide solution (10 ml of ethidium bromide in 200 ml of distilled water) by immersion for 20 min under stirring. Gels were visualized with ultraviolet light and were photographed with Image Store 5000 UVP, thanks to the software ChemiGenius (GenSnap from SynGene).

Gold standards

The gold standard of the data set has been created using the *ROI Manager* tool (Figure 4) of ImageJ. The ROI Manager allows the user to fix a set of regions of interest (ROIs)—such regions might have different shapes (e.g. rectangle, oval, polygon or free shape)—and save it as a zip file for latter use. Using this tool, four biological experts in the processing of PFGE images have manually segmented, by consensus, the gel and lane images of the data set; as a result, a set of gold standards have been created. In particular, 677 lanes and 1818 bands have been manually segmented respectively for the gel and lane images.

Analysis tools

We have extended ImageJ with two plug-ins that allow the evaluation of lane-segmentation and band-detection algorithms. These plug-ins are called LaneSegPerformanceJ and BandSegPerformanceJ.

LaneSegPerformanceJ

This plug-in serves to measure the performance of lane-segmentation algorithms in terms of two different criteria: (i) the number of lanes that must be added or removed from the segmentation, and (ii) the adjustment that is necessary for the segmented lanes.

To measure the performance of segmentation algorithms regarding the former criteria, the input of LaneSegPerformanceJ is 3-fold: a gel image, the associated gold standard and the segmentation of the gel. The output generated by LaneSegPerformanceJ is the set of measures provided in the centre column of Table 4 (for more information about these measures, please refer to [53, 54]). The measures presented in such a table are based on the following values.

- True Positive (TP): the number of lanes of the gold standard that are located by the segmentation.
- False Positive (FP): the number of segmented lanes that do not correspond to any lane of the gold standard, and those segmented lanes that correspond to lanes of the gold standard that have been previously detected by other lanes of the segmentation.
- False Negative (FN): the number of lanes of the gold standard that are not located by the segmentation.
- True Negative (TN): the number of regions of the gel that do not contain either lanes of the gold standard or segmented lanes.

Unfortunately, the above values and the metrics presented in Table 4 are not intuitive for most experimental scientists, and an alternative formulation is preferred by many investigators, as shown in a series of recent publications [55–62]. In the alternative formulation, N^+ is the total number of lanes of the gold standard, N^+_+ is the number of true lanes of the gold standard predicted to be lanes by the segmentation (i.e. the TP value), N^+_- is the number of true lanes of the gold standard that are not located by the segmentation (i.e. the FN value), N^-_+ is the number of segmented lanes that do not belong to the gold standard (i.e. the FP value) and N^-_- is the number of regions of the gel that do not contain either lanes of the gold standard or segmented lanes (i.e. the TN value). Using this alternative formulation, the output generated by LaneSegPerformanceJ is given by the set of measures provided in the right column of Table 4. A crystal clear interpretation of the different metrics is provided in [55–62].

For the latter criteria, i.e. to measure the adjustment that is necessary for the segmented lanes, LaneSegPerformanceJ takes the same input and produces the same output as explained in the previous case. However, the measures presented in Table 4 are based on the area correctly segmented for each lane.

- True Positive (TP or N^+_+): the number of pixels that belong both to the gold standard and the segmentation.
- False Positive (FP or N^-_+): the number of pixels that belong to the segmentation but not to the gold standard.
- False Negative (FN or N^+_-): the number of pixels that belong to the gold standard but not to the segmentation.
- True Negative (TN or N^-_-): the number of pixels that neither belong to the gold standard nor to the segmentation.

BandSegPerformanceJ

The criteria used by this plug-in to measure the performance of band-detection algorithms is based on the number of bands that must be manually added or removed after the automatic detection.

The input of BandSegPerformanceJ is 3-fold: a lane image, the associated gold standard and the position of the detected bands in the lane (provided by a set of points). The output generated by BandSegPerformanceJ is the same set of measures generated in the LaneSegPerformanceJ plug-in, but computed using the following values.

- True Positive (TP or N^+_+): the number of bands of the gold standard that are located by the estimation.
- False Positive (FP or N^-_+): the number of points of the estimation that do not correspond to any band of the gold standard, and those estimated points that correspond to bands that have been previously detected by other point of the estimation.
- False Negative (FN or N^+_-): the number of bands of the gold standard that are not located by the estimation.

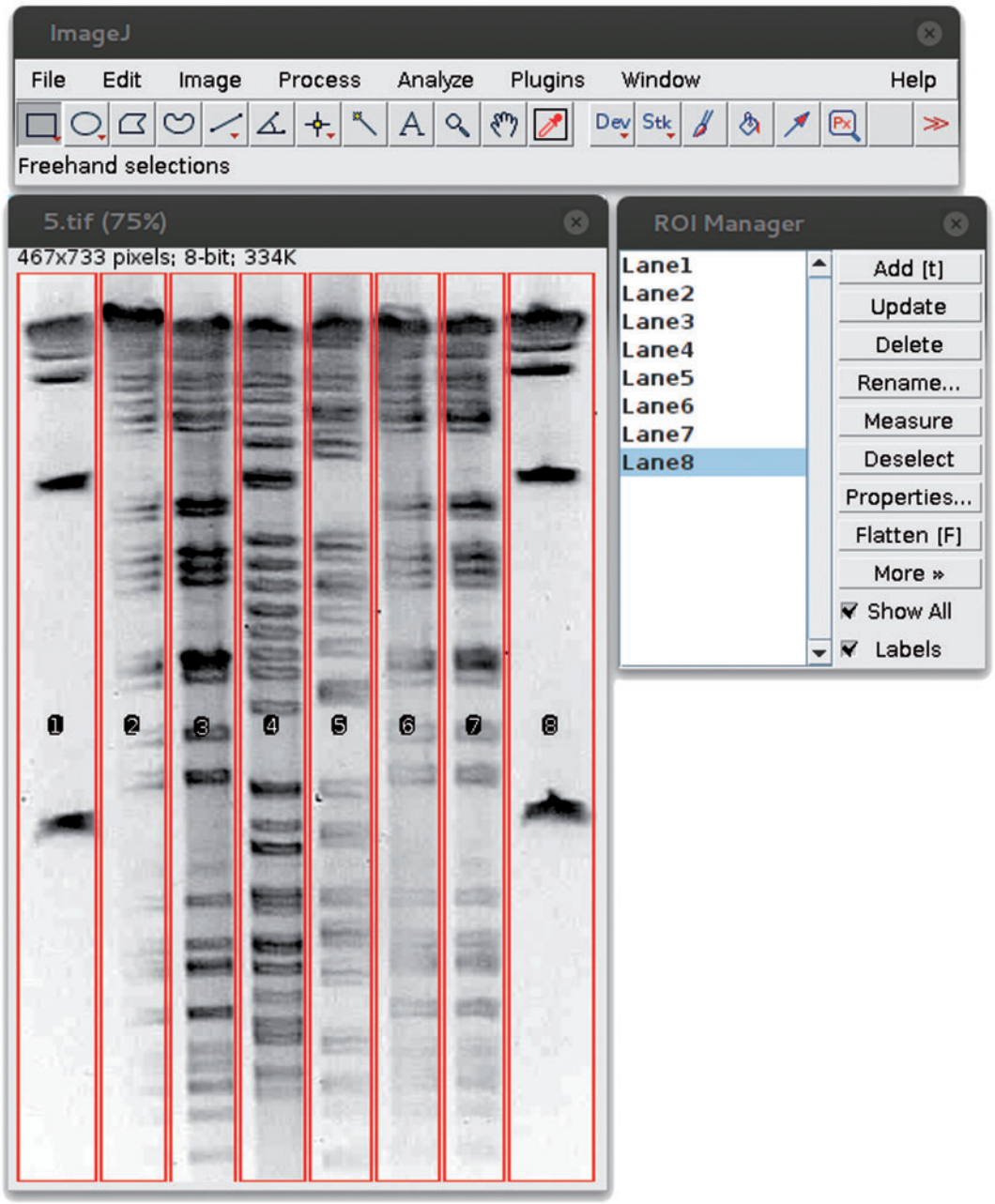

Figure 4. The ROI Manager tool and the gold standard for a gel image.

- True Negative (TN or N^-): the number of regions of the lane that do not contain either bands or estimated points.

As we have explained in 'Lane segmentation and band detection' section, different height threshold can be fixed to detect bands; hence, it might be interesting to measure the evolution of band-detection algorithms when altering such a threshold. BandSegPerformanceJ can be used to achieve this goal. To this aim, the plug-in takes as input a lane image, the associated gold standard and a batch of successive band detections. The output produced by the plug-in is the receiver operating characteristic (ROC) curve [63] (and the area under the ROC curve, also known as AUROC) associated with the batch, and the above mention measures for each individual detection of the batch.

Implementation and evaluation of lane-segmentation and band-detection algorithms

In this section, we present the implementation of the core algorithms for lane segmentation and band detection presented in 'Lane segmentation and band detection' section. Additionally, we evaluate them using the benchmark introduced in 'Benchmarking: data set, gold standard, and analysis tools' section.

Implementation

We have implemented two ImageJ plug-ins called *LaneManagerJ* and *BandManagerJ*—they can be downloaded from http://www.unirioja.es/cu/joheras/surveying/.

Table 4. Measures generated by the LaneSegPerformanceJ and BandSegPerformanceJ plug-ins

Measure	Value using traditional formulation	Value using alternative formulation
Positive	$TP+FN$	N^+
Negative	$FP+TN$	N^-
Accuracy	$\frac{TP+TN}{TP+FN+FP+TN}$	$1-\frac{N^+_+ + N^-_-}{N^+ + N^-}$
Precision	$\frac{TP}{TP+FP}$	$\frac{N^+_+}{N^+_+ + N^-_+}$
Sensitivity/ recall	$\frac{TP}{TP+FN}$	$1-\frac{N^+_+}{N^+}$
Fallout	$\frac{FP}{FP+TN}$	$\frac{N^-_+}{N^-}$
Specificity	$\frac{TN}{FP+TN}$	$1-\frac{N^-_+}{N^-}$
Negative predictive value	$\frac{TN}{FN+TN}$	$\frac{N^-_-}{N^-_- + N^-_+}$
False discovery rate	$1-precision$	$1-precision$
False negative rate	$\frac{TN}{FP+TN}$	$1-\frac{N^-_-}{N^-}$
Likelihood ratio $LR+$	$\frac{sensitivity}{1-specificity}$	$\frac{sensitivity}{1-specificity}$
Likelihood ratio $LR-$	$\frac{1-sensitivity}{specificity}$	$\frac{1-sensitivity}{specificity}$
F-score/ F-measure ($\alpha = 0.5, 1, 2$)	$\frac{(1+\alpha)\times precision\times recall}{\alpha\times precision+recall}$	$\frac{(1+\alpha)\times precision\times recall}{\alpha\times precision+recall}$
Mathews correlation coefficient	$\frac{(TP\times TN)-(FP\times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$	$\frac{1-\left(\frac{N^+_+ + N^-_-}{N^+ + N^-}\right)}{\sqrt{\left(1+\left(\frac{N^-_- + N^+_-}{N^+}\right)\right)\left(1+\left(\frac{N^+_+ + N^-_+}{N^-}\right)\right)}}$

The LaneManagerJ plug-in takes as input a gel image, and allows the user to segment the lanes of such an image choosing one of the seven algorithms based on the seven profiles presented in 'Lane segmentation and band detection' section (`average`, `binary`, `derivative`, `maximum`, `STD`, `STD-derivative` and `sum`). The output produced by this plug-in is a set of regions (rois) that can be stored as a zip file—this zip file can be fed as input to the analysis tools presented in 'Benchmarking: data set, gold standard, and analysis tools' section. Likewise, the BandManagerJ plug-in takes as input a lane image and generates as output the position of the bands of such an image using the profile selected by the user. These two plug-ins could be extended to incorporate the more refined algorithms that rely on the different profiles (Supplementary Appendix B).

Both plug-ins also feature batch generation; that is, they can automatically generate the roi files produced by the seven algorithms and store them into a folder. Additionally, the BandManagerJ plug-in supports the batch generation with different height thresholds—the number of thresholds can be fixed by the user.

Evaluation

The rest of this section will be devoted to show the behaviour of the seven algorithms implemented in LaneManagerJ and BandManagerJ when applied to the benchmark presented in 'Benchmarking: data set, gold standard, and analysis tools' section. In particular, we include an evaluation of the lane-segmentation and the band-detection algorithms—the workflow to evaluate the performance of the algorithms in our benchmark is depicted in Figure 5. To this aim, we provide the mean (standard derivation) of the FPs and FNs, as explained in 'Benchmarking: data set, gold standard, and analysis tools' section, together with the accuracy, precision (positive predictive value), sensitivity, specificity, negative predictive value and F-score with $\alpha = 1$ (additionally, AUROC is also included in the case of band-detection algorithms) of the different algorithms in the evaluated images—these values have been obtained using the analysis tools presented in 'Benchmarking: data set, gold standard, and analysis tools' section.

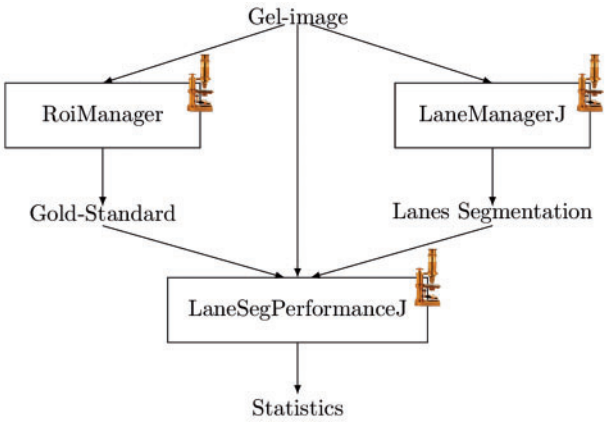

Figure 5. Workflow to evaluate the performance of a lane-segmentation algorithm using our benchmark and LaneManagerJ. A gel image is given as input to the ROIManager tool of ImageJ where an expert defines the gold standard. In addition, the same image is the input of LaneManagerJ where the user can segment the lanes of the gel image using the available methods. Subsequently, the gel image, the gold standard and the lane segmentation are provided as input to LaneSegPerformanceJ that evaluates how good is the segmentation. The workflow for band detection is analogous using BandManagerJ instead of LaneManagerJ, and BandSegPerformanceJ instead of LaneSegPerformanceJ.

We include a statistical evaluation in the study. In the literature, a parametric test is preferred for model comparison when the necessary requirements are satisfied, but a non-parametric test is also acceptable when the distribution does not fulfil the assumptions [64–66]. Therefore, we firstly use analysis of variance with repeated measures to test whether there are differences on the evaluated methods. Secondly, we compare each pair of methods with a paired t-test using Bonferroni correction. When parametric conditions are not verified, we take into account the corresponding non-parametric tests (i.e. Friedman or Wilcoxon tests). These last comparisons allow us to sort (from best to worst) the methods according to a studied characteristic. A symbol $>$ between two methods will mean that a significant difference between two consecutive methods has been found.

Evaluation of lane-segmentation algorithms

As we have explained in 'Benchmarking: data set, gold standard, and analysis tools' section, the performance of lane-segmentation algorithms is evaluated regarding two criteria: the correct detection and the correct segmentation of lanes—these two criteria measure respectively how many lanes must be added or removed, and the adjustment of the thickness and shape of the lanes. We start by analysing the seven algorithms implemented in LaneManagerJ with respect to the lane-detection criterion.

Table 5 includes the means and standard deviations of the different algorithms to detect lane positions. There exist significant differences among the different methods in all the studied aspects. The `average` algorithm is the method with significant less FPs—i.e. the method that requires less user-intervention

Table 5. Evaluation of surveyed profiles for lane detection

Metric	AVG	BIN	DER	MAX	STD	STDDER	SUM
Accuracy	0.95 (0.08)	0.78 (0.11)	0.85 (0.11)	0.8 (0.11)	0.93 (0.1)	0.88 (0.12)	0.82 (0.1)
Precision	0.95 (0.07)	0.68 (0.2)	0.78 (0.18)	0.7 (0.2)	0.89 (0.11)	0.82 (0.13)	0.72 (0.19)
Sensitivity	0.96 (0.09)	0.98 (0.06)	0.98 (0.04)	0.98 (0.06)	0.98 (0.06)	0.95 (0.07)	0.99 (0.05)
Specificity	0.95 (0.06)	0.68 (0.2)	0.78 (0.18)	0.7 (0.2)	0.9 (0.11)	0.83 (0.13)	0.72 (0.18)
NPV	0.97 (0.07)	0.98 (0.05)	0.98 (0.03)	0.99 (0.04)	0.98 (0.05)	0.96 (0.06)	0.99 (0.04)
F-score	0.95 (0.06)	0.78 (0.16)	0.85 (0.12)	0.80 (0.14)	0.93 (0.08)	0.87 (0.08)	0.81 (0.14)
FP	0.72 (1.16)	7.56 (7.55)	4.7 (5.57)	7.5 (6.94)	1.96 (2.89)	3.24 (3.52)	5.94 (5.96)
FN	0.46 (0.99)	0.32 (0.87)	0.22 (0.46)	0.26 (0.88)	0.28 (0.7)	0.6 (0.95)	0.14 (0.45)

	F or χ^2	Differences after Bonferroni correction
Accuracy	28.879***	AVG,STD>STDDER,DER,MAX,SUM,BIN; STDDER>MAX
Precision	38.328***	AVG>STD>STDDER,DER,MAX,SUM,BIN; STDDER>MAX
Sensitivity	2.590*	SUM,MAX,DER,BIN,STD,AVG,STDDER
Specificity	39.322***	AVG>STD,BIN,DER>MAX>SUM>STDDER
NPV	2.955*	SUM,MAX,DER,BIN,STD,AVG,STDDER
F-score	26.260***	AVG,STD>STDDER,DER>SUM,MAX,BIN
FP	26.628***	AVG>STD>STDDER,DER,SUM,MAX>BIN; STDDER>SUM;DER>MAX
FN	2.845*	SUM,DER,MAX,STD,BIN,AVG,STDDER; SUM,DER>STDDER

Note. *$P < 0.05$; *** $P < 0.001$.

Top. Means (standard deviations) of different methods to obtain lane positions ($N = 50$). We use the following abbreviations: AVG = average, BIN = binary, DER = derivative, MAX = maximum and STDDER = STD-derivative. *Bottom.* Differences among methods and paired differences between two different methods after Bonferroni correction.

removing lanes. All the methods work relatively well with respect to FNs—that is, usually, the user does not need to add new lanes with these methods—the `sum` algorithm being the best one.

Figure 6 includes the representation of the accuracy, precision, sensitivity, specificity, negative predictive value and F-score of the four best lane-segmentation algorithms according to the data included in Table 5: `average`, `derivative`, `STD` and `STD-derivative`. `Average` is the best method in four of the six characteristics (accuracy, precision, specificity and F-score), and it is similar to `STD` in sensitivity and negative predictive value (the best in these characteristics). `Average` has a mean above 0.95 in all these six characteristics, and the second best method is `STD` with a mean above 0.89 in these characteristics.

When we study the lane-detection algorithms in the images grouped by their quality, we obtain a similar classification of the methods in the three categories (bad, intermediate and good quality). `Average` is the best method in the three categories. It has a mean of <1 FP in all the categories. With good images it obtains a mean of 0.17 FNs, which equals the best methods (`sum` and `derivative`) in that category, and obtains 1.1 FNs in the bad-quality images. See Supplementary Appendix C for details.

We focus now on the evaluation of lane-segmentation algorithms regarding the segmentation criterion. Table 6 includes the means and standard deviations of different methods to obtain the segmentation of the lanes. In this case, FPs and negatives are the number (divided by 1000) of pixels that should be removed or added to the segmentation, respectively. There exist significant differences among the methods in all the studied aspects. `Derivative` and `STD-derivative` are the methods with significant less FP but with more FNs. On the contrary, `average` is the method with significant less FNs but with more FPs. This property is shared with the rest of the methods: more FN (positive) follows less FP (negative).

Figure 7 includes the representation of the accuracy, precision, sensitivity, specificity, negative predictive value and F-score of the four best lane-segmentation algorithms according to the

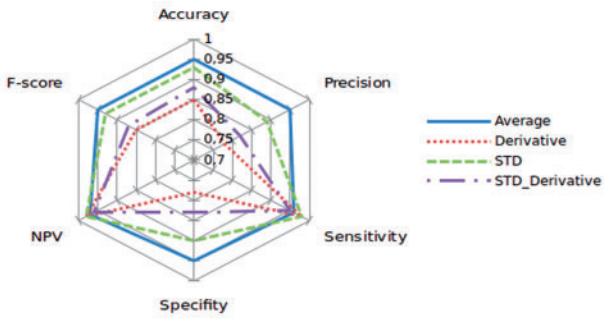

Figure 6. Representation of the accuracy, precision, sensitivity, specificity, negative predictive value and F-score of the four best lane-segmentation algorithms according to the data included in Table 5.

data included in Table 6: `average`, `maximum`, `STD` and `sum`. It is worth noting that the big amount of TN in the segmentation makes accuracy, specificity and predictive negative value almost 1 in all the methods. `Average` has the best sensitivity and F-score, and `derivative` and `STD-derivative` the best precision. `Average` and `STD` are the best methods with almost a mean above 0.85 in all these six characteristics.

Again, when we study the lane-segmentation algorithms in the images grouped by their quality (Supplementary Appendix C), we obtain a similar classification of the methods in the three categories (bad, intermediate and good quality).

In the literature, there are several papers that use the lane-detection measures included here to evaluate lane-segmentation algorithms; however, none of them evaluates how good is the actual segmentation of the lanes, or compares different approaches. The only paper that evaluates algorithms from different sources is [20], it takes two data sets of gel images and measures the performance of the algorithms presented in [20, 24, 67, 68]—all these algorithms use the average profile as core algorithm—considering the number of TP, FP and FN, and the recall, precision and F-score ($\alpha = 1$) values; however, no

Table 6. Evaluation of surveyed profiles for lane segmentation

Metric	AVG	BIN	DER	MAX	STD	STDDER	SUM
Accuracy	0.98 (0.01)	0.96 (0.02)	0.96 (0.02)	0.97 (0.02)	0.98 (0.02)	0.97 (0.02)	0.97 (0.02)
Precision	0.84 (0.09)	0.85 (0.09)	0.93 (0.08)	0.84 (0.09)	0.83 (0.1)	0.92 (0.08)	0.85 (0.07)
Sensitivity	0.92 (0.06)	0.66 (0.13)	0.46 (0.1)	0.73 (0.14)	0.85 (0.1)	0.62 (0.15)	0.78 (0.13)
Specificity	0.98 (0.01)	0.99 (0.01)	1 (0.01)	0.99 (0.01)	0.99 (0.01)	1 (0.01)	0.99 (0.01)
NPV	0.99 (0.01)	0.97 (0.03)	0.96 (0.02)	0.98 (0.02)	0.99 (0.01)	0.97 (0.02)	0.98 (0.02)
F-score	0.87 (0.06)	0.73 (0.1)	0.61 (0.1)	0.77 (0.1)	0.84 (0.08)	0.73 (0.13)	0.80 (0.09)
FP	66.71 (49.6)	46.76 (37.48)	11.38 (13.55)	55.98 (57.4)	63.72 (47.98)	17.58 (17.99)	50.75 (37.07)
FN	29.54 (29.66)	112.66 (72.48)	179.94 (96.3)	90.09 (61.4)	51.85 (40.4)	125.97 (72.7)	71.27 (54.66)

	F or χ^2	Differences after Bonferroni correction
Accuracy	22.134***	AVG,STD,SUM,MAX,STDDER,BIN>DER; AVG>SUM; STD>MAX; SUM>BIN
Precision	28.396***	DER,STDDER>SUM,BIN,AVG,MAX,STD
Sensitivity	129.805***	AVG>STD>SUM,MAX >BIN,STDDER>DER
Specificity	21.891***	DER,STDDER>BIN,SUM,STD,MAX,AVG; BIN>AVG
NPV	48.049***	AVG,STD>SUM,MAX,BIN,STDDER>STD; SUM>BIN; MAX>STDDER
F-score	74.930***	AVG>STD,SUM,MAX,BIN,STDDER>STD; STD>MAX; SUM>BIN
FP	28.679***	DER,STDDER>BIN,SUM,MAX,STD,AVG; SUM>STD
FN	72.682***	AVG>STD,SUM,MAX,BIN,STDDER>DER; STD>MAX>STDDER

Note. ***$P < 0.001$.

Top. Means (standard deviations) of different methods to obtain lane rois ($N = 50$). *Bottom.* Differences among methods and paired differences between two different methods after Bonferroni correction.

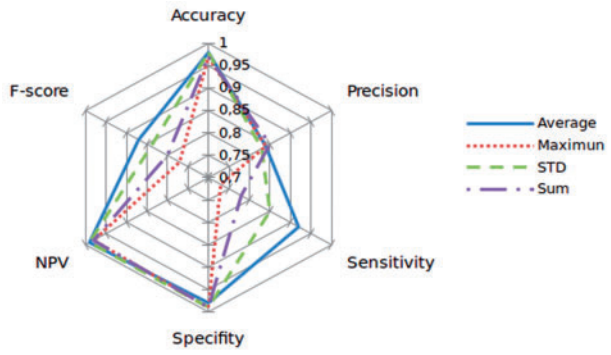

Figure 7. Representation of the accuracy, precision, sensitivity, specificity, negative predictive value and F-score of the four best lane-segmentation algorithms according to the data included in Table 6.

comparison is provided. In [19], the algorithms of four software tools were evaluated using the number of TP, FP and FN; similarly, the algorithms implemented in *laneruler* and *image* were compared in [4] using the sensitivity and specificity values. The number of TP, FP and FN has been used to analyse different configurations of the algorithms introduced in [5, 14, 36]. The algorithm presented in [32] was studied using the precision, recall and F-score ($\alpha = 1$) values. Finally, the three methods presented in [13] were analysed using the number of TP, FP, FN and TN, and the sensitivity, specificity and accuracy values. A summary of the measures used in the literature is provided in Table 7. It is worth mentioning that none of these papers provides the data set of images, the gold standard or the implemented algorithms to reproduce their results.

Evaluation of band-detection algorithms

Table 8 includes the means and standard deviations of different methods provided by BandManagerJ to obtain the positions of the bands in a lane image. There exist significant differences among the different methods in all the studied aspects. `Average` is the method with significant less FPs and the second method with less FNs (only slightly overrated by `sum`).

Figure 8 includes the representation of the AUROC, accuracy, precision, sensitivity, specificity, negative predictive value and F-score of the four best band-detection algorithms according to the data included in Table 5: `average`, `maximum`, `STD` and `sum`. AUROC can be considered as the fundamental comparative measure because it is independent of any threshold to differentiate between positive and negative samples, and it provides a good representation of the quality of the classifier in terms of both false-positive and false-negative detection [45]. Considering this parameter, `Average` and `sum` share the first position. `Average` is the best method in four of the other six characteristics (accuracy, precision, sensitivity and F-score), and `sum` has the best specificity and predictive negative value. `Average` is the best method, and together with `maximum` has a mean above 0.80 in all these six characteristics. `STD-derivative` (which is not included in the figure) has clearly the worst performance with most of the parameters below 0.5.

When we study the band-detection algorithms in the images grouped by their quality (see Supplementary Appendix C for details), we obtain a similar classification of the methods in the three categories (bad, intermediate and good quality). `Average` is the best method in the three categories. The FP means of the methods are stable with respect to the quality of the image. For instance, `average` only increases to a mean of 1.23 FPs with bad quality images. The false-negative means varies in one or two depending on the quality. For instance, `average` ranges from 2.31 with good images to 4.23 with bad images, or `sum` from 1.64 with good images to 3.23 with bad images.

In the literature, we can find a few papers that evaluate the performance of band-detection algorithms; nevertheless, none of these papers compares its approach with other methods. The performance of four software tools in [19] was measured using the number of TP, FP and FN. The algorithm presented in [10] was evaluated, with several configurations, in two sets of lane images with different qualities using as measures the TP rate (sensitivity), the FP rate (fall-out) and the accuracy. The authors

Table 7. Measures used in the literature to evaluate the performance of lane-segmentation and band-detection algorithms

Authors	Data set size	Analyse	Evaluate	Measures
Akbari *et al.* [13]	20 gel images (456 lanes)	LD	Algorithms presented in the article	TP, FP, FN, TN, SENS, SPC and ACC
Caricade *et al.* [19]	96 lane images, 12 gel images	BD, LD	Software tools	TP, FP and FN
Chan *et al.* [10]	2 sets of lane images	BD	Variants of the same algorithm	SENS, FPR and ACC
Ismail *et al.* [14]	20 gel images	LD	Variants of the same algorithm	FP and FN
Labyed *et al.* [37]	15 lane images (63 bands)	BD	Variants of the same algorithm	TP, FP and FN
Lee *et al.* [27]	10 lane images (430 bands)	BD	Variants of the same algorithm	TP, FP and FN
Machado *et al.* [5]	22 gel images (365 lanes)	LD	Variants of the same algorithm	TP, FP and FN
Moreira *et al.* [20]	235 gel images (2073 lanes)	LD	Algorithms from several sources	TP, FP, FN, TN, SENS, PPV and F1
Park *et al.* [32]	38 gel images	LD	Algorithm presented in the article	PPV, SENS and F1
Sotaquirá [36]	25 lane images, 25 gel images	BD, LD	Variants of the same algorithm	TP, FP and FN
Wong *et al.* [4]	161 gel images	LD	Software tools	SENS and SPC

Note. The following abbreviations are used:
BD = band detection, LD = lane detection, TP = true positive, FP = false positive, FN = false negative, TN = true negative, FPR = false-positive rate, ACC = accuracy, SENS = sensitivity, SPC = specificity, PPV = precision, F1 = F-measure ($\alpha = 1$).

Table 8. Evaluation of surveyed profiles

Metric	AVG	BIN	DER	MAX	STD	STDDER	SUM
AUROC	0.89 (0.07)	0.77 (0.13)	0.8 (0.12)	0.87 (0.11)	0.86 (0.1)	0.44 (0.14)	0.89 (0.09)
Accuracy	0.88 (0.08)	0.75 (0.13)	0.71 (0.12)	0.81 (0.15)	0.84 (0.12)	0.46 (0.13)	0.8 (0.16)
Precision	0.93 (0.11)	0.85 (0.19)	0.79 (0.15)	0.81 (0.19)	0.87 (0.15)	0.42 (0.16)	0.78 (0.2)
Sensitivity	0.83 (0.12)	0.63 (0.19)	0.55 (0.21)	0.83 (0.14)	0.79 (0.14)	0.4 (0.16)	0.87 (0.11)
Specificity	0.93 (0.11)	0.88 (0.18)	0.87 (0.14)	0.81 (0.2)	0.89 (0.15)	0.52 (0.16)	0.77 (0.23)
NPV	0.86 (0.08)	0.74 (0.11)	0.69 (0.11)	0.85 (0.11)	0.83 (0.1)	0.5 (0.12)	0.88 (0.1)
F-score	0.87 (0.09)	0.71 (0.15)	0.63 (0.17)	0.80 (0.14)	0.82 (0.12)	0.40 (0.15)	0.81 (0.14)
FP	0.89 (1.25)	2.19 (4)	1.78 (1.25)	2.97 (3.03)	1.55 (1.68)	8.02 (2.85)	3.95 (4.36)
FN	2.72 (2.07)	5.57 (3.17)	7.21 (4.64)	2.78 (2.48)	3.28 (2.55)	9 (3.57)	2.06 (1.9)

	F or χ^2	Differences after Bonferroni correction
ROC area	324.723***	AVG,SUM,MAX,STD>DER,BIN>STDDER;AVG>MAX; SUM>STD
Accuracy	254.942***	AVG>STD>MAX,SUM>BIN>DER>STDDER
Precision	238.282***	AVG>STD,BIN,MAX,DER,SUM>STDDER; STD>MAX; BIN>DER
Sensitivity	207.803***	SUM>AVG,MAX>STD>BIN>DER>STDDER
Specificity	173.945***	AVG>STD,BIN,DER>MAX>SUM>STDDER
VPN	296.337***	SUM>AVG,MAX,STD>BIN,>DER>STDDER; AVG>STD
F-score	266.549***	AVG>STD,SUM,MAX>BIN>DER>STDDER
FP	230.423***	AVG>STD,DER,BIN,MAX>SUM>STDDER; DER>MAX
FN	184.032***	SUM>AVG,MAX>STD>BIN,DER>STDDER

***$P < 0.001$.
Top. Means (standard deviations) of different methods to obtain bands ($N = 120$). AUROC is computed using 25 successive thresholds for the height criterion. *Bottom.* Differences among methods and paired differences between two different methods after Bonferroni correction.

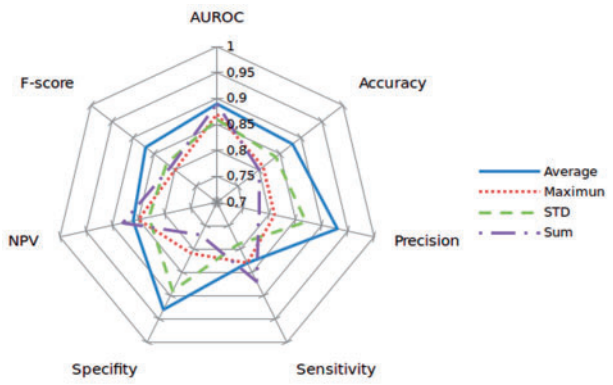

Figure 8. Representation of the AUROC, accuracy, precision, sensitivity, specificity, negative predictive value and F-score of the four best band-detection algorithms according to the data included in Table 8.

of [27, 36, 37] evaluated their algorithms with different configurations using the number of TP, FP and FN. A summary of the measures used in the literature is provided in Table 7.

Discussion and conclusions

The analysis of DNA gel fingerprint images is a widely studied problem, and several approaches have been proposed to simplify the two stages that require user intervention: lane segmentation and band detection. In this article, we have surveyed the main techniques available in the literature—not only for lane segmentation and band detection but also for the other stages of the procedure. The conclusion for the surveyed lane-segmentation and band-detection algorithms is that, even if several approaches exist, they are all based on enhancing the location of peaks of a profile (it can be the average, binary,

derivative, maximum, STD, STD-derivative or sum profile) obtained from the horizontal (or vertical) projection of an image.

The examined papers rarely provide the implementation of their approaches, a comparison with other methods, or the data set that was used to evaluate their algorithms. Hence, our survey was not enough to establish what were the best algorithms for lane segmentation and band detection. We have overcome this drawback, thanks to the development of a publicly available benchmark that includes two data sets of gel and lane images, the gold standards associated with those images and a set of tools—implemented as ImageJ plug-ins—to analyse the performance of lane-segmentation and band-detection algorithms. The infrastructure provided by our benchmark avoids the burden of choosing data sets for testing algorithms, and reimplementing analysis tools and evaluation metrics for comparisons.

The evaluation measures implemented in the two plug-ins of the benchmark are related to three criteria: the correct segmentation of lanes, the correct detection of lanes and the correct detection of bands. Measures related to the latter two criteria have been previously used in the literature, see Table 7; however, the evaluation of lane-segmentation algorithms based on the correctness of the segmentation is new. None of the papers including an evaluation provides the data set, the tools to perform the measurements or the comparison among different algorithms—three aspects covered in the current article.

Finally, we have implemented the seven core algorithms, based on finding the location of peaks of seven profiles (`average`, `binary`, `derivative`, `maximum`, `STD`, `STD-derivative` and `sum`), to segment lanes and detect bands. These algorithms have been implemented inside ImageJ plug-ins and have been evaluated using our benchmark. In general, the `average` algorithm excels the rest of the algorithms regarding the three evaluated criteria: lane segmentation and lane and band detection.

The ImageJ plug-ins for lane segmentation and band detection are freely available and open source; therefore, they can be improved with the enhancements available in the literature. It is also worth mentioning that these plug-ins are a good starting point for fast prototyping because they can be combined with the vast number of features to process images available in ImageJ (e.g. filtering, background subtraction, morphological operations and so on); namely, they have been used as a basis for the GelJ tool [69]. Moreover, the plug-ins devoted to evaluate the performance of lane-segmentation and band-detection algorithms can be used in other contexts where the analysis of image segmentation or object detection is required. As further work, and as we have explained in 'Similarity matrices' section, it would be interesting to include in GelJ the texture-based method presented in [45] and compare it with the traditional band- and curve-based approaches for fingerprint comparisons.

Finally, and although it is out of the reach of this article, it is worth mentioning the explosive growth of genomic sequences. Such an impressive work provides an unprecedented opportunity to explore genetic variability and biological function of organisms from a fundamental point. In this line, there has recently been a rapid advance in developing various powerful web servers to formulate DNA/RNA sequences with their feature vectors [70–75].

Key Points

- We survey the techniques applied in DNA fingerprint analysis.
- We have created a benchmark for lane-segmentation and band-detection algorithms.
- Benchmark contains a data set of images and their gold standards.
- Tools to measure the performance of algorithms have been implemented in ImageJ.
- Core algorithms have been implemented in ImageJ, and tested using the benchmark.

Supplementary Data

Supplementary data are available online at https://academic.oup.com/bib.

Funding

This work was partially supported by the Agencia de Desarrollo Económico de La Rioja (ADER) [2013-I-IDD-00123] and Ministerio de Economía y Competitividad [MTM2014-54151-P].

References

1. Kirby LT. *DNA Fingerprinting: An Introduction*. Oxford University Press, New York, 1993.
2. Heras J, Domínguez C, Mata E, *et al*. A survey of tools for analysing DNA fingerprints. *Brief Bioinform*. doi:10.1093/bib/bbv016.
3. Agarwal A, Srinivas C, Pujari AK. Identification of lanes and bands in DNA autoradiogram images. In: *14th Conference of the Biomedical Engineering Society of India*, IEEE, New York, 1995.
4. Wong RTF, Flibotte S, Corbett R, *et al*. LaneRuler: automated lane tracking for DNA electrophoresis gel images. *IEEE Trans Autom Sci Eng* 2010;**7**:706–8.
5. Machado A, Campos M, Siqueira, *et al*. An iterative algorithm for segmenting lanes in gel electrophoresis images. In: *10th Brazilian Symposium of Computer Graphics and Image Processing*, 1997, IEEE, New York, pp. 140–6.
6. Ahmad AR, Zakaria H, Ahmad F, *et al*. Gel electrophoresis image segmentation with kapur method based on particle swarm optimization. In: *5th International Conference on Computational Intelligence, Communication Systems and Networks*, 2013, IEEE, New York, pp. 393–6.
7. Koprowski R, Wróbel Z, Korzyńska A, *et al*. Automatic analysis of 2D polyacrylamide gels in the diagnosis of DNA polymorphisms. *Biomed Eng Online* 2013;**12**(68):1–14.
8. Cheng WZ, Chiao T, Hsin C, *et al*. Comparing lanes in the pulsed-field gel electrophoresis (PFGE) images. In: *23rd Annual EMBS International Conference*, 2001, pp. 2911–13.
9. Bajla I, Holländer I, Kollár M. Novel algorithms implemented in the gel image analysis system GAS2. *Measurement Science Rev* 2003;**3**(2):57–66.
10. Chan YK, Guo SW, Cheng HM, *et al*. Automatic band detection of 1D-gel images. In: *International Conference on Electronics, Communications and Control*, 2011, IEEE, New York, pp. 3586–9.
11. Noor MHM, Hussain Z, Ahmad KA, *et al*. Gel electrophoresis image segmentation with otsu method based on particle swarm optimization. In: *7th International Colloquium on Signal Processing and its Applications*, 2011, IEEE, New York, pp. 426–9.
12. Vauterin L, Vauterin P. Integrated Databasing and Analysis. In: *Molecular Identification, Systematics, and Population Structure of Prokaryotes*. Springer-Verlag, Berlin, 2006.
13. Akbari A, Albregtsen F. Automatic lane detection and separation in one dimensional DNA gel images. In: *4th International Conference on Bioinformatics of Genome Regulation and Structure*, 2004, Rusian Academy of Sciences, Novosibirsk, pp. 22–5.

14. Ismail I, Eltaweel GS, Nassar H. Bands detection and lanes segmentation in DNA fingerprint images. *J Inf Comput Sci* 2014;**9**(4):243–51.
15. Sternberg SR. Biomedical image processing. *Computer* 1983;**16**(1):22–34.
16. Bailey D, Christie BC. Processing of DNA and protein electrophoresis gels by image analysis. In: *2nd New Zealand Conference on Image and Vision Computing,* 1994, Massey University, New Zealand, pp. 221–8.
17. Maramis C, Delopoulos A. Efficient quantitative information extraction from PCR-RFLP gel electrophoresis images. In: *International Conference on Pattern Recognition,* 2010, IEEE, New York, pp. 2560–3.
18. Ye X, Suen C, Cheriet M, *et al.* A Recent development in image analysis of electrophoresis gels. In: *Vision Interface'99,* 1999, Canadian Image Processing and Recognition Society, Quebec, pp. 432–8.
19. Caridade CMR, Marcal ARS, Mendonca T, *et al*. Automatic information extraction from gel electrophoresis images using GEIAS. In: *7th International Conference on Image Analysis and Recognition, Vol. 6112 of LNCS,* 2010, Springer, Berlin, pp. 185–94.
20. Moreira B, Sousa A, Mendonca A, *et al*. Automatic lane segmentation in TLC images using the continuous wavelet transform. *Comput Math Methods Med* 2013;**2013**:218415.
21. Park SC, Lee S. Lanes detection in PCR gel electrophoresis images. In: *11th IEEE International Conference on Computer and Information Technology,* 2011, IEEE, New York, pp. 306–13.
22. Wang D, Keller J, Carson C. Pulsed-field gel electrophoresis pattern recognition of bacterial DNA: a systemic approach. *Pattern Anal Appl* 2001;**4**:244–55.
23. Skutkova H, Vitek M, Krizkova S, *et al*. Preprocessing and classification of electrophoresis gel images using dynamic time warping. *Int J Electrochem Sci* 2013;**8**:1609–22.
24. Akbari A, Albregtsen F, Jakobsen K. Automatic lane detection and separation in one dimensional gel images using continuous wavelet transform. *Anal Methods* 2010;**2**:1360–71.
25. Akbari A, Albregtsen F. Automatic segmentation of DNA bands in one dimensional gel images produced by hybridizing techniques. In: *26th Annual International Conference of the IEEE EMBS,* 2004, IEEE, New York, pp. 2852–5.
26. Zerr T, Henikoff S. Automated band mapping in electrophoretic gel images using background information. *Nucleid Acids Res* 2005;**3**(9):2806–12.
27. Lee J, Huang C, Wang N, *et al*. Automatic DNA sequencing for electrophoresis gels using image processing algorithms. *J Biomed Sci Eng* 2001;**4**:523–8.
28. Labyed Y, Kaabouch N, Schultz RR, *et al*. An improved 1-D Gel electrophoresis image analysis system. In: *Advances in Computational Biology, Vol. 680 of Advances in Experimental Medicine and Biology*, 2010, Springer, New York, pp. 609–17.
29. Bajla I, Holländer I, Burg K. Improvement of electrophoretic gel image analysis. *Measurement Science Rev* 2001;**1**(1):5–10.
30. Bajla I, Holländer I, Burg K, *et al.* A novel approach to quantitative analysis of electrophoretic gel images of DNA fragments. In: *IEEE International Symposium on Biomedical Imaging,* 2002, IEEE, New York, pp. 899–902.
31. Bajla I, Holländer I, Fluch S, *et al*. An alternative method for electrophoretic gel image analysis in the GelMaster software. *Comput Methods Programs Biomed* 2005;**77**:209–31.
32. Park SC, Han TH, Kim SH, *et al*. Lane detection and tracking in PCR gel electrophoresis images. *Comput Electron Agric* 2012;**83**:85–91.
33. Khodabakhshi S, Hassanpour H. Automatic lane extraction in hemoglobin and serum protein electrophoresis using image processing. *J Adv Comput Res* 2012;**3**(4):25–31.
34. Caridade CMR, Margal AS, Mendonga T, *et al*. An automatic Method to identify and extract information of DNA bands in gel electrophoresis image. In: *31st Annual International Conference of the IEEE EMBS,* 2009, IEEE, New York, pp. 393–6.
35. Pavel A, Vasile C. PyElph—a software tool for gel images analysis and phylogenetics. *BMC Bioinformatics* 2012;**13**(9). doi:10.1186/1471-2105-13-9.
36. Sotaquirá M. On the use of distance maps in the analysis of 1D DNA gel images. In: *International Conference on Digital Image Processing,* 2009, IEEE, New York, pp. 172–6.
37. Labyed Y, Kaabouch N, Schultz RR, *et al*. Automatic segmentation and band detection of protein images based on the standard deviation profile and its derivative. In: *IEEE International Conference on Electro/Information Technology,* 2007, IEEE, New York, pp. 577–82.
38. Pot B, Gillis M, Hoste B, *et al*. Intra- and intergeneric relationships of the genus Oceanospirillum. *Int J Syst Evol Microbiol* 1898;**39**(1):23–34.
39. Vauterin L, Vauterin P. Computer-aided objective comparison of electrophoresis patterns for grouping and identification of microorganisms. *Eur Microbiol* 1992;**2**:37–41.
40. Dice LR. Measures of the amount of ecologic association between species. *Ecology* 1945;**26**(3):297–302.
41. Jaccard P. Nouvelles recherches sur la distribution florale. *Bulletin de la Société vaudoise des sciences naturelles* 1908;**44**:223–70.
42. Ochiai A. Zoogeographic studies on the soleoid fishes found in Japan and its neighbouring region. *Bull Jpn Soc Sci Fish* 1957;**22**:526–30.
43. Pearson K. Notes on regression and inheritance in the case of two parents. *Proc R Soc Lond* 1895;**58**:240–2.
44. Fullaondo A, Vicario A, Aguirre A, *et al*. Quantitative analysis of two-dimensional gel electrophoresis protein patterns: a method for studying genetic relationships among Globodera pallida populations. *Heredity* 2001;**87**:266–72.
45. Fernández-Lozano C, Seoane JA, Gestal M, *et al*. Texture classification using feature selection and kernel-based techniques. *Soft Comput* 2015;**19**:2469–80.
46. Anderberg MR. *Cluster Analysis for Applications*. Academic Press, London, 1973.
47. Saitou N, Nei M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol* 1987;**4**:406–25.
48. Rauss PJ, Phillips PJ, Moon H, *et al*. The FERET evaluation methodology for face recognition algorithms. *IEEE Trans Pattern Anal Mach Intell* 2000;**22**(10):1090–104.
49. Martin D, Fowlkes C, Tal D, *et al*. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *8th IEEE International Conference on Computer Vision, Vol. 2,* 2001, IEEE, New York, pp. 416–23.
50. Nattkemper TW, Twellmann T, Schubert W, *et al*. Human vs. machine: evaluation of fluorescence micrographs. *Comput Biol Med* 2003;**33**(1):31–43.
51. Gelasca ED, Obara B, Fedorov D, *et al*. A biosegmentation benchmark for evaluation of bioimage analysis methods. *BMC Bioinformatics* 2009;**10**:368.
52. Schneider CA, Rasband WS, Eliceiri KW. NIH Image to ImageJ: 25 years of image analysis. *Nat Methods* 2012;**9**:671–5.
53. Liu B, Xu J, Lan X, *et al*. iDNA-Prot|dis: identifying DNA-binding proteins by incorporating amino acid distance-pairs and reduced alphabet profile into the general pseudo amino acid composition. *Plos One* 2014;**9**:e106691.

54. Liu B, Fang L, Liu F. iMiRNA-PseDPC: microRNA precursor identification with a pseudo distance-pair composition approach. *J Biomol Struct Dyn* 2015;**3**:1–13.
55. Chen W, Feng PM, Lin H. IRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition. *Nucleic Acids Res* 2013;**41**:e68.
56. Guo SH, Deng EZ, Xu L, *et al*. iNuc-PseKNC: a sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition. *Bioinformatics* 2014;**30**:1522–9.
57. Lin H, Deng EZ, Ding H, *et al*. iPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. *Nucleic Acid Res* 2014;**42**:12961–72.
58. Qiu WR, Xiao X. iRSpot-TNCPseAAC: identify recombination spots with trinucleotide composition and pseudo amino acid components. *Int J Mol Sci* 2014;**15**:1746–66.
59. Liu B, Fang L, Wang S, *et al*. Identification of microRNA precursor with the degenerate K-tuple or Kmer strategy. *J Theo Biol* 2015;**385**:153–9. doi:10,1016/j.jtbi.2015.08.025.
60. Jia J, Liu Z, Xiao X. iPPI-Esml: an ensemble classsifier for identifying the interactions of proteins by incorporating their physicochemical properties and wavelet transforms into PseAAC. *J Theor Biol* 2015;**377**:47–56.
61. Liu Z, Xiao X, Qiu WR. iDNA-Methyl: identifying DNA methylation sites via pseudo trinucleotide composition. *Anal Biochem* 2015;**474**:69–77.
62. Chen W, Feng P, Ding H. IRNA-Methyl: identifying N6-methyladenosine sites using pseudo nucleotide composition. *Anal Biochem* 2015;**490**:26–33. doi:10.1016/j.ab.2015.08.021.
63. Fawcett T. An introduction to ROC analysis. *Pattern Recognit Lett* 2006;**27**:861–74.
64. García S, Fernández A, Luengo J, *et al*. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. *Inf Sci* 2010;**180**(10):2044–64.
65. García S, Fernández A, Luengo J., *et al*. A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. *Soft Comput* 2009;**13**(10):959–77.
66. Demsar J. Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 2006;**7**:1–30.
67. Sousa AV, Aguiar R, Mendonca A, *et al*. Automatic lane and band detection in images of thin layer chromatography. In: *Image Analysis and Recognition, Vol. 3121 of LNCS*, 2004, Springer, Berlin, pp. 158–65.
68. Moreira B, Sousa A, Mendonca A, *et al*. Automatic lane detection in chromatography images. In: *Image Analysis and Recognition, Vol. 7325 of LNCS*, 2012, Springer, Berlin, pp. 180–7.
69. Heras J, Domínguez C, Mata E, *et al*. GelJ — a tool for analyzing DNA fingerprint gel images. *BMC Bioinformatics* 2015;**16**:270.
70. Chen W, Lei TY, Jin DC. PseKNC: a flexible web-server for genetic pseudo K-tuple nucleotide composition. *Anal Biochem* 2014;**456**:53–60.
71. Chen W, Zhang X, Brooker J. PseKNC-general: a cross-platform package for generating various modes of pseudo nucleotide compositions. *Bioinformatics* 2015;**31**:119–20.
72. Liu B, Liu F, Fang L, *et al*. repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects. *Bioinformatics* 2015;**31**:1307–9.
73. Liu B, Liu F, Fang L. repRNA: a web server for generating various feature vectors of RNA sequences. *Mol Genet Genomics* 2015. doi:10.1007/s00438-015-1078-7.
74. Liu B, Liu F, Wang X, *et al*. Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic Acids Res* 2015;**43**:W65–71.
75. Chen W, Lin H, Chou KC. Pseudo nucleotide composition or PseKNC: an effective formulation for analyzing genomic sequences. *Mol Biosyst* 2015;**11**:2620–34.