A novel ensemble feature selection method for pixel-level segmentation of HER2 overexpression

Ana Aguilera · Raquel Pezoa · Andrea Rodríguez-Delherbe

Received: 19 August 2021 / Accepted: 6 May 2022 / Published online: 28 May 2022 © The Author(s) 2022

Abstract
Classifying histopathology images on a pixel-level requires sets of features able to capture the complex characteristics of the images, like the irregular cell morphology and the color heterogeneity on the tissue aspect. In this context, feature selection becomes a crucial step in the classification process such that it reduces model complexity and computational costs, avoids overfitting, and thereby it improves the model performance. In this study, we propose a new ensemble feature selection method by combining a set of base selectors, classifiers, and rank aggregation methods, aiming to determine from any initial set of handcrafted features, a smaller set of relevant color and texture pixel-level features, subsequently used for segmenting HER2 overexpression on a pixel-level, in breast cancer tissue images. We have been able to significantly reduce the set of initial features, using the proposed ensemble feature selection method. The best results are obtained using $\chi^2$, Random Forest, and Runoff as the based selector, classifier, and aggregation method, respectively. The classification performance of the best model trained on the selected features set results in 0.939 recall, 0.866 specificity, 0.903 accuracy, 0.875 precision, and 0.906 F1-score.

Keywords Feature selection · Rank aggregation · Pixel-level classification · Histopathology images · HER2 overexpression segmentation

Introduction

Machine learning methods have been fundamental for achieving successful outcomes in the analysis of histopathology images. Classification and image segmentation are frequent tasks in biomedical image analysis, and the computational algorithms used for performing these analysis must take into account diverse challenges including image characteristics complexity produced by high morphological variance of cancer cells or by the color heterogeneity of stained tissues. Feature selection is the process of identifying the most relevant set of features and is a fundamental preceding step for many machine learning algorithms, considering that irrelevant and redundant features used in the later training may result in high computational complexity and performance reduction. This work is focused on handcrafted features, i.e., diverse characteristics are extracted from the images using techniques from the image processing field. Even though during the last years models based on handcrafted features have drawn less attention in comparison to deep learning-based systems, handcrafted features can be used when there are reduced datasets and when interpretability is required.

Traditionally, the histopathology assessment has been performed by pathologists through the visual examination of immunohistochemically stained tissue sections using an optical microscope. However, the advances in virtual microscopy, including whole slide imaging (WSI) technology, have promoted the emergence of high-resolution tissue images. Hence, the automatic image analysis of the digitized tissues becomes wider, with the potential to increase the accu-
Fig. 1 Examples of breast cancer tissue image stained with IHC. Images (A) and (D) are images of 500 × 1000 pixels, and are tissue samples sections (two different patients) that were scanned at 40x magnification, with 0.23 [μm/pixel]. The structures of interest are cell membranes with HER2 overexpression which are stained in brown color (DAB). Nuclei are stained in a blue color (hematoxylin) to give contrast. Both tissues samples were digitized with the same scanner, but they have differences in brightness, contrast, and brown color intensity. The images (B) and (E) zoom (B) Shows the HER2 overexpression with a strong brown color on the cell membrane and (E) highlights two cells, which are visualized by the blue color of their nuclei, but no brown membrane is observed. This happens here because those cells do not have HER2 overexpression. (C) and (F) Correspond to the gold standard, i.e., the pixel-level manual identification performed by a pathologist. White pixels represent membranes with HER2 overexpression.

Accuracy, reproducibility, and objectivity of the histopathology assessment.

HER2 (Human Epidermal growth factor Receptor 2) is a protein that is normally expressed on the cell membrane surface of the epithelia of various organs [1]. This protein plays a role in the normal cell growth development, but when the HER2 is overexpressed, i.e., there are too many copies of the protein, cells multiply and grow more rapidly than usual, promoting carcinogenesis in diverse types of cancer [2]. Immunohistochemistry (IHC) is an histological procedure that is frequently used to identify the HER2 overexpression. The hematoxylin and diaminobenzidine (DAB) are chemical compounds—often referred to as stains—used during the IHC procedure to visualize cell nuclei and membranes. The hematoxylin is used as a counterstain that produces blue-violet staining of nuclear elements. The DAB stain produces a dark brown reaction on the cell membrane only when there is HER2 overexpression, as shown in Fig. 1.

Pathologists classify the observed tissue into four categories (0, 1+, 2+ or 3+) according to the guidelines of the American Society of Clinical Oncology and College of American Pathologists (ASCO/CAP) [3]. Only HER2 positive patients are suitable to receive the HER2-targeted treatment, thus, automatic and objective computational methods are important support tools for the pathologists. The tissue assessment, as a hand labour, can be time-consuming and prone to inconsistencies, specially when the pathologist must handle large volume of samples. Literature reports 20% of discordance in HER2 overexpression assessment [4], which directly impact in the therapeutic decision.

Keeping into consideration the previous issues and also the selection of the most relevant features is fundamental on the classification process, this study proposes a feature selection technique based on a combination of different selectors or “ensemble of selectors” for addressing the HER2 overexpression identification, as an image segmentation problem, using a pixel-level classifier. This ensemble feature selection approach includes filters and embedded methods that are used as base selectors to add heterogeneity during the selection, considering that there is no best feature selection algorithm that ensures the optimal outcome. The base selectors discard the irrelevant or redundant features in the context of HER2 overexpression segmentation, without causing a significant loss in information, reducing the model complexity, and hence avoiding overfitting. Also, an aggregation ranking stage is included to combine the different ranked lists provided by each base selector, and hence, assigning an overall score into one ranking list, and identifying a set of optimal features in this context. The novelty of this study includes (i) the combination of filter and embedded feature selection methods and aggregation ranking techniques, (ii) the selection of a relevant set of color and texture pixel-level features in the context of histopathological image analysis, that contributes on the (iii) quantitative and objective identification of HER2 overexpression. We verify the impact of the selected feature set using different pixel-level supervised classifiers.
and the proposed model achieves a true positive rate of 0.939, i.e., it is able to identify on average a 94% of the HER2 over-expressed pixels, reducing the initial set of 210 features to a set containing 65 features. Note that this work is focused on a set of 210 color and texture features, but this set can be increased to include any other kind of pixel-level feature.

The literature describes diverse approaches that assess feature selection in biomedical image analysis, but to the best of our knowledge, our study is the first one to propose an ensemble method for selecting features for classifying HER2 overexpression at pixel-level. The paper organization is described as follows. The next section describes the related work about HER2 overexpression identification and feature selection, and the subsequent sections present the proposed method, including the computation of color and texture pixel-level features, the base selectors, classifiers, the rank aggregation methods and the performance metrics, respectively. Next, the experiments and results are presented, and in the penultimate section, discussion is given. The final section presents the conclusions and future work.

Related work

Authors have addressed the computational analysis of HER2 overexpression using different machine learning approaches. Masmoudi et al. [5] identify membrane pixels using a linear regression classifier, representing pixels with color information. Ficarra et al. [6] identify the overexpressed membranes using a color deconvolution filter [7] and reconstruct cell membranes using a tessellation-based approach. Tuominen et al. [8] proposed a whole process to classify breast cancer tissue with HER2 overexpression. They also use a color deconvolution filter [7] as the first step to separate overexpressed membranes.

Wdoviak et al. [9, 10] use morphological tools for the membrane analysis, specifically, they use hourglass shape structuring elements for membrane shape detection. Subsequently, their proposal describes the use of diverse linear shape patterns that are combined to represent the HER2 overexpressed membrane shape. There is a significant increase of recent medical image analysis studies applying deep learning based techniques [11]. As for HER2 evaluation of IHC stained microscopic images, Khameneh et al. [12] developed a method based on two techniques: In first place, a SVM classifies image regions to identify epithelial andstromal regions. Then, a convolutional neural network approach is used for segmenting membrane on epithelial regions. Other deep learning approach was proposed by Zakrzewski et al [13] who developed a pipeline for automatically detect HER2 amplification status in Fluorescence in situ Hybridization (FISH) images, using RetinaNet, a CNN approach for object detection. Also, pre-trained CNNs were proposed by Kieffer et al. [14], for classifying histopathological images using features from pre-trained networks including VGG16 and Inception-v3.

Regarding the use of color spaces in immunohistochemistry image analysis, the investigation conducted by Vahid et al. [15] presented an automated approach for extracting proliferative and normal cells, using HSV color spaces clustered with Fuzzy C-means algorithm, to find groups of pixels with homogeneous local textures and gray levels. Another recent application was implemented by Bamford et al. [16], aiming to estimate the expression level of genes in tissue samples by counting the pixels classified as genes, chromosomes or non-genetic material. This application counts with a computer-based system running an user-interface screen where the user is allowed to select cell nuclei later used for segmentation of nucleus and dots counting. Part of the implementation includes the transformation of RGB color space into L*a*b color space, to produce a new image that emphasizes red and black colors, where dots are more easily detected. On the other hand, Syed et al [17] studied intratumoral heterogeneity in HER2-positive breast cancer tissue images in response to therapy, following two approaches for two different types of images. The first study consists of converting images to CIELAB color space, for later use K-Means technique to segment out cell nuclei. The second workflow included converting images to HSV color space and then used Otsu thresholding for segmentation of Ki-67 nuclei. In general, most works implementing color space analysis mainly use RGB, HSV, CMYK, L*a*b* or YCbCr [18–20]. Multiple studies have been conducted to analyze the accuracy of different color spaces to represent histopathological images [21–23], where RGB outperformed the other color spaces. These studies addressed individual color analysis in order to keep only those spaces that better represent the image. Our approach is generally based on the same principle and aims to automate this process. Additionally, each color space describes different color profiles of the images. Here, literature demonstrates that models based on a combination of color spaces have better performance than models based on only one color space [24].

Besides color spaces, many studies in the field of image analysis are focused on identifying image textures for interpreting image properties. One of the most used techniques is gray level co-occurrence matrix (GLCM) [25, 26], which has been widely used for quantifying cellular heterogeneity. Chang et al. [27] extracted GLCM features per image region in order to reflect vascular complexity of different molecular markers, reaching a 58.8% of accuracy. Besides GLCM, Chang formed various different groups of features and performed a Chi-square test in order to compare their performances. In general, GLCM is one of the most used feature extraction techniques for medical images analysis, mainly since it allows to obtain the spatial information of pixel dis-
distribution in an image or image regions \cite{28}. Other related works described their application using Haralick features \cite{29}, gray-level run length matrix (GLRLM), uniform local binary patterns (ULBP) \cite{30}, Law’s texture energy measure \cite{31,32}, bag-of-words, local binary patterns \cite{33,34}, among others.

**Methods**

This study is focused on selecting the relevant features for identifying HER2 overexpression in breast cancer tissue images using an ensemble feature selection method. There are several feature selection techniques in the literature, however, the selection of the best method for a particular task is still an open problem. An ensemble method, also called ensemble learning, is based on combining multiple models instead of a single model to solve a particular problem \cite{35}. For this study we have configured a combination of different methods of feature selection and classifiers that are evaluated using different metrics. Figure 2 summarizes the ensemble method applied to our problem. In the first stage, we compute the features on a pixel-level basis for each image in our dataset. Then, features are placed in a two-dimensional structure where each row is a pixel characterized by all its computed features—each one as a column. These structures are the input to our ensemble method, which is composed of a set of base selectors ($\chi^2$, ANOVA, RandomForest and ExtraTree) and a set of classifiers (RandomForest, Logistic Regression, Support vector machine and k-nearest neighbours). This part of the experimentation consists of taking the features subset selected by each selector on every separated image, pass it to each classifier that selects the features subset with best performance metrics. Since each two-dimensional structure passed through the above process generates a subset of features, an aggregation process is required to compile the optimal subset of features. The aggregation stage uses rank aggregation approaches, like Borda count, Instant-runoff voting, Dowdall, and Average rank, and its output is a ranked list of features for each image independently. Finally, to assess the quality of the obtained feature set, another classification model is built and evaluated upon all of the images pixels (full set of pixels) to identify HER2 overexpression. The method is resumed by the algorithm 1.
The next subsections describe every ensemble component and the metrics used for evaluation:

1. Pixel-level features computation (“Pixel-level feature computation”)
2. Base selectors (“Base selectors”)
3. Classifiers (“Classifiers”)
4. Rank Aggregation (“Rank aggregation”)
5. Metrics for evaluation (“Metrics for model assessment”)

### Pixel-level feature computation

The proposed method is designed to select a set of relevant features—in terms of the final pixel classification performance—from any initial set of handcrafted features, and in this study, pixels of breast cancer tissue images are represented by 210 features of texture and color. Color features correspond to the color channels known as: RGB, CMYK, HSV, and CIEL*a*b* color spaces, and the grayscale image. Also, we use the DAB channel obtained with a color deconvolution filter [7]. Considering that the selection of a color space is fully dependent on the application, we compare the effects of different color spaces sets over the results and its accuracy. Therefore, one of the main motivations of our research is to measure the contribution of diverse color space channels to compute highly representative features.

In this work, texture features are obtained from Gray Level Co-occurrence Matrices (GLCMs), which are the basis for the computation of Haralick texture features [25]. GLCM $P^\theta_d$ is a matrix of relative frequencies between pairs of neighbour pixels in a gray-scale image, meaning that each element in the GLCM matrix corresponds to the number of times a certain combination of gray level pairs appears on the image, normalized by the total number of gray level pairs. Neighbour pixels are determined considering two parameters: a distance $d$ and a direction given by an angle $\theta$.

Let us consider a grayscale image $\mathcal{I}$ with $m \times n$ pixels $(r, s)$, $1 \leq r \leq m$ and $1 \leq s \leq n$, of gray-levels denoted by $\mathcal{I}(r, s) \in \{0, 1, \ldots, N_g - 1\}$, where $N_g = 256$ typically.

For each pixel $(r, s)$ of the image $\mathcal{I}$ we define its $\mathcal{N}_d^\theta(r, s)$-neighbourhoods as the sets:

$$\mathcal{N}_d^\theta(r, s) = \{(r \pm d, s)\},$$

$$\mathcal{N}_d^{45\circ}(r, s) = \{(r - d, s - d), (r + d, s + d)\},$$

$$\mathcal{N}_d^{90\circ}(r, s) = \{(r, s \pm d)\},$$

$$\mathcal{N}_d^{135\circ}(r, s) = \{(r - d, s + d), (r + d, s - d)\}.$$

The coefficient $(i, j)$ of the gray-level co-occurrence $N_g \times N_g$-matrices $P^\theta_d$, for each $\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$, is then defined as the number of occurrences of the pattern “gray-level-$i$, gray-level-$j$” among all $\mathcal{N}_d^\theta(r, s)$-neighbour pixels, for all pixels $(r, s)$ of the image $\mathcal{I}$. More precisely:

$$P^\theta_d(i, j) = \# \{(r, s), (t, v) : 1 \leq r \leq m, 1 \leq s \leq n, (t, v) \in \mathcal{N}_d^\theta(r, s), \mathcal{I}(r, s) = i, \mathcal{I}(t, v) = j\},$$

(1)

where $\#$ denotes the set cardinality. Then, $P^\theta_d$ is normalized by dividing each element of the matrix by the number $R^\theta_d$:

$$R^\theta_d = \sum_{i=0}^{N_g - 1} \sum_{j=0}^{N_g - 1} P^\theta_d(i, j).$$

Therefore, the normalized coefficients of $P^\theta_d$, can be considered as the probability that a pixel with gray-intensity value $i$ will be neighbour to a pixel with gray-intensity value $j$ along the direction $\theta$ at distance $d$. Figure 3a–d shows a diagram of the computation of a GLCM. The Haralick features are not rotation invariant [36]. Consequently, as a way to diminish the rotation issue, we compute four normalized gray level co-occurrence matrices, considering the four main directions ($\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$), and a distance $d = 1$. Then, we compute an average matrix:

$$\mathcal{T}_d = \frac{1}{4} \left( P^0_d + P^{45\circ}_d + P^{90\circ}_d + P^{135\circ}_d \right).$$

(2)
Gray Level Co-occurrence Matrix. **a** Different distances \(d\), and **b** depicts the most common used directions. **c** A toy image with \(5 \times 5\) pixels and only 4 gray levels. **d** The GLCM associated with the image, which is a \(4 \times 4\) matrix, where \(P^d_{\theta}(0,0) = 4\) represents the number of occurrences (pink arrows) that neighbour pixels have the gray level values \((0,0)\), \(P^d_{\theta}(0,1) = 2\) (light blue arrows) is the number of occurrences that neighbour pixels have the gray level values \((0,1)\), and so on.

**e** Representation of a sub-image \(S^{(r,s)}\) associated with a pixel \((r,s)\). **f** The average GLCM \(\overline{P}^d_{\theta}\) is computed from the four different GLCMs associated to the sub-image \(S^{(r,s)}\), to compute the Haralick features in a pixel-level. In this work, we use \(d = 3\), i.e., \(7 \times 7\) sliding window

Haralick features are computed from this average matrix. In addition, for pixel-level classification, we need Haralick features on a per pixel basis. However, the GLCM \(P^d_{\theta}\) matrices and their Haralick features are given on a per image basis. Thus, we calculate GLCM’s using a small sliding window, associated to each pixel, moving over the image. For each pixel \((r,s)\), we define a sub-image \(S^{(r,s)}\) of \(k \times k\) pixels with the pixel \((r,s)\) as the central pixel (see Fig. 3-(e)). Then, the pixel-level average matrix \(\overline{P}^d_{\theta}\) is calculated (see Fig. 3-(f)). With this procedure, we compute the Haralick features on a per-pixel basis, which also yields the texture images associated with each of the features. The size of the sliding window used in this work is \(7 \times 7\) pixels, and this size was selected considering that the images were obtained at \(40x\) magnification with \(0.23 [\mu m/pixel]\), and a cell membrane is a fine structure with an approximate width of \(1.7 \mu m\) corresponding to \(7\) pixels. Therefore, with this procedure, we compute 13 Haralick features (described in Appendix B), the sum average and sum variance, and the average of the GLCM matrix, on a per pixel basis, for each color feature, which generates 195 texture features, as described in Table 1.

Note that we use the \(7 \times 7\) sliding window or kernel only on the texture feature to obtain features on a pixel-level. In the case of the color features, we intend to keep the original features because HER2 overexpression, i.e., the brown color present only on the membrane cell, can be very weak, sometimes even being difficult to get detected by the pathologists. Hence, applying a kernel to reduce the feature or a kernel of image blurring can eliminate critical information to identify HER2 overexpression.

**Base selectors**

Feature selection methods include a set of techniques aiming to reduce the number of input variables, keeping the most relevant set of features, and making a model more simple to interpret. Some predictive modelling problems have a large number of variables, and the performance of some models can be affected by including irrelevant input variables concerning to the target variable. Hence, it is desirable to reduce the number of input variables to both reducing computational cost of modeling and, in some cases, improving the model performance. Ang et al. [37] resume the benefits of feature selection as: improvements to prediction performance, understandability, scalability, and generalization capability of the classifiers; reduction of computational complexity and
Table 1 Color and texture pixel-level features

| Type               | Color Space                      | Feature                                      |
|--------------------|----------------------------------|----------------------------------------------|
| Color              | RGB                              | 3 Channels: R, G, and B                     |
|                    | CMYK                             | 4 Channels: C, M, Y, and K                   |
|                    | CIEL*a*b*                        | 3 Channels: L*, a* and b*                    |
|                    | HSV                              | 3 Channels: H, S, and V                      |
| Grayscale image    |                                  | 1 Channel: grey                             |
| H/DAB stains       |                                  | 1 Channel: DAB                              |
|                    |                                  | 15 Color features                           |

| Type               | Haralick Feature                     | Description                                           |
|--------------------|---------------------------------------|-------------------------------------------------------|
| Texture            | Angular second moment, GLCM average,  | 13 Haralick features computed on                     |
|                    | contrast, correlation, variance,      | a pixel-level per to each color                       |
|                    | difference entropy, difference        | feature                                               |
|                    | entropy, inverse different moment,    |                                                       |
|                    | sum average, sum entropy, GLCM        |                                                       |
|                    | standard deviation, and sum variance  |                                                       |
|                    | (see App. Appendix B)                 |                                                       |
|                    |                                      | 195 Texture features                                  |

Storage in providing faster and more cost-effective model. The available selection methods can be broadly categorized into three groups: filters, wrappers, and embedded [38], which are briefly described as follows:

1. Filters. Filter methods select the subset of features as a pre-processing step, independent of the chosen classifier. A typical filter algorithm consists of two steps: (i) it ranks features based on certain criteria. Feature evaluation could be either univariate or multivariate. In the univariate scheme, each feature is ranked independently of the feature space, while the multivariate scheme evaluates features in an batch way; (ii) the features with highest rankings are chosen to induce classification models. Filter methods are computationally simple and fast, they can handle extremely large-scale datasets [39]. However, their main disadvantage in comparison to the other methods, is that they do not retrieve feedback from the learning performance using the selected features.

2. Wrappers. Wrapper models utilize a specific classifier to evaluate the quality of selected features, and offer a simple and powerful way to address the problem of feature selection, regardless of the chosen learning machine [38]. A wrapper consists of three steps: searching a subset of features; evaluating the selected subset of features by the performance of the classifier, and; repeating the previous steps until the desired quality is reached. The disadvantage of this method is that the predefined classifier works as a black box and it is inevitably biased to the predefined classifier. It is also very computationally expensive.

3. Embedded. The selection process relies on the intrinsic capacity of certain classification algorithms to assign weights to the features, without a systematic search through different candidate subsets [40]. This method has the advantages of (i) wrapper models—they include the interaction with the classification model- and (ii) filter models—they are far less computationally intensive than wrapper methods. Since no single selection algorithm into these groups seems to be capable of ensuring optimal results in terms of predictive performance, robustness and stability, researchers have increasingly explored the effectiveness of ensemble approaches involving the combination of different selectors [40–43]. The concept of ensemble learning has been applied to improve the performance of feature selection [43] and is a prolific field in machine learning based on the assumption that combining the output of multiple models is better than using a single model, and it usually provides good results. Usually, it is employed for classification, but it can also be used to improve other disciplines such as feature selection [44]. The ensemble aims to exploit the strengths of the three different groups (Filters, Wrappers and Embedded) and at the same time to overcome their weaknesses [40]. The idea is to integrate feature selection and ensemble learning in two senses: (i) to use ensemble learning for feature selection, generating a good feature subset by combining multiple feature subsets, and (ii) to utilize feature selection for ensemble learning, that is, to use different feature subsets to construct an ensemble of diverse-based classifiers.

In this scenario, we propose an ensemble learning for feature selection using two methods from filters and two from embedded methods. The former is based on univariate analysis and the latter is based on multivariate analysis [45]. Univariate analysis works by selecting the best features based on univariate statistical tests. Each feature is
compared to the target variable to see whether there is any statistically significant relationship between them. Multivariate analysis computes multivariate statistics for feature ranking considering the dependencies between the features when calculating their scores. It considers the whole groups of variables together. Tree-based algorithms and models are well-established algorithms that not only offer good predictive performance but can also provide a way to feature selection called feature importance. They are multivariate feature selection [45]. Feature importance indicates which variables are more important in making accurate predictions on the target variable/class. In other words, it identifies which features are the most used by the machine learning algorithm to predict the target.

The feature selection methods used in this work are described as follows:

- **Filters**
  - ANOVA F value. Analysis of variance (ANOVA) is addressed as a statistical test for detecting differences in groups of data means when there are: one parametric dependent variable and one or more independent variables. ANOVA uses F-tests to statistically test the equality of means [46]. In the feature selection context, for each feature Filter ANOVA performs an analysis of variance where the class variable is explained by the feature. The value of the statistic is used as the score. The higher the statistic, the more difference between the mean values of the corresponding feature among classes [45].
  - Chi-squared stats. Score used to select the $n$ features with the highest values for the $\chi^2$ statistic test from X [47]. For each feature, filter $\chi^2$ performs an independence test between a dichotomized transformation of this feature and the class variable. The value of the statistic is used as the score [45].

- **Embedded**
  - RandomForest classifier. A random forest (RF) is a meta-estimator that fits many decision tree classifiers on various sub-samples of the dataset, it uses averaging to improve the predictive accuracy and control over-fitting, and it has its built-in feature selection methods. Therefore, when training a tree, it is possible to compute how much each feature decreases the impurity. The more a feature decreases the impurity, the more important the feature is [48].
  - Extra-trees classifier. This classifier implements a meta-estimator that fits several randomized decision trees (also Known as extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control the overfitting.

Feature importance is an inbuilt class that gives a score for each feature from data; the higher the score, the more relevant the feature is towards your output variable [49].

**Classifiers**

To evaluate the predictive ability of each feature selection, we have explored four supervised learning techniques for data classification. This choice was made taking into consideration flexibility, numerical data processing, hyperparameter tuning, size of data, and amount of features. The four analyzed classifiers were

- Logistic regression (LR) is a classification algorithm used to assign observations to a discrete set of classes. It is a predictive analysis algorithm and is based on the concept of likelihood. Logistic Regression uses a more complex cost function than linear regression, which can be defined as the ‘Sigmoid function’, also known as the ‘logistic function’ instead of a linear function. This is suitable for binary and multi-class classification [50].
- K-Nearest Neighbours (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. It assumes the principle that similar things exist in close proximity. In other words, similar things are near to each other [51].
- Support vector machines (SVM) is a machine learning method widely used for solving classification and regression problems. The main idea of the SVM algorithm is to build an optimal hyperplane with the largest margin that best separates the data into classes [52]. The SVM is a popular algorithm due to its high generalization capability together with its ability to find global and non-linear classification solutions.
- Random Forest (explained before).

**Rank aggregation**

As described in “Base selectors”, we use four different feature selection techniques: $\chi^2$, ANOVA, ExtraTrees, and Random Forest. The output of each technique is represented by a ranked list; thus, we use aggregation methods to combine the different feature rankings into one ranked list.

Literature describes diverse rank aggregation approaches. In this work, we analyze Borda count, Instant-runoff Voting, Dowdall and Average rank methods. To the best of our knowledge, there are no studies done on feature selection based on rank aggregation methods in the HER2 pixel-level segmentation area.
Borda count method

The Borda count method was formalized by Jean-Charles de Borda in 1781 [53] produces an aggregated ranked feature list, computing the Borda score of each feature according the following steps:

1. Let us consider $k$ feature ranking lists $L_k$, and each list has $N_k$ tuples containing $(r_{ik}, f_i)$, $i = 1 \ldots N_k$, where $r_i$ is the rank of feature $f_i$, sorted in descending order.

2. For each ranking list $L_k$, compute the new score of each feature $f_i$ considering that the first ranked feature is scored with the value $s_{i,j} = N_k$, the second ranked feature is scored with $N_k - 1$, until the last ranked feature is scored with 1. Therefore, a new ranking list $B_k$ is generated, containing the tuples $(s_{i,k}, feature_i)$.

3. Finally, the Borda rank $b_i$ of each feature $f_i$ is computed as

$$b_i = \sum_{j=1}^{k} s_{i,j}.$$  

In this work, we use the Borda count method, using 9 features ranked lists $L_1, \ldots, L_9 (k = 9)$. Each list is obtained using a feature selection method on each image of our dataset. Therefore, Borda count method generates a general ranking list.

Dowdall

Dowdall method [54] is a modification of Borda count method. The main difference is that in Borda count method, the score assigned to each feature varies with the size of the rankings, while for Dowdall method, the score for an alternative is always constant and is the reciprocal of its position.

Considering the same steps described in the Borda count method, on each ranking list $L_k$, the new score for the first ranked feature is $s_{i,j} = 1$, then the second ranked feature is scored with $\frac{1}{2}$, and so on until the last ranked feature, with value $\frac{1}{N_k}$. Therefore, Dowdall ranking $d_i$ for each feature, is calculated as

$$d_i = \sum_{j=1}^{k} d_{i,j}.$$  

Instant runoff voting

Instant runoff voting [55] is an electoral system that can be adapted to create an aggregated ranking list. The aggregation is performed by selecting the winner feature as the one with more than 50% of the scores (considering all the ranking lists). If there is no feature with a strict majority, the feature with the lowest score is deleted, and that feature score is redistributed to the features next choice. This is an iterative process that continues until a feature has a majority (over 50%). Therefore, the winners are sorted in descending order to create a consensus ranking.

Average rank method

The aggregated ranking is generated by computing the average score of each feature. Given the $k$ ranked lists $L_1, L_2, \ldots, L_k$, for each feature $f_i \in L_j$, $j = 1, \ldots, k$, with a score $s_{i,j}$, corresponding to the score of feature $f_i$ in the ranking list $R_j$, the average score $a_i$ of feature $f_i$ is computing by

$$a_i = \frac{1}{k} \sum_{j=1}^{k} s_{i,j}.$$  

The new aggregated list is generated by sorting in descending order based on their average rank.

Metrics for model assessment

The confusion matrix [56] is the base for the evaluation of the classifier’s performance. In the case of binary classification, the confusion matrix is a $2 \times 2$ matrix, with four possible outcomes: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). To evaluate the performance of the feature selection methods combined with different classifiers, we use the following performance metrics:

- Recall or True Positive Rate, defined as the fraction of samples from a class which are correctly predicted by the model:

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (3)$$

- Specificity or True Negative Rate:

$$\text{Specificity} = \frac{TN}{TN + FP}. \quad (4)$$

- Accuracy defined as the number of correct predictions divided by the total number of predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (5)$$

- Precision:

$$\text{Precision} = \frac{TP}{TP + FP}. \quad (6)$$
The dataset

A total of 9 images were used in this study, containing $9 \times 10^6$ pixels that were manually labeled by a pathologist. The images correspond to regions of interest selected by a pathologist from a paraffin-embedded breast cancer tissues. The samples were obtained from the archives of the Departamento de Patología, Hospital Clínico of the Universidad de Chile, and were scanned using a NanoZoomer-XR C12000 whole slide imaging scanner (Hamamatsu Photonics K.K.) at a magnification of 40x and a resolution of 0.23 $\mu$m/pixel, at SCIAN Lab (www.scian.cl).

In this work, pixel segmentation is assessed as a binary classification problem; therefore, the image pixels are divided in two classes: “HER2” (positive class) or “non-HER2” (negative class). HER2 class is very small compared to the non-HER2 class (percentages of HER2 pixels of our dataset vary from 1.12% to 17.33%) which produces an imbalanced classification problem. As described in “Introduction”, HER2 tissues are categorized into four categories: 0, 1+, 2+, and 3+, then the prefix of the name of each image indicates the class it belongs to. Although in this work we do not perform tissue classification, the number of HER2 pixels directly depends on the category that the image belongs to, which also produces an impact on the pixel classification outcomes.

It is worth to mention that there is a lack of datasets on HER2 overexpression segmentation, considering that the generation of pixel-level labels requires manual annotation of histopathological images, which is a highly time-consuming task. Literature reports public datasets on HER2 [57,58], but the images of these datasets are only labeled into the 0, 1+, 2 and 3+ categories, without pixel-level manual annotation.

### Experimental results

#### The dataset

A total of 9 images were used in this study, containing $9 \times 10^6$ pixels that were manually labeled by a pathologist. The images correspond to regions of interest selected by a pathologist from a paraffin-embedded breast cancer tissues. The samples were obtained from the archives of the Departamento de Patología, Hospital Clínico of the Universidad de Chile, and were scanned using a NanoZoomer-XR C12000 whole slide imaging scanner (Hamamatsu Photonics K.K.) at a magnification of 40x and a resolution of 0.23 $\mu$m/pixel, at SCIAN Lab (www.scian.cl).

In this work, pixel segmentation is assessed as a binary classification problem; therefore, the image pixels are divided in two classes: “HER2” (positive class) or “non-HER2” (negative class). HER2 class is very small compared to the non-HER2 class (percentages of HER2 pixels of our dataset vary from 1.12% to 17.33%) which produces an imbalanced classification problem. As described in “Introduction”, HER2 tissues are categorized into four categories: 0, 1+, 2+, and 3+, then the prefix of the name of each image indicates the class it belongs to. Although in this work we do not perform tissue classification, the number of HER2 pixels directly depends on the category that the image belongs to, which also produces an impact on the pixel classification outcomes.

It is worth to mention that there is a lack of datasets on HER2 overexpression segmentation, considering that the generation of pixel-level labels requires manual annotation of histopathological images, which is a highly time-consuming task. Literature reports public datasets on HER2 [57,58], but the images of these datasets are only labeled into the 0, 1+, 2 and 3+ categories, without pixel-level manual annotation.

### Experiment settings

An important issue to consider with the algorithms in machine learning is to determine the parameters to be considered in their execution. This is due to the parameters selection impact the performance of the algorithms executions and the metrics resulting. We have split the estimation of these parameters in two: one of them for the feature selection algorithms and the other one for the classifiers.

#### Hardware

The first set experiments corresponding to the generation of the 9 aggregated ranking lists were executed in an Intel(R) Core(TM) workstation equipped with i7-8700 CPU (6 main processors and 12 logical processors), 32GB RAM and 2 disks of 256 GB SSD and 4TB HDD. The second set of experiment related to the generation of an aggregated ranked

### Table 2 Feature selection hyperparameters

| Feature selection method | Hyperparameters |
|--------------------------|-----------------|
| RandomForest            | n_estimators = 100, max_depth = 2, random_state = 0 |
| ANOVA                   | k = 100         |
| $\chi^2$                | k = 65          |
| ExtraTrees              | n_estimators = 100, random_state = 0 |

#### Table 3 Classifier hyperparameters

| Classifier      | Hyperparameters |
|-----------------|-----------------|
| RandomForest    | {n_estimators: [100, 325, 550, 775, 1000], min_samples_split: [2, 5, 10], max_features: [auto, sqrt], max_depth: max_depth, min_samples_leaf: [1, 2, 4]} |
| Logistic Regression | {C:[1.e-03, 1.e-02], class_weight:(balanced, None), max_iter: [1,000,000, 5,000,000], solver:(lbfgs, sag)} |
| SVM             | {kernel:rbf, C:[1, 10], gamma:[scale]} |
| Kneighbors      | {algorithm:(auto, ball_tree, kd_tree), leaf_size:[30,50], p:[1,2,3], n_neighbors:[2,4,8,16]} |

A crucial parameter for the feature selection algorithms is the $k$ parameter. This parameter indicates the number of features to be selected or retained. Even though, the algorithms return a ranked list of all features, the value of $k$ fixes the size of the subset selected. To estimate it we have tested it with different values and used two classifiers to evaluate the resulting metrics. Thus, we assigned it values within the range of [10, 100] with a step of 5, embedded with KNN and Logistic Regression classifiers. The best values for $k$ were between 65 and 100 features. Table 2 summarizes this result. When RandomForest and ExtraTrees are used, the top $k$ value is considered as the n_estimators, and it was set as 100. For $\chi^2$ and ANOVA-F the values were 65 and 100, respectively.

We have also explored a set of hyperparameters with different values using a grid search for each classifier. As an exhaustive search, each iteration evaluates a new classifier with a unique subset of parameter values.

Table 3 shows these hyperparameters per each classifier.
list of features were executed using the computational facilities of UTFSM cluster (hpc.utfsm.cl).

Software

Experiments were performed using Python 3.7.5 and libraries scikit-learn 0.21.3. Furthermore, we have used Jupyter notebooks. The version of the notebook server is 6.0.1. The server is running on Python 3.7.5. The Python libraries includes functions of scikit-learn 0.23.2: feature_selection (SelectFromModel, SelectKBest, f_classif and chi2), svm, ensemble (ExtraTreesClassifier, RandomForestClassifier), linear_model (LogisticRegression), neighbours (KNeighborsClassifier), model_selection (train_test_split, GridSearchCV) and metrics (classification_report, confusion_matrix, accuracy_score) [59]. The SelectKBest is a filter that chooses features from data contributing most to the target variable, i.e. the best predictors for the target variable. For this, the filter uses two parameters to indicate the stats to be employed, f_classif for ANOVA, chi2 for χ2 and number of top features to select (referred by k). The GridSearchCV is a class to execute exhaustive search over specified parameter values for an estimator. The parameters of the estimator used are optimized by cross-validated grid-search over a parameter grid. This class permits to carry on the hyper-parameters study. Rank aggregation method instances were provided by the library rankaggregation: 0.1.2, with functions: borda, instant_runoff, dowdall and average_rank.

Experiment execution

We have combined the feature selection methods (Table 2) and classifiers (Table 3) for a total of 16 experiments on each image of our dataset (“The dataset”). The execution follows a Cartesian product of (feature selection method × classifiers) detailed as follow: {ANOVA, χ2, ExtraTrees. RF} × {RF, LR, SVM, KNN}. Then, the total number of experiments was 144 considering the 9 images. Additionally, the grid search procedure was performed in each one of the previous 144 experiments to obtain the optimal parameters for each classifier. Considering the class imbalance issue, the training set is generated partitioning the pixels data into the two classes (HER2 and non-HER2), and a sample is randomly selected within each class. Hence, this sampling approach allows selecting random pixels, independently in each class to adequately sample the HER2 pixels that are widely under-represented. In this experiment, we select 13,005 pixels from each one of the nine images of 106 pixels each, and 30% of the 13,005 pixels, i.e., 3901 pixels, are randomly selected from the HER2 class, and the rest 9104 pixels, are also randomly selected from the non-HER2 class. We selected the training set size of 13,005 pixels considering that one cell is approximately contained in a patch of 51x51 pixels. Therefore, we consider 5 cells—as a minimum number of cells—which would be contained in 51x51x5 pixels, corresponding to 13,005 pixels.

Best classification results

Table 4 shows the performance metrics obtained for each image, considering the best combination, based on the F1 score, between the features selection method and the classification algorithm. This combination generates a set of ranked features for each image, that will be used as input for the aggregation method. The last column of Table 4 shows the number of selected features for each image. Table 4 shows also that filter selection methods have a domination over embedded methods. In fact, filter methods evaluate the discriminative power of features based only on intrinsic properties of the data and they are independent of classifiers. This singular characteristic of filters avoids all influence of classifier’s bias in the feature selection process [60]. Applying the filter selection method and reducing the number of features imply that in the next step of the learning process, classifiers will also train with a smaller number of features avoiding redundant or noisy features. Lazar [60] establishes “that wrapper and embedded methods guarantee accurate results only if the classifier used for feature selection is also used to predict new samples while filters can be used with a broader range of classifiers”. The Table 5 resumes the parameters employed by classifiers for these results. In order to compare our results, we employed as a baseline the results without feature selection methods. Table 6 shows the results with the set of 210 features. We can observe that the results are very similar to Table 4, which evidences that less features do not change significantly the results of the full set of features. That is good because with less features is also possible to get good performance results, i.e. parsimonious models. A parsimonious model implies learning models less complex, thus easier to interpret, to understand and more robust when predicting new samples [61]. The learning process can also be executed more efficiently. For 6 of the 9 images, the best combination was RF classifier with χ2 and 65 selected features. The best results were obtained with the univariate selection method, for 8 of 9 images. Finally, the multivariate selection method did not provide the best metrics.

Rank aggregation experiments

Aggregation experiments

From the previous experiment, we obtained 9 different ranked feature sets, with different set size, as shown in Table 4. Hence, we aggregate individual rankings using four ranking aggregation methods: Borda count (BC), Instant Runoff (RO), Dowdall (DO), and Average (AV). Each method generates an aggregated list containing 146 ranked
features, and from each list, we select the top 65 features for further pixel classification. Table 7 shows the top 20 features selected by each aggregated method.

Color features correspond to the different color channels described in “Pixel-level feature computation”, and considering the top 65 ranked features, the 4 methods agreed in selecting 5 features: DAB, K, S, V, and Y. Additionally, each aggregation method selected the following color features: BC method: B, H, M, and R; RO method: B, L, and M; DO method: B, C, G, grey, L, and M; and AV method: b*, C, G, grey, L, and R.

Regarding the texture features, we observed that the BC and RO methods selected the same Haralick texture features, including the sum variance (svar), contrast (con), variance (var), average (avg), sum average (savg), and difference variance (dvar) Haralick texture features applied on different color channels (see Haralick features in Appendix B). However, these 6 Haralick features were not applied exactly in the same channels. For instance, the BC selected the difference variance on the K and L* channels (k.dvar, l.dvar), which were not selected by the Runoff method. The DO and AV methods agreed in the selection of almost all texture features, with 8 features in common including sum variance (svar), sum average (savg), average (avg), variance (var), contrast (con), difference variance (dvar), standard deviation (std), and difference entropy (dent). Additionally, the AV method selected the entropy (enth) feature. The sum variance (svar) was the most repeated texture feature, applied on different color channels. More precisely, the sum variance applied to the Y channel of the CMYK color space is in the first position in three of the four methods. This is quite consistent considering that the Yellow or the Y channel highlights most of the HER2 overexpressed membrane stained with DAB, as shown in Fig. 4.

Table 4 Best classification results considering F1 performance metric. NF: number of features on ranking list

| Image | Feature selection | Classif. method | Rec.  | Spec.  | Acc.  | Prec.  | F1   | NF  |
|-------|-------------------|-----------------|-------|--------|-------|--------|------|-----|
| 1+2   | χ²                | RF              | 0.894 | 0.948  | 0.948 | 0.113  | 0.201| 65  |
| 1+20  | χ²                | RF              | 0.913 | 0.941  | 0.940 | 0.134  | 0.234| 65  |
| 1+25  | χ²                | RF              | 0.905 | 0.936  | 0.936 | 0.096  | 0.174| 65  |
| 2+1   | ANOVA             | LR              | 0.858 | 0.948  | 0.947 | 0.183  | 0.302| 100 |
| 2+8   | χ²                | RF              | 0.869 | 0.941  | 0.940 | 0.191  | 0.313| 65  |
| 2+9   | χ²                | RF              | 0.912 | 0.941  | 0.941 | 0.341  | 0.496| 65  |
| 3+15  | ANOVA             | RF              | 0.837 | 0.899  | 0.888 | 0.632  | 0.720| 65  |
| 3+16  | RF                | RF              | 0.900 | 0.920  | 0.917 | 0.655  | 0.758| 62  |
| 3+19  | χ²                | RF              | 0.858 | 0.915  | 0.913 | 0.228  | 0.361| 65  |

Table 5 Selected parameters for best classifier

| Image | Classifier | Parameters |
|-------|------------|------------|
| 1+2   | RF         | {max_depth:20, max_features: auto, min_samples_leaf: 1, min_samples_split: 2, n_estimators: 550} |
| 1+20  | RF         | {max_depth:80, max_features: auto, min_samples_leaf: 1, min_samples_split: 10, n_estimators: 100} |
| 1+25  | RF         | {max_depth: 50, max_features: auto, min_samples_leaf: 1, min_samples_split: 2, n_estimators: 100} |
| 2+1   | LR         | {C: 0.01, class_weight: None, max_iter: 1,000,000, solver: lbfgs} |
| 2+8   | RF         | {max_depth: 90, max_features: auto, min_samples_leaf: 1, min_samples_split: 5, n_estimators: 1000} |
| 2+9   | RF         | {max_depth: 20, max_features: auto, min_samples_leaf: 1, min_samples_split: 5, n_estimators: 325} |
| 3+15  | RF         | {max_depth: 40, max_features: auto, min_samples_leaf: 1, min_samples_split: 5, n_estimators: 100} |
| 3+16  | RF         | {max_depth: None, max_features: auto, min_samples_leaf: 4, min_samples_split: 2, n_estimators: 325} |
| 3+19  | RF         | {max_depth: 80, max_features: auto, min_samples_leaf: 1, min_samples_split: 2, n_estimators: 550} |

Table 6 Best classification results considering F1 performance metric without feature selection

| Image | Classif. method | Rec.  | Spec.  | Acc.  | Prec.  | F1   |
|-------|-----------------|-------|--------|-------|--------|------|
| 1+2   | RF              | 0.896 | 0.948  | 0.948 | 0.114  | 0.202|
| 1+20  | RF              | 0.913 | 0.941  | 0.940 | 0.136  | 0.237|
| 1+25  | RF              | 0.908 | 0.936  | 0.936 | 0.097  | 0.176|
| 2+1   | LR              | 0.861 | 0.948  | 0.946 | 0.182  | 0.301|
| 2+8   | RF              | 0.871 | 0.941  | 0.940 | 0.192  | 0.315|
| 2+9   | RF              | 0.915 | 0.942  | 0.941 | 0.342  | 0.498|
| 3+15  | RF              | 0.840 | 0.900  | 0.889 | 0.634  | 0.723|
| 3+16  | RF              | 0.903 | 0.920  | 0.918 | 0.656  | 0.760|
| 3+19  | RF              | 0.860 | 0.914  | 0.913 | 0.228  | 0.360|
We assessed the effectiveness of each aggregation method measuring the performance of pixel-level classifiers. The classification was performed by generating a classifier per image and to avoid the effect of imbalanced distribution, we built a training set consisting of all positive-class pixels plus the same number of randomly selected negative-class pixels in each image. In other words, the training set was generated through a majority class undersampling to produce a balanced-class dataset. From this dataset, we use 80% for training and 20% for validation. Table 8 displays four rows corresponding, from top to bottom, to the total number of pixels belonging to the positive class (HER2), the number of pixels of the negative class (non-HER2) that were randomly selected, the total amount of negative class pixels and the size of the training set, which is computed as the sum of the first and second row values, per each image.

Different classification techniques were used to build the pixel-level classifiers on each image. Appendix D tables show in detail the best performance obtained, and in most cases, the Random Forest-based classifier performed better. Table 9 shows the average results obtained on each aggregation method, considering the set of 9 images. We observe that Borda count and Instant Runoff methods perform similarly, as well as Dowdall and Average methods. The difference between these, as low as a few ten-thousandths of a unit, is imperceptible at the used scale. It is also important to note that Borda count and Runoff methods top 65 feature sets, have 61 features in common, while Dowdall and Average methods have 54.

Thus, considering that Borda count and Instant Runoff showed the best results, we selected the aggregated list provided by each of these methods, and assessed them on a general pixel classifier using the hyperparameters defined in Table 3. This process is described in the following subsection.

**Evaluation on full set of pixels**

The last experiment consisted of measuring the classification performance using the aggregated feature sets provided by Borda count and Instant Run-off methods. We assessed the performance, building a general pixel classifier. The training set was generated using the same strategy described in the previous “Aggregation experiments”. Considering the whole set of pixels from all of the images together (10^9 pixels), we built a balanced training set taking all the positive pixels (HER2 pixels), undersampling the negative pixels (non-HER2 pixels), and splitting the dataset considering 80% for training and 20% for validation.

In this stage, we used four classifier techniques for identifying HER2 overexpression on a pixel-level: Random Forests, Logistic Regression, KNN, and SVM. Table 10 shows the pixel-level classification performance using the set of 65 features provided by Borda and Instant Run-Off rank aggregation methods. We observe that Random Forest
gets better performance metrics than the other classification methods. More precisely, when Random Forest uses the set of features provided by the Borda count method, the HER2 overexpression identification achieves 0.905 of \(F_1\), 0.939 of recall, and 0.873 of precision. On the other hand, using the set of features given by the Instant Runoff aggregation method, Random Forests achieves 0.906 of \(F_1\), 0.939 of recall, and 0.875 of precision. These experimental results validate that the proposed method can identify 93.9\% of HER2 overexpressed pixels. A high recall (or true positive rate) means a reduction the false negative predictions, which is quite important in any medical imaging support tool.

### Discussion

It is well known that the feature selection process aims to identify and remove unneeded, irrelevant, and redundant attributes from data that does not impact the accuracy of a predictive model in any way. The reduction of the set of features allows producing a learning model that is easier to understand, to explain, and to implement. Furthermore, as data dimensionality decreases, the training process becomes more efficient. Our experiments results show that a trained model using a reduced feature set of size 65, instead of 210, still provides high performance metrics. These results are promising and indicate this method can be applied on bigger sets of features, guaranteeing the selection of all significant features.

As stated, feature selection performance improves when using ensemble methods instead of one method. The high complexity of the HER2 pixel segmentation problem makes it suitable to explore a solution based on ensemble methods as a base selectors combination. We propose an ensemble based on two algorithm types of feature selection: filters and embedded. Our experiments demonstrate that the performance of a model built upon 65 features is still high.

Random Forests is the classification technique that produces the best performance values. Its success is due to the generation of a great amount of relatively uncorrelated models (trees) operating together to outperform any of the individual constituent models. Although, it is important to

### Table 7

Top 20 features selected by each aggregation method. The feature name corresponds \textit{channel.haralick-feature}. Channels: b: RGB blue channel, c: CMYK cyan channel, blab: CIEL*a*b* b channel, g: RGB green channel, grey: Gray-scale gray channel, h: HSV hue channel, k: CMYK key (black) channel, l: L* channel, m: CMYK magenta channel, r: RGB red channel, s: HSV saturation channel, v: HSV value channel, y: CMYK yellow channel. Haralick features: avg: Gray Level Co-occurrence Matrix average, con: contrast, dab: diaminobenzidine, dvar: difference variance, savg: sum average, svar: sum variance, var: variance. The Haralick texture features notation are described in Appendix B and abbreviations are displayed in Appendix C

| Method         | Borda         | Runoff        | Dowdall       | Average       |
|----------------|---------------|---------------|---------------|---------------|
| Top 20 features| y.svar, y.svar| y.svar, y.svar| y.svar, y.svar| y.svar, y.svar|
|                | b.svar, b.svar| h.svar, c.svar| b.svar, r.savg| b.svar, r.savg|
|                | m.svar, h.svar| s.svar, l.svar| r.svar, g.svar| l.svar, k.svar|
|                | g.svar, g.svar| y.svar, m.svar| c.svar, k.svar| c.svar, k.svar|
|                | v.svar, v.svar| g.svar, g.svar| s.svar, y.svar| r.svar, c.svar|
|                | l.svar, h.svar| svar, y.svar, v.svar, y.svar, y.svar, y.svar, y.svar |
|                | y.svar, y.svar| y.svar, y.svar, y.svar, y.svar, y.svar, y.svar, y.svar |

### Table 8

Total HER2 pixels distributed by class

|          | 1+2 | 1+20 | 1+25 | 2+1 | 2+8 | 2+9 | 3+15 | 3+16 | 3+19 |
|----------|-----|------|------|-----|-----|-----|------|------|------|
| Total HER2 pixels | 11,181 | 13,753 | 11,285 | 17,120 | 19,504 | 35,517 | 173,342 | 146,647 | 32,017 |
| Random non-HER2 pixels | 11,181 | 13,753 | 11,285 | 17,120 | 19,504 | 35,517 | 173,342 | 146,647 | 32,017 |
| Total non-HER2 pixels | 988,819 | 986,247 | 988,715 | 982,880 | 980,496 | 964,483 | 826,658 | 853,353 | 967,983 |
| Total training pixels | 22,362 | 27,506 | 22,570 | 34,240 | 39,008 | 71,034 | 346,684 | 293,294 | 64,034 |
Table 10  Classification performance using top 65 Borda count (BC) and Runoff (RO) aggregated features on the full set of pixels

| Classifier | Recall  | Specificity | Accuracy | Precision | F1       |
|------------|---------|-------------|----------|-----------|----------|
|            | BC      | RO          | BC       | RO        | BC       | RO       | BC       | RO       | BC       | RO       | BC       | RO       |
| RF         | 0.939   | **0.939**   | 0.863    | **0.866** | 0.901    | **0.903** | 0.873    | **0.875** | 0.905    | **0.906** |
| LR         | 0.921   | 0.924       | 0.842    | 0.843     | 0.882    | 0.883     | 0.854    | 0.854     | 0.886    | 0.888    |
| KNN        | 0.921   | 0.920       | 0.845    | 0.845     | 0.883    | 0.883     | 0.856    | 0.856     | 0.887    | 0.887    |
| SVM        | 0.930   | 0.931       | 0.842    | 0.844     | 0.886    | 0.887     | 0.885    | 0.856     | 0.891    | 0.892    |

Fig. 5  Generated model with Instant Runoff aggregation method and the Random Forest classifier

note that the difference between the classifier’s performance metrics values is not significant.

As the ensemble method generates different subsets of features for each image, it was necessary to employ a rank aggregation method to generate a unique and generalized set of features. Therefore, we compared four rank aggregation methods: the Borda count (BC), Instant Runoff (RO), Dowdall (DO), and Average (AV), each one producing a different set of ranked features. We selected the top 65 features provided by each aggregation method, and we observed a reduction in the initial 13 Haralick texture features. The four methods agreed in selecting six Haralick features: the sum variance (svar), contrast (con), variance (var), average (avg), sum average (savg), and difference variance (dvar).
DO and AV methods also included the standard deviation (std), and difference entropy (dent) features, and AV additionally selected the entropy feature (enth). These selected Haralick features were applied on different color channels, and the svar applied on the Y (or Yellow) color channel was the first ranked feature on BO, RO, and DO methods. The number of color features was also reduced, and DAB, K, S, V, and Y were the features that the four aggregation methods had in common, and channels of the RGB color space did not appear as common features.

To verify the impact of each ranked set, we trained learning models with four classification techniques that were fed with the top 65 ranked feature sets. We observed that using Random Forests, Borda count, and Instant Runoff performed the best, even though Instant Runoff performed slightly better. The similar results can be explained, considering that both ranked feature sets differ only by 4 features out of 65. Thus, as shown in Table 10, the RO provided the best performance, with 0.939 recall, 0.866 specificity, 0.875 precision, and 0.906 F1-score. From the experimental results, we also observed that recall always performed better than the precision metric. This means the classification model can identify a high percentage of HER2 pixels, but it can be imprecise, depending on the number of false positives (predicting HER2 pixels that are non-HER2).

Additionally, we qualitative analyze the performance, visualizing the pixel classification on three images, as shown in Fig. 5.

Conclusions

In this work, we proposed a new ensemble method for selecting representative features for segmenting HER2 overexpression in breast cancer images, using a binary pixel classification approach. The proposed ensemble method considered four different base selectors (χ², ANOVA, RandomForest, and ExtraTree), and four aggregation methods (Borda count, Instant Runoff voting, Dowdall, and Average rank). Also, we used the following binary classification techniques: Random Forest, Logistic Regression, KNN, and SVM. The proposed ensemble method was able to reduce the initial set of 210 color and texture features to 65 features, still obtaining high classification performance.

In the first stage, the base selectors were used for identifying the feature set size that would provide the best classification performance for each image, separately. Then, we obtained the most relevant feature sets of size 65 using the rank aggregation techniques. We analyzed the color and texture features and observed that the four aggregation methods agreed on five color features and six Haralick textures features (applied on different color channels).

To assess the generalization capacity of the proposed ensemble method on a pixel-level, we trained a learning-model using the aggregated set of features, provided by each aggregation method, and a full set of pixels. This set was generated by joining the pixels from all of the 9 images. The experimental results indicate that the general classifier was able to identify the HER2 pixels with high classification performance, and showed that the Instant Runoff (RO) method was the method that provided the best performance, but with a slight difference concerning BC method. Therefore, using 65 pixel-level color and texture features, we were able to identify HER2 pixels with 0.939 recall and 0.875 precision.

The proposed architecture was used to produce a novel algorithm trained in manually labeled pixels for HER2 image feature selection, able to effectively reduce the dataset dimensionality. Results showed that is clearly possible to add new pixel-level color and texture features, because the method will select the more relevant features in terms of the final classification performance. Additionally, a new dataset of manually labeled HER2 images by a specialist was set. No other same-conditioned datasets or similar ensemble method proposals in HER2 overexpression studies were found in the literature, which does not allow us to test the model with different datasets or compare it with other author results.

All of the nine tissue images were taken by the same scanner machine and the use of another may produce images with varied-value properties like contrast, flash, among others, which are essential for feature computing.

As future work, we will incorporate new color and texture features on a pixel-level, and we envisage testing the ensemble method using other datasets in the same context of histopathological images. Since our dataset is composed of images of three tissue categories (1+, 2+, and 3+), one possible way of following our study is to compute a quantification score over HER2 pixels to identify the category the tissue sample belongs to. In addition, we aim to enlarge the dataset by increasing the number of manually labeled images to further feed to deep-learning-based models such as convolutional neural networks.

Acknowledgements The authors would like to thank the assistance in the tissue digitization performed at the SCIAN Laboratory (University of Chile, Santiago, Chile).

Funding This work has been partially funded by CCTVal ANID PIA/APOYO AFB180002, and FONDECYT Postdoc Project 3190740.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.
Appendix A. Abbreviations

See Table 11.

Table 11

| Abbreviation | Definition |
|--------------|------------|
| AV           | Average    |
| BC           | Borda count|
| CIEL*a*b*    | L\^{}* for perceptual lightness, and a\^{}* and b\^{}* for the four unique colors of human vision: red, green, blue, and yellow. Defined by the International Commission on Illumination |
| CMYK         | Cyan, Magenta, Yellow, Black |
| CNN          | Convolutional Neural Networks |
| DAB          | Diaminobenzidine |
| DO           | Dowdall |
| FISH         | Fluorescence in situ Hybridization |
| GLCM         | Gray Level Co-occurrence Matrices |
| GLRLM        | Gray-level run length matrix |
| HER2         | Human Epidermal growth factor Receptor 2 |
| HSV          | Hue, Saturation, Value |
| L\^{}*a*b*   | L\^{}* indicates lightness, a\^{}* is the red/green coordinate, and b\^{}* is the yellow/blue coordinate |
| IHC          | Immunohistochemistry |
| KNN          | K-Nearest Neighbours |
| LR           | Logistic regression |
| PET          | Positron emission tomographies |
| RF           | Random Forest |
| RGB          | Red, Green, Blue |
| RO           | Instant Runoff |
| SVM          | Support Vector Machine |
| ULBP         | Uniform Local Binary Patterns |
| WSI          | Whole Slide Imaging |

Table 11 continued

| Abbreviation | Definition |
|--------------|------------|
| YCbCr        | Y is the luma component of the color, Cb is the blue component relative to the green component. Cr is the red component relative to the green component |

Appendix B. Haralick features

Let \( P_{d}^{\theta}(i, j) \) be the \((i, j)\)-th entry of a GLCM with distance \(d\) and angle \(\theta\). For simplicity of presentation, we will denote \( P_{\theta}^{d}(i, j) = p(i, j) \) in this section. The mean and standard deviations for the rows and columns of the GLCM are shown as follows:

\[
\mu_{x} = \frac{1}{N_{x} - 1} \sum_{i=0}^{N_{x} - 1} \sum_{j=0}^{N_{y} - 1} i \ p(i, j),
\]

\[
\mu_{y} = \frac{1}{N_{y} - 1} \sum_{i=0}^{N_{x} - 1} \sum_{j=0}^{N_{y} - 1} j \ p(i, j),
\]

\[
\sigma_{x} = \sqrt{\frac{1}{N_{x} - 1} \sum_{i=0}^{N_{x} - 1} \sum_{j=0}^{N_{y} - 1} (i - \mu_{x})^2 p(i, j)},
\]

\[
\sigma_{y} = \sqrt{\frac{1}{N_{y} - 1} \sum_{i=0}^{N_{x} - 1} \sum_{j=0}^{N_{y} - 1} (j - \mu_{y})^2 p(i, j)},
\]

and they are used for the computation of some Haralick features.

1. Angular second moment (a2m):

\[
f_{1} = \sum_{i=0}^{N_{x} - 1} \sum_{j=0}^{N_{y} - 1} p(i, j)^2.
\]

2. Contrast (con):

\[
f_{2} = \sum_{n=0}^{N_{x} - 1} n^2 \left\{ \sum_{i=1}^{N_{x}} \sum_{j=1}^{N_{y}} p(i, j) \right\}, \text{ if } |i - j| = n.
\]

3. Correlation (cor):

\[
f_{3} = \frac{\sum_{i=0}^{N_{x} - 1} \sum_{j=0}^{N_{y} - 1} (ij)p(i, j) - \mu_{x}\mu_{y}}{\sigma_{x}\sigma_{y}}.
\]
4. Sum of squares or variance (var):

\[ f_4 = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i - \mu)^2 p(i, j). \]

5. Inverse different moment (idm):

\[ f_5 = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{1}{1 + (i - j)^2} p(i, j). \]

6. Sum average (savg):

\[ f_6 = \sum_{i=2}^{2N_g} p_{x+y}(i), \]

where:

\[ p_{x+y}(k) = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} p(i, j) \quad k = 2, 3, \ldots, 2N_g. \]

7. Sum variance (svar):

\[ f_7 = \sum_{i=2}^{2N_g} (i - f_6)^2 p_{x+y}(i). \]

8. Sum entropy (sent):

\[ f_8 = -\sum_{i=2}^{2N_g} p_{x+y} \log(p_{x+y}(i)). \]

9. Entropy: (enth):

\[ f_9 = -\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} p(i, j) \log(p(i, j)). \]

10. Difference variance (dvar):

\[ f_{10} = \sum_{i=0}^{N_g-1} (i - f'_{10})^2 p_{x-y}(i), \]

where

\[ f'_{10} = \sum_{i=0}^{N_g-1} i p_{x-y}(i), \]

11. Difference entropy (dent):

\[ f_{11} = -\sum_{i=1}^{N_g} p_{x-y}(i) \log(p_{x-y}(i)). \]

**Appendix C. Haralick texture feature notation**

See Table 12.

**Table 12** Haralick texture feature notation

| notation | description               |
|----------|---------------------------|
| a2m      | Angular second moment     |
| avg      | GLCM average              |
| con      | Contrast                  |
| cor      | Correlation               |
| dent     | Difference entropy        |
| dvar     | Difference variance       |
| enth     | Entropy                   |
| idm      | Inverse different moment  |
| savg     | Sum average               |
| sent     | Sum entropy               |
| std      | GLCM standard deviation   |
| svar     | Sum variance              |
| var      | Variance                  |

**Appendix D. Aggregation methods**

See Tables 13, 14, 15, 16.
### Table 13: Classification performance using top 65 Borda features

| Image | Classifier | Recall | Specificity | Accuracy | Precision | F1  |
|-------|------------|--------|-------------|----------|-----------|-----|
| 1+2   | RF         | 0.962  | 0.919       | 0.940    | 0.920     | 0.941|
| 1+20  | RF         | 0.966  | 0.927       | 0.946    | 0.926     | 0.946|
| 1+25  | RF         | 0.973  | 0.915       | 0.944    | 0.919     | 0.945|
| 2+1   | LR         | 0.932  | 0.907       | 0.919    | 0.909     | 0.920|
| 2+8   | RF         | 0.947  | 0.902       | 0.924    | 0.906     | 0.926|
| 2+9   | RF         | 0.965  | 0.920       | 0.943    | 0.922     | 0.943|
| 3+15  | RF         | 0.949  | 0.858       | 0.903    | 0.870     | 0.908|
| 3+16  | RF         | 0.961  | 0.893       | 0.927    | 0.899     | 0.929|
| 3+19  | RF         | 0.959  | 0.873       | 0.915    | 0.879     | 0.917|
| Average|            | 0.957  | 0.902       | 0.929    | 0.906     | 0.931|

### Table 14: Classification performance using top 65 Runoff features

| Image | Classifier | Recall | Specificity | Accuracy | Precision | F1  |
|-------|------------|--------|-------------|----------|-----------|-----|
| 1+2   | RF         | 0.960  | 0.919       | 0.939    | 0.921     | 0.940|
| 1+20  | RF         | 0.965  | 0.927       | 0.945    | 0.925     | 0.945|
| 1+25  | RF         | 0.975  | 0.918       | 0.946    | 0.922     | 0.948|
| 2+1   | LR         | 0.931  | 0.907       | 0.919    | 0.908     | 0.920|
| 2+8   | RF         | 0.947  | 0.902       | 0.924    | 0.905     | 0.926|
| 2+9   | RF         | 0.966  | 0.920       | 0.943    | 0.922     | 0.944|
| 3+15  | RF         | 0.949  | 0.857       | 0.903    | 0.870     | 0.908|
| 3+16  | RF         | 0.960  | 0.893       | 0.926    | 0.899     | 0.928|
| 3+19  | RF         | 0.958  | 0.872       | 0.914    | 0.878     | 0.916|
| Average|            | 0.957  | 0.902       | 0.929    | 0.906     | 0.931|

### Table 15: Classification performance using top 65 Dowdall features

| Image | Classifier | Recall | Specificity | Accuracy | Precision | F1  |
|-------|------------|--------|-------------|----------|-----------|-----|
| 1+2   | RF         | 0.958  | 0.917       | 0.937    | 0.918     | 0.938|
| 1+20  | RF         | 0.966  | 0.926       | 0.945    | 0.925     | 0.945|
| 1+25  | RF         | 0.971  | 0.916       | 0.943    | 0.920     | 0.944|
| 2+1   | LR         | 0.930  | 0.906       | 0.918    | 0.908     | 0.919|
| 2+8   | RF         | 0.945  | 0.901       | 0.923    | 0.905     | 0.924|
| 2+9   | RF         | 0.964  | 0.918       | 0.941    | 0.921     | 0.942|
| 3+15  | RF         | 0.947  | 0.856       | 0.901    | 0.868     | 0.906|
| 3+16  | RF         | 0.959  | 0.891       | 0.925    | 0.897     | 0.927|
| 3+19  | RF         | 0.956  | 0.870       | 0.912    | 0.876     | 0.914|
| Average|            | 0.955  | 0.900       | 0.927    | 0.904     | 0.929|
Table 16 Classification performance using top 65 Average features

| Image | Classifier | Recall  | Specificity | Accuracy | Precision | F1      |
|-------|------------|---------|-------------|----------|-----------|---------|
| 1+2   | RF         | 0.959   | 0.917       | 0.938    | 0.919     | 0.938   |
| 1+20  | RF         | 0.961   | 0.927       | 0.943    | 0.925     | 0.943   |
| 1+25  | RF         | 0.972   | 0.914       | 0.943    | 0.918     | 0.944   |
| 2+1   | LR         | 0.932   | 0.905       | 0.918    | 0.907     | 0.919   |
| 2+8   | RF         | 0.944   | 0.900       | 0.922    | 0.904     | 0.924   |
| 2+9   | RF         | 0.965   | 0.919       | 0.942    | 0.921     | 0.943   |
| 3+15  | RF         | 0.946   | 0.856       | 0.901    | 0.868     | 0.906   |
| 3+16  | RF         | 0.959   | 0.891       | 0.925    | 0.897     | 0.927   |
| 3+19  | RF         | 0.955   | 0.871       | 0.912    | 0.876     | 0.914   |
| Average|            | 0.955   | 0.900       | 0.927    | 0.904     | 0.929   |

References

1. Alvarez R, Cortés J, Mattos-Arruda L, M. F. et al. (2013) Handbook of HER2-targeted agents in breast cancer. Springer Healthcare Limited, http://books.google.co.id/books?id=Z/Hb0AEACAAJ
2. Fiore PD, Pierce J, Kraus M, Segatto O, King C, Aaronson S (1987) erbB-2 is a potent oncogene when overexpressed in NIH/3T3 cells. Science 237(4811):178–182
3. Wolff A, Elizabeth M, Hicks D, Dowsett M, McShane L, Allison K, D. A. et al. (2013) Recommendations for human epidermal growth factor receptor 2 testing in breast cancer: American society of clinical oncology/college of american pathologists clinical practice guideline update. Journal of Clinical Oncology 31(31):3997–4013. https://doi.org/10.1200/JCO.2013.50.9984
4. Perez E, Suman V, Davidson N, Martino S, P. K. et al. (2006) HER2 testing by local, central, and reference laboratories in specimens from the north central cancer treatment group N9831 intergroup adjuvant trial. Journal of Clinical Oncology 24(19):3032–3038
5. Masmoudi H, Hewitt S, Petrick N, Myers K, Gavrielides M (2009) Automated quantitative assessment of HER2/neu immunohistochemical expression in breast cancer. Medical Imaging, IEEE Transactions on 28(6):916–925. https://doi.org/10.1109/TMI.2009.212901
6. Ficarra E, Cataldo S, Acquaviva A, Macii E (2011) Automated segmentation of cells with ihc membrane staining. IEEE Transactions Biomedical Engineering 58(5):1421–1429
7. Ruijrok A, Johnston D (2001) Quantification of histochemical staining by color deconvolution. Analytical and quantitative cytology and histology 23(4):291–299. http://view.ncbi.nlm.nih.gov/pubmed/11531144
8. Tuominen V, Tolonen T, Isola J (2012) Immunomembrane: a publicly available web application for digital image analysis of HER2 immunohistochemistry. Histopathology 60(5):758–767. https://doi.org/10.1111/j.1365-2559.2011.04142.x
9. Wdowiak M, Markiewicz T, Osowski S, Swiderska Z, Patera J, Kozlowski W (2015) Hourglass shapes in rank grey-level hit-or-miss transform for membrane segmentation in HER2/neu images. In: ISMM, Vol. 9082 of Lecture Notes in Computer Science, Springer, pp. 3–14
10. Wdowiak M, Markiewicz T, Osowski S, Patera J, Kozlowski W (2016) Novel segmentation algorithm for identification of cell membrane staining in HER2 images. Pattern Recogn Lett 84:225–231. https://doi.org/10.1016/j.patrec.2016.10.003
11. Altaf F, Islam S, Akhtar N, Janjua N (2019) Going deep in medical image analysis: Concepts, methods, challenges, and future directions. IEEE Access 7:99540–99572. https://doi.org/10.1109/access.2019.2929365
12. Khameneh F, Razavi S, Kamasak M (2019) Automated segmentation of cell membranes to evaluate HER2 status in whole slide images using a modified deep learning network. Comput Biol Med 110:164–174. https://doi.org/10.1016/j.compbiomed.2019.05.020
13. Zakrzewski F, de Back W, Weigert M, Wenke T, Zeugner S, Mantey R., C. S. et al. Automated detection of the HER2 gene amplification status in fluorescence in situ hybridization images for the diagnostics of cancer tissues. Scientific Reports 9(1). https://doi.org/10.1038/s41598-019-44643-z
14. Kieffer B, Babaie M, Kalra S, Tizhoosh HR (2017) Convolutional neural networks for histopathology image classification: Training vs. using pre-trained networks. In: 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), IEEE, pp. 1–6
15. Anari V, Bakhshi M. Automatic Detection of Proliferative Cells in Immunohistochemically Images of Meningioma Using Fuzzy C-Means Clustering and HSV Color Space. International Journal of Electronics and Communication Engineering 13(11). https://doi.org/10.5281/ZENODO.3566417
16. Bamford P, Chukka S, et al. (2014) Lab color space silver and red in situ hybridization-based techniques for detecting genes in tissue samples, http://www.freepatentonline.com/y2020/0117883.html
17. Syed A, Woodall R, Whisenant J, Yankeelov T, Sorace A (2019) Characterizing trastuzumab-induced alterations in intratumoral heterogeneity with quantitative imaging and immunohistochemistry in HER2+ breast cancer. Neoplasia 21(1):17–29. https://doi.org/10.1016/j.neo.2018.10.008
18. Les T, Markiewicz T, Jesiotr M, Kozlowski W (2016) Dots detection in HER2 FISH images based on alternative color spaces. Procedia Computer Science 90:132–137. https://doi.org/10.1016/j.procs.2016.07.025
19. Placzek B, Lewandowski M, Buldak R, Michalski M (2018) Automated immunohistochemical stains analysis for computer-aided diagnosis of parathyroid disease. In: Computer Information Systems and Industrial Management, Springer International Publishing. pp. 68–79. https://doi.org/10.1007/978-3-319-99954-8_7
20. Mungle T, Tewary S, Arun I, Basak B, Agarwal S, Ahmed R, Chatjee S et al (2017) Automated characterization and counting of Ki-67 protein for breast cancer prognosis: A quantitative immunohistochemistry approach. Comput Methods Programs Biomed 139:149–161. https://doi.org/10.1016/j.cmpb.2016.11.002
21. Nasrin S, Alam Md Z, Taha TM, Asari VK (2020) PColorNet: investigating the impact of different color spaces for pathological image classification, in: Medical Imaging 2020: Digital Pathology, Vol. 11320, SPIE, pp. 316–326. https://doi.org/10.1117/12.2550046
58. Kabakçı KA, Çakır A, Türkmen İlknur, Töreyin BU, Çapar A (2021) Automated scoring of cerbb2/her2 receptors using histogram based analysis of immunohistochemistry breast cancer tissue images. Biomedical Signal Processing and Control 69:102924. https://doi.org/10.1016/j.bspc.2021.102924. https://www.sciencedirect.com/science/article/pii/S1746809421005218
59. Scikit-learn: machine learning in Python — scikit-learn 0.23.2 documentation, https://scikit-learn.org/stable (Aug 2020)
60. Lazar C, Taminau J, Meganck S, Steenhoff D, Coletta A, Molter C, de Schaetzen V, Duque R, Bersini H, Nowe A (2012) A Survey on Filter Techniques for Feature Selection in Gene Expression Microarray Analysis. IEEE/ACM Trans Comput Biol Bioinf 9(4):1106–1119. https://doi.org/10.1109/TCBB.2012.33

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

61. Laimighofer J, Melcher M, Laaha G (2022) Parsimonious statistical learning models for low-flow estimation. Hydrol Earth Syst Sci 26(1):129–148. https://doi.org/10.5194/hess-26-129-2022