Analysis of Corneal Data in R with the rPACI Package

by Darío Ramos-López and Ana D. Maldonado

Abstract In ophthalmology, the early detection of keratoconus is still a crucial problem. Placido disk corneal topographers are an essential tool in clinical practice, and many indices for diagnosing corneal irregularities exist.

The main goal of this work is to present the R package **rPACI**, providing several functions to handle and analyze corneal data. This package implements primary indices of corneal irregularity (based on geometrical properties) and compound indices built from the primary ones, either using a generalized linear model, or as a Bayesian classifier using a hybrid Bayesian network and performing approximate inference.

rPACI aims to make the analysis of corneal data accessible for practitioners and researchers in the field. Moreover, a **shiny** app was developed so that **rPACI** can be used in any web browser, in a truly user-friendly graphical interface, without installing R or writing any R code. It is openly deployed at https://admaldonado.shinyapps.io/rPACI/.

Introduction

One of the major problems in the field of ophthalmology is the early detection of keratoconus (Martínez-Abad and Piñero, 2017; Accardo and Pensiero, 2002; Rabinowitz, 1998). Keratoconus is a serious disease that deforms and weakens the cornea (the outer part of the eye). There are several grades of this disease (Alió and Shabayek, 2006), with the most advanced stages being easy to detect. However, early or subclinical keratoconus cases are much more difficult to identify (de-Sanctis et al., 2008; Maguire and Bourne, 1989). Diagnosing keratoconus is crucial before undergoing any kind of refractive eye surgery (Llorente et al., 2004), in which vision is corrected by modifying the corneal curvature, normally with lasers. Failure to detect early keratoconus in these circumstances can result in an ectasia (Randleman et al., 2003), which in the most severe cases can lead to blindness or corneal transplantation.

For the measurement and analysis of the cornea, the main clinical tool is the corneal topography (Fan et al., 2018; Piñero, 2015; Read et al., 2009; Rowsey et al., 1981). Depending on the employed technology, they can be basically classified into two types: Scheimpflug camera devices (Read et al., 2009) or Placido disk devices (Alkhaldi et al., 2009; Samapunphong and Azar, 1998; Rand et al., 1997), although some devices combine these two technologies (Fan et al., 2018; Piñero, 2015). As a result of the measurement, topographers internally compute and provide altimetric (elevation), curvature, or diopters data of the anterior or posterior sides of the cornea. Currently, most of the topographers available in clinics world-wide are Placido disk topographers, as the Scheimpflug technology is newer and more expensive. Thus, it is not so well established in small or medium-sized clinics nor in countries with a more limited health care system.

Many indices or metrics using corneal topography have been proposed for the early detection of keratoconus (Alió, 2016; Prakash et al., 2012). Some of the most widely used indices are KPI (Maeda et al., 1994) and KISA (Rabinowitz and Rasheed, 1999). In recent years, work is continuing to find more reliable indices for detecting keratoconus (Alió, 2016; Ramos-López et al., 2011; Castro-Luna et al., 2020; Bühren et al., 2010), with a special focus on the early stages of this clinical condition (Ramos-López et al., 2013; Issarti et al., 2019). However, the early detection of keratoconus is still an open research problem and new approaches have been proposed (Issarti et al., 2020; Yousefi et al., 2018; Ortiz-Toquero et al., 2020).

In ophthalmological practice, however, the reference indices are normally only those included in the topographer by default, computed internally in a black-box scheme from the raw measurements. It can be difficult for a practitioner to calculate other indices that are not directly provided by the device, either for comparison, to complement a diagnosis, or for research purposes. Therefore, this article aims to present and explain the R package **rPACI** (Placido Analysis of Corneal Irregularity, Ramos-López and Maldonado (2021)) and its use cases. This package was developed to facilitate the calculation and interpretation of several indices for detecting corneal irregularities, and especially keratoconus (the indices introduced in Castro-Luna et al. (2020); Ramos-López et al. (2011, 2013)). These indices proved to be effective in detecting keratoconus and early keratoconus (see those references for more details).

The main goal of **rPACI** (Ramos-López and Maldonado, 2021) is that these indices can be easily computed by an ophthalmologist, an optometrist, or any practitioner in general. To this end, several intuitive and easy-to-use functions are provided, including those that can read the data of a Placido

disk topography from a file, analyze the data read and calculate the indices, return the results in a manageable format, and represent them graphically so that they are easily interpreted. To the best of the authors' knowledge, there are neither other similar R packages nor other pieces of non-commercial software sharing this goal.

As some practitioners may have difficulties in using the R syntax, we provide, in addition to the package itself, a user-friendly intuitive web application for **rPACI**, built using **shiny** (Chang et al., 2020). This graphical user interface is available to use from any web-browser and it allows the user to readily use the functionality of **rPACI** without installing R or any of its packages.

Placido corneal topography and supported file formats

The majority of topographers used in clinical practice to measure corneal topography rely on the Placido disk technology (either alone or in combination with others, such as the Scheimpflug camera) (Piñero, 2015; Samapunphong and Azar, 1998; Fan et al., 2018). In such a topographer, an illuminated pattern of concentric rings or mires (which is the actual Placido disk) is projected into the corneal surface, and its reflection on the anterior surface of the cornea is captured by a camera situated at the center of the system. The picture is then digitized along a certain number of radii at equally spaced angles (Samapunphong and Azar, 1998; Espinosa et al., 2013).

Many manufacturers produce corneal topographers that employ the Placido disk technology. Some devices are often referred to as aberrometers or tomographers (as they also include other technologies than the Placido disk), or even keratometers or corneal analyzers. Some of these manufacturers are: CSO (Firenze, Italy), Gaush (Beijing, China), Medmont (Victoria, Australia), Nidek (Aichi, Japan), Optikon (Roma, Italy), Topcon (Tokyo, Japan), Zeiss (Oberkochen, Germany), and Ziemer (Port, Switzerland). This list has been elaborated with publicly available information or reported in scientific papers. The authors do not have any commercial nor financial interest in the manufacturers reported here or in any other related companies. The **rPACI** package includes a vignette entitled "Corneal topographers and data formats" where additional information about topographers and manufacturers is provided.

There are two basic magnitudes in the Placido corneal measurement: the number of rings or mires in the actual Placido disk N_R and the number of angles N_A (or number of points per ring) in the digitization process. These may vary from a device to another. Typical values of N_R in commercial topographers are 20, 22, or 24, even though some topographers have as few as 10 or as many as 33. With respect to N_A, the most common values used by commercial topographers are 256 and 360, although some of them use other values.

The plain coordinates (x and y) of these points are the primary raw information measured by the Placido corneal topographers. Afterward, they usually post-process the data to obtain a third coordinate (z), yielding altimetric (elevation) or curvature data, and provide color maps and other easy-to-interpret metrics and indices to the practitioner. In some topographers, the raw measurements can be exported in a structured text file.

rPACI can read corneal topography files in two different formats, by now. It will possibly be expanded in the future allowing other formats. Both are structured plain text files. The first supported file format is basically the one employed by CSO topographers, but with some more flexibility. The second supported file format has the structure used internally by **rPACI**. These two formats are described in the next two subsections.

CSO file format

This file format is the one in which some CSO topographers (EyeTop2005, Sirius, Antares, or Osiris-T), export the raw data measurements. The corneal topography file should have the following 3-block structure:

- An optional header with meta-data: patient data, optical measurements, date of exploration, etc. Its size depends on the device. For CSO topographers, the header typically has 24 lines. The **rPACI**'s reading function for these files is able to identify the header block and to skip it.

- A list of size $N_R \times N_A$ with the ρ coordinate (distance to the origin in polar coordinates, measured in mm) of the digitized points, at equally spaced angles (θ, the argument in polar coordinates). The θ coordinate is inferred from the position (assuming a uniform distribution), as it is not explicitly given in the data file.

- A list of size $N_R \times N_A$ with the z coordinate, which can be altitude (elevation) or curvature, depending on the topographer and the exportation settings. These data are post-processed by

the device using different algorithms, not directly measured, and the z values are not used in **rPACI**.

In the best scenario (which is not very common in the clinical practice), the N_R rings are fully available, having N_A points each, giving a total of $N_R \times N_A$ data points. However, many points are often missing, especially those corresponding to outermost rings (as they could not be digitized properly). Missing data are generally codified in the file with a specific key number (such as "-1" or "-1000"). Thus, these values do not correspond to real measured points and should be removed or substituted by NAs when reading the data file.

rPACI file format

This file format has been developed to directly manage the datasets handled by **rPACI**, allowing to save and read data easily. A corneal topography file with this format should have the following 2-block structure:

- An optional header of any length (its size is automatically detected; may be missing). If the file was saved from a dataset generated by simulation, the header contains the simulation parameters
- The data block, which consists of three separated columns (x and y coordinates of each point (in mm), and its ring index), with $N \times N_A$ lines (a line per data point), with N being the number of rings or mires and N_A being the number of points in each ring. The last column (ring index) will consist only of positive integer values, whereas the two coordinates can be any real numbers.

Since files with this format have been exported by **rPACI**, their data will consist only of full rings (either if the dataset was read from another format or generated by simulation). Thus, these files should not have any missing data.

Placido indices of corneal irregularity

Several indices of corneal irregularity can be calculated from the raw measurements of a Placido disk corneal topographer (Ramos-López et al., 2011, 2013). Many other similar indices exist (Alió, 2016; Prakash et al., 2012; Maeda et al., 1994; Rabinowitz and Rasheed, 1999; Issarti et al., 2019; Bühren et al., 2010), but they cannot be computed using just the raw data of a Placido topographer, as they require besides altimetry, pachymetry, or curvature data, information about both surfaces of the cornea, or other parameters.

The Placido indices based on the raw ring images have demonstrated their ability in diagnosing clinical diseases such as keratoconus and subclinical keratoconus, showing high accuracy (Ramos-López et al., 2011, 2013; Castro-Luna et al., 2020). For the readers' convenience, a brief summary of those indices is reported below (more details and derivations can be found in those references).

The indices available in the **rPACI** package make use only of the plain coordinates of the points (either ρ and θ in polar coordinates or x and y in Cartesian coordinates), as that is the raw information measured by a Placido corneal topographer. The post-processed data in z, if any, are discarded. Also, incomplete rings (which are normally in the periphery) are not considered for the calculation of the indices, and only the innermost, complete rings are used. Incomplete rings could be used too, but they are usually highly affected by noise and measurement errors, and their use could distort the results. Hence, after reading the data file (in any format), one has $N \times N_A$ 2D points P_j, $j = 1, \ldots, N \times N_A$, where N denotes the number of innermost complete rings ($N \leq N_R$), given in Cartesian coordinates by (x_j, y_j) or in polar coordinates by (ρ_j, θ_j).

If the data file was in CSO format, the values ρ_j are read from the file, and θ_j values are computed as $\theta_j = (2\pi/N_A)j \mod 2\pi$ (assuming these angles or arguments θ_j are equally spaced along each ring). Thus, if R_k is the k-th ring ($1 \leq k \leq N$), then

$$P_j = (\rho_j, \theta_j) \in M_k \quad \Leftrightarrow \quad j \in J_k := \{n_k, n_k + 1, \ldots, n_k + (N_A - 1)\}, \quad n_k = 1 + N_A(k - 1).$$

The Cartesian coordinates (x_j, y_j) of point P_j can be readily computed using the variable change $x_j = \rho_j \cos(\theta_j)$, $y_j = \rho_j \sin(\theta_j)$. Figure 1 shows an example of the digitized points, both in polar and Cartesian coordinates, as given by a CSO commercial corneal topographer.

If the data file was in rPACI format, then the Cartesian coordinates (x_j, y_j) of full rings are directly read, with the same structure explained above, but without the need of applying the variable change.

The Placido irregularity indices computed by **rPACI** can be split into two categories: primary and combined indices. The primary indices measure certain geometrical properties of the data distribution. Based on them, other combined or compound indices are computed.
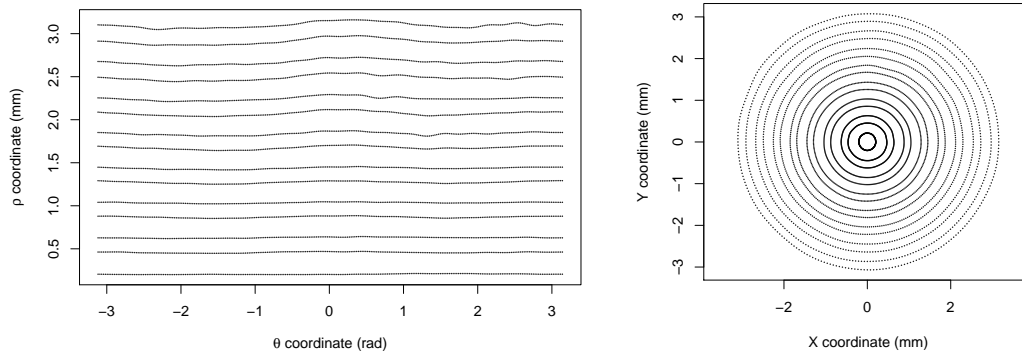
Figure 1: Corneal topography data given by a commercial Placido topographer, in polar coordinates (left) and Cartesian coordinates (right).

Primary indices

With the notation introduced before, the primary information for computing the Placido irregularity indices are the Cartesian coordinates of points belonging to full rings (no missing data):
$P_j = (x_j, y_j) \in R_k, k \in \{1, \ldots, N\}, j \in J_k$.

The first step to calculate the indices is to compute the best-fitting circle (with center C_k and (average) radius $AR(k)$) for each ring. This is done by least-squares fitting of a general circle equation to the Cartesian coordinates of data points in ring k, and then, the geometrical parameters (center and radius) are computed from the coefficients (see Ramos-López et al. (2011) for more details). Similarly, one can find also best-fitting ellipses (with centers $\tilde{C}_k$ and axis ratio c_k) for each ring. Additionally, a best-fitting line can be adjusted to the coordinates of the centers C_k, with slope m.

With these quantities, four indices labeled as PI_n (from "Placido Irregularity indices") and SL (from "SLope" index) can be defined:

$$PI_1 = \frac{1}{N} \max_{1 \leq n,m \leq N} \|C_n - C_m\|,$$

corresponds to the diameter of the set of centers C_k (normalized by the total number of rings N), where $\|\cdot\|$ is the standard Euclidean norm in $\mathbb{R}^2$.

$$PI_2 = \frac{1}{N-1} \sum_{1 \leq n \leq N-1} \|C_{n+1} - C_n\|,$$

corresponds to the total length of the path connecting consecutive centers.

$$PI_3 = \sqrt{\frac{1}{N} \sum_{1 \leq k \leq N} (c_k - \bar{c})^2}, \quad \text{where} \quad \bar{c} = \frac{1}{N} \sum_{1 \leq k \leq N} c_k,$$

measures the variability of the axis ratios or eccentricities of the individual rings.

$$SL = |m|$$

is the absolute value of the slope of the best-fitting line to the coordinates of the centers C_k.

In order to obtain indices with values in the same range (more easily comparable and to prevent scale problems when combining them), a normalization of each primary index was performed (Ramos-López et al., 2011, 2013). After the normalization, the indices will be in general in the range $[0, 150]$, and values outside that interval are truncated. Values near zero correspond to normal corneas, whereas large values (above 70, approximately) correspond to irregular corneas. See additional details on the mathematical definition of these indices and their normalization, including the normalization coefficients, in (Ramos-López et al., 2011, 2013).

Building compound indices

The primary Placido indices described above, PI_1, PI_2, PI_3, SL, and $AR(k)$ show sensitivity to detecting various irregularities in the cornea, such as keratoconus. However, a single index does not reach sufficient accuracy in the task, and compound indices have been proposed and tested (Ramos-López et al., 2013; Castro-Luna et al., 2020) in order to improve the performance and precision of the
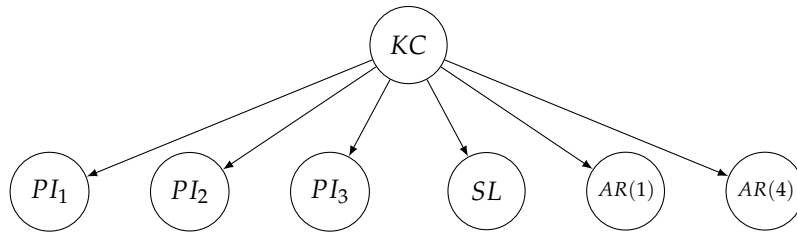
Figure 2: Structure of the Bayesian network used to compute the naïve Bayes index NBI.

individual indices. These compound indices showed a significant improvement in accuracy when predicting keratoconus. A brief description of them is given as follows (see the mentioned references for more details):

The *GLPI* index (from Generalized Linear Placido Index) (Ramos-López et al., 2013) is a generalized linear model computed from the primary indices, in the following way:

$$GLPI = 100\Phi\left(\left(-224.90 + 1.69 PI_1 + 1.28 PI_3 + 1.89 AR(4) + 0.19 SL\right)/20\right)$$

where Φ stands for the cumulative distribution function of the standard normal distribution. This corresponds to a generalized linear regression model with *probit* (Φ^{-1}) link function. Therefore, *GLPI* has values between 0 and 100.

The compound index *NBI* (from Naïve Bayes Index) (Castro-Luna et al., 2020) is a Bayesian classifier, more precisely, it is a Bayesian network with the naïve Bayes structure depicted in Figure 2. This Bayesian network is a conditional linear Gaussian (CLG) network, with its root node (*KC*) being a discrete binary variable and the rest of the nodes being continuous. The parameters of the model were reported in Castro-Luna et al. (2020). This model can be used for predicting whether a specific cornea (whose values of the primary indices are known) is a normal cornea or a keratoconic cornea. Moreover, the probability of being one type or another can be computed as well, either using exact inference or approximate inference with algorithms such as evidence weighting, likelihood weighting, or more generally, importance sampling (Fung and Chang, 1990; Cheng and Druzdzel, 2000; Ramos-López et al., 2018). Even though exact inference is feasible in this case, approximate inference is easier to implement and to generalize to more complex network structures.

In a nutshell, these algorithms consist of simulating a large number of samples from the network, according to the evidence (i.e., variable values that are known a priori), and averaging their likelihoods to estimate the probability of each state of the target variable (in this case, *KC*).

More formally, if Y is the class variable with m states (*KC* in this model, with 2 states), and X is the vector of explanatory variables (PI_1, PI_2, PI_3, SL, $AR(1)$, and $AR(4)$ in this model), then the posterior probability distribution over Y can be calculated with

$$p(Y = y_j \mid X) = \frac{p(X, Y = y_j)}{p(X)} = \frac{p(Y = y_j) \prod_{i=1}^{n} p\left(X_i \mid Y = y_j\right)}{\sum_{j=1}^{m} p(Y = y_j) \prod_{i=1}^{n} p\left(X_i \mid Y = y_j\right)}, \quad j = 1, \ldots, m.$$

The quantities above can be estimated by drawing, according to the evidence, S independent samples $\left(y^{(s)}, x^{(s)}\right)_{s=1}^{S} = \left(y^{(s)}, x_1^{(s)}, \ldots, x_n^{(s)}\right)_{s=1}^{S}$ from the Bayesian network and computing:

$$\hat{p}(X, Y = y_j) = \sum_{s=1}^{S} \mathbf{1}_{y_j}\left(y^{(s)}\right) p\left(Y = y^{(s)}\right) \prod_{i=1}^{n} p\left(X_i = x_i^{(s)} \mid Y = y^{(s)}\right),$$

$$\hat{p}(X) = \sum_{s=1}^{S} p\left(Y = y^{(s)}\right) \prod_{i=1}^{n} p\left(X_i = x_i^{(s)} \mid Y = y^{(s)}\right),$$

where $\mathbf{1}_{y_j}(y^{(s)})$ denotes the indicator function for y_j (i.e., its values are 1 if $y_j = y^{(s)}$, and 0 otherwise). Note that in the first expression we are adding up only the terms corresponding to $Y = y_j$, whereas in the second one, terms corresponding to all samples are summed up. Thus, the probability of the class Y to take the value y_j can be estimated as:

$$\hat{p}(Y = y_j \mid X) = \frac{\sum_{s=1}^{S} \mathbf{1}_{y_j}\left(y^{(s)}\right) p\left(Y = y^{(s)}\right) \prod_{i=1}^{n} p\left(X_i = x_i^{(s)} \mid Y = y^{(s)}\right)}{\sum_{s=1}^{S} p\left(Y = y^{(s)}\right) \prod_{i=1}^{n} p\left(X_i = x_i^{(s)} \mid Y = y^{(s)}\right)}.$$

The posterior probability estimation $\hat{p}(Y = y_j \mid X)$ can be computed efficiently with several approaches

(Ramos-López et al., 2018). The R package **bnlearn** (Scutari, 2010; Scutari and Ness, 2019) includes an implementation of the likelihood weighting algorithm, which is used by **rPACI** for computing the index NBI.

The rPACI package

This section is intended to illustrate the usage of the R package **rPACI** (Ramos-López and Maldonado, 2021), which implements easy-to-use functions to read corneal topographic data and to compute the corneal irregularity indices as defined in Ramos-López et al. (2011, 2013); Castro-Luna et al. (2020) (described also in the previous section). To the best of the authors' knowledge, there are no other R packages related to the analysis of the corneal data, other indices for detecting keratoconus or other corneal diseases, or any other related topics.

The **rPACI** package includes several useful functions to analyze the corneal data of a patient and 11 example data sets, with seven of them being real measurements from a CSO topographer, and the remaining are simulated data sets. Among them, six are in CSO file format, whereas the other five are in rPACI file format. One can install and load the package as follows:

```
install.packages("rPACI", dependencies = TRUE)
library("rPACI")
```

Reading and writing data

The **rPACI** package is able to read corneal topography files in the two formats described before: CSO format and rPACI format. The package contains three functions to read data: readCSO(), readrPACI() and readFile(). The two former ones read data in CSO and rPACI formats, respectively, while the latter one is a wrapper function able to read both formats. In general, readFile() is the reading function recommended by default, as it is able to read any supported file format. These three reading functions produce a "data.frame" in the rPACI format, i.e., a data frame with three columns (*x* and *y* coordinates of each point and its ring index), and a row per data point. On the other hand, the package contains a function to save corneal topography data sets in the format used by **rPACI**: writerPACI().

External files with a corneal topography in the format exported by some Placido disk topographers, especially those from CSO, can be loaded using the function readCSO(), which takes six arguments: filepath, the path of the file to be read; ringsTotal, the total number of rings available in the measurement (including incomplete rings or missing data; by default, 24); pointsPerRings, the number of points per ring that are digitized in the measurement (by default, 256); ringsToUse, the effective number of innermost rings to use (by default, 15); onlyCompleteRings, a logical value indicating whether to use rings with complete data only or not (by default, TRUE); and NAvalues, a numerical value or vector indicating the value(s) encoding NAs (missing data) in the file (by default, c(-1,-1000)).

On the other hand, external files with a corneal topography in the format exported by **rPACI** can be loaded using the readrPACI() function. This function has two arguments: filepath, the path to the file; and sep, the character used as column separator in the file. Finally, the general wrapper function readFile() internally determines the format of the specified file and applies either readCSO() or readrPACI() if possible, or else it throws an error (if none can be applied, which occurs when the file format does not fit any of these two available formats). The readFile() function takes the file path as a mandatory argument (filepath) and, optionally, any of the arguments available for the other two reading functions. In the following example, readFile() is used the read a sample file with the corneal topography of a normal eye (included in the package as 'N01.txt'):

```
dataset_N = readFile(system.file("extdata","N01.txt", package="rPACI"))
```

In order to save a corneal topography data set to disk, the writerPACI() function can be used. This function takes three arguments: dataset, a data.frame containing the corneal topography points in rPACI format; filename, a character string naming a file (including the extension); and sep, the field separator string (by default, ','). The file is saved in structured plain text following the rPACI format. If the data set to be saved was simulated using the simulateData() function (see more details on this function below), the exported file will include the parameters used for the simulation in its header. The following example exports the previously loaded object, dataset_N, in a text file with comma-separated values:

```
writerPACI(dataset_N, filename = "newData.txt", sep = ",")
```

Corneal analyses

The **rPACI** package includes functions to perform three different analyses: 1) analyze a single eye; 2) analyze a single eye based on repeated measures over time; and 3) analyze multiple eyes simultaneously. All analyses compute the Placido irregularity indices of the given data sets and return a table with the numerical results and a plot, which differs depending on the type of analysis performed.

To compute the Placido irregularity indices of a single eye, the `computePlacidoIndices()` can be used, which takes the object returned by `readFile()` (or, alternatively, `readCSO()` or `readrPACI()`) and two other arguments: `truncateIndicesAt150`, a logical value (by default, `TRUE`) indicating whether the primary indices should be truncated at 150 (so they are in the range 0-150) or not; and `useMaxRings`, a positive integer (by default 15) to specify the maximum number of innermost rings to use (as long as there are enough in the data set). Note that this function requires a minimum of five complete rings to work; otherwise, it will throw an error.

```
results_N = computePlacidoIndices(dataset_N)

> results_N
       Diagnose NBI GLPI PI_1 PI_2 PI_3 SL AR_1 AR_2 AR_3 AR_4 AR_5
Normal cornea   0    0   13   20   18  36   43   22   25   21   23
```

The object returned by `computePlacidoIndices()` is of class `"data.frame"` and contains 12 columns and one row (because we are analyzing only one eye). Note that the results have been rounded to integers here for a better display. The first column of the returned `"data.frame"` is the diagnosis, based on the GLPI index, which can be either 'Irregular cornea' (GLPI≥70), 'Suspect cornea' (30 ≤ GLPI <70) or 'Normal cornea' (GLPI <30). The next column is the Naïve Bayes Index (NBI), which ranges between 0 and 100 and can be interpreted as the probability of suffering from keratoconus. The remaining columns correspond to the other indices (see definitions in the previous sections).

The Placido irregularity indices can be plotted using function `plotSingleCornea()`, which takes three arguments: `dataset`, a `"data.frame"` containing the corneal topography data, i.e., the object returned by a reader function, for instance `readFile()`; `PlacidoIndices`, a `"data.frame"` containing the computed Placido indices, i.e., the object returned by `computePlacidoIndices()`; and, optionally, `filename`, a character vector to be displayed on the plot (for instance, the filename of the corneal topography data set).

```
plotSingleCornea(dataset_N, results_N, filename = "N01.txt")
```

The result of this function can be seen in Figure 3. The left-hand side of the figure shows the input data, whereas the right-hand side shows two charts: the GLPI index plot, which visually indicates the value taken by this index on a colored scale of possible values, and the PI indices distribution, which shows the distribution of the PI_1, PI_2, PI_3 and SL indices in a boxplot placed on a scale of possible values. The color of the charts indicate whether the indices fall within the *normal cornea* region (green color), *suspicious cornea* region (orange color), or *irregular cornea* region (red color). In the example shown in Figure 3, we can clearly see that the eye is diagnosed as normal.
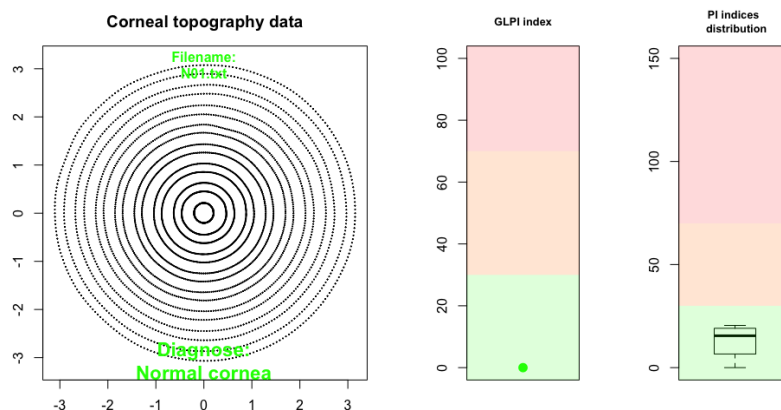
Figure 3: Results of the analysis of a normal cornea, based on a single measurement.

Alternatively to using the `readFile()`, `computePlacidoIndices()` and `plotSingleCornea()` functions, the wrapper function `analyzeFile()` can be used instead, which takes 2 arguments: `path`, which is a character element indicating the location of the corneal topography file, and `drawplot`, which is a logical argument indicating whether the results should be plotted or not.

```
res_N = analyzeFile(system.file("extdata","N01.txt", package="rPACI"), drawplot=TRUE)
```

This function returns a `"data.frame"` containing the same information as the object returned by `computePlacidoIndices()` and, optionally, the plot returned by `plotSingleCornea()` if the argument `drawplot` is `TRUE`. An analogous function to `analyzeFile()`, called `analyzeDataset()`, can be used if the corneal topography data is already loaded in memory.

In order to examine the evolution of a patient's eye over time, the function `analyzeEvolution()` can be used. This function takes two arguments: `data`, which can be either 1) the path to a folder that contains corneal topography files, in any format supported by **rPACI**, or 2) a `"list"` containing properly formatted data (loaded from a file using the function `readFile()` (or `readCSO()` or `readrPACI()`), simulated using `simulateData()`, or by other ways, as long as it meets the data set requirements). If `data` is a path to a folder, the second argument, `fileExtension`, must be specified and all the files (with the given extension) in that folder will be assumed to be corneal topography files of a patient's eye and, therefore, will be loaded.

Moreover, it is assumed that the files are arranged chronologically, i.e., the filenames should follow some date format (for instance, 'YYYY-MM-DD.txt'). On the other hand, if the data are stored in a list, it is assumed that the temporal order corresponds with the index of each data set in the list. The next example analyzes a patient's cornea at three different moments (the data sets are included in the package) :

```
analyzeEvolution(data = system.file("extdata/evolution/", package="rPACI"),
    fileExtension = 'txt')
```

```
         Diagnose NBI GLPI PI_1 PI_2 PI_3 SL AR_1 AR_2 AR_3 AR_4 AR_5 Time
1    Normal cornea 100   29  126  100    0  9    0    0    0    0    0    1
2   Suspect cornea 100   44  126  100    0 50    0    0    0    0    0    2
3 Irregular cornea 100   98  150  138    0 60    0    0    0    0    0    3
```

The `analyzeEvolution()` function returns a `"data.frame"` containing 13 columns and as many rows as files analyzed. Note that the results have been rounded to integers for a better display. The first 12 columns correspond to the diagnosis and the indices, as in the object returned by `computePlacidoIndices()`. The last column corresponds to the time step at which the measures were taken. Moreover, two temporal plots are returned (Figure 4). The left-hand side plot shows the $GLPI$ index, represented by a red line, and the boxplots of the primary indices PI_1, PI_2, PI_3, and SL over time. The right-hand side plot shows the times series of these five indices, $GLPI$, PI_1, PI_2, PI_3, and SL, individually. Finally, both plots present a colored background, corresponding with the final diagnosis of the patient.

In this example shown in Figure 4, the results indicate that the patient's cornea was normal at time 1, suspected at time 2, and irregular at time 3.
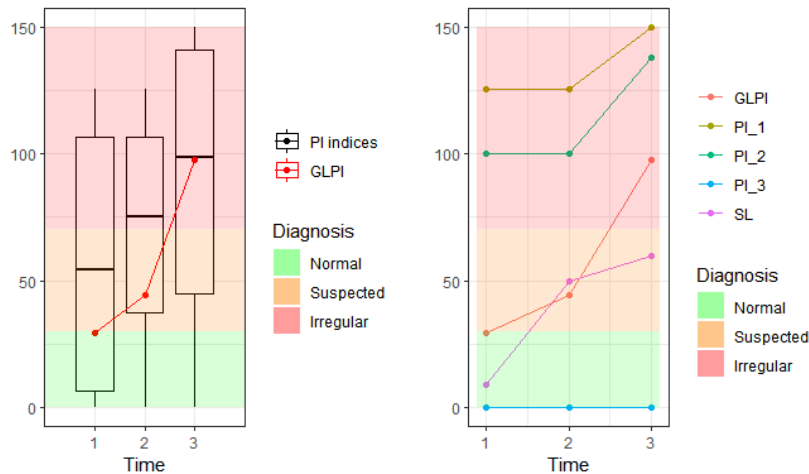

Figure 4: Results of the analysis of a cornea, based on repeated measurements.

So far, `analyzeFile()`, `analyzeDataset()`, and `analyzeEvolution()` are intended to analyze a single cornea, based on single or repeated measurements. In order to analyze multiple corneas simultaneously, the function `analyzeFolder()` can be used. This function takes 4 arguments: `path`, to indicate the location of the folder containing the files; `fileExtension`, which indicates the extension of the files, is set to 'txt' by default; `individualPlots`, which is an optional logical argument indicating

whether the plot for each file should be displayed or not; and summaryPlot, which is an optional logical argument indicating whether a summary plot of all files analyzed should be displayed or not.

```
resultsAll = analyzeFolder(system.file("extdata", package="rPACI"),
individualPlots = FALSE, summaryPlot = T)
```

```
> resultsAll
```

	Diagnose	NBI	GLPI	PI_1	PI_2	PI_3	SL	AR_1	AR_2	AR_3	AR_4	AR_5	Filename
3	Irregular cornea	100	100	150	150	150	142	119	72	83	73	76	K01.txt
5	Irregular cornea	100	100	128	101	119	98	57	56	47	59	51	K03.txt
4	Irregular cornea	100	100	94	85	93	4	0	0	0	0	0	K02.txt
8	Suspect cornea	0	32	43	53	68	26	39	26	31	27	30	S01.txt
7	Normal cornea	0	0	8	21	50	35	57	33	38	34	36	N02.txt
1	Normal cornea	0	0	13	20	18	36	43	22	25	21	23	ds1.txt
6	Normal cornea	0	0	13	20	18	36	43	22	25	21	23	N01.txt
2	Normal cornea	0	0	0	0	0	13	0	0	0	0	0	ds2.txt

This function returns a "data.frame" containing 13 columns and as many rows as files analyzed. Again, the results have been rounded to integers for a better display. The first 12 columns correspond to the diagnosis and the indices, as in the object returned by computePlacidoIndices(). The last column corresponds to the file name so that a specific patient can be easily found. To see the diagnosis for each analyzed file, the first and last columns can be selected:

```
resultsAll[,c(13,1)]
```

```
  Filename        Diagnose
3  K01.txt Irregular cornea
5  K03.txt Irregular cornea
4  K02.txt Irregular cornea
8  S01.txt   Suspect cornea
7  N02.txt    Normal cornea
1  ds1.txt    Normal cornea
6  N01.txt    Normal cornea
2  ds2.txt    Normal cornea
```

Note that the rows are sorted from 'Irregular' to 'Normal cornea'. If the argument summaryPlot() is TRUE, then a barplot is depicted, as shown in Figure 8 (plotted using the Shiny interface, see next section). This barplot shows the absolute frequency of each possible value of diagnosis. In this example, three eyes are diagnosed as irregular, one as suspect and four as normal. This is an easy and straightforward way to check if any patient potentially suffers from keratoconus.

Simulating corneal topographies

rPACI includes the function simulateData() that permits to simulate corneal topography data. This function has a large number of geometrical parameters that allow to generate a wide variety of corneal topographies of different clinical conditions. It produces a data set in the rPACI format, i.e., a data.frame with three columns (x and y coordinates of each point, and its ring index), and a row per data point, according to the function parameters (by default, $15 \times 256 = 3840$ rows or data points, without missing data).

The arguments of simulateData() are: rings, the total number of rings or mires in the sample (by default, 15); pointsPerRing, the number of points to be sampled in each ring (by default, 256); diameter, the diameter of the simulated dataset in mm (typically around 8-12 mm; by default, 12 mm); ringRadiiPerturbation, adds a stochastical perturbation of the mires radii distribution, typically between 0 (no perturbation) and 1 (high perturbation) (by default, 0); maximumMireDisplacement the total mires displacement, drift or off-centering in mm, should be a reasonable number according to the diameter (by default, 0); mireDisplacementAngle, the direction of mires drift given as an angle in degrees, with 0 meaning positive x direction (by default, 0); mireDisplacementPerturbation adds a stochastical perturbation to the mires drift, typically between 0 (no perturbation) and 1 (high perturbation) (by default, 0); ellipticAxesRatio, the ratio or quotient between the major and minor axes of each elliptic ring, a ratio of 1 means a perfect circle (no eccentricity) (by default, 1); ellipticRotation direction or orientation of the elliptic mires given as an angle in degrees, with 0 meaning positive x direction (by default, 0); overallNoise, includes random, white noise of a certain magnitude in the Cartesian coordinates of the sampled points; the noise magnitude is relative to the

diameter and the number of rings; 0 means no noise, and 1 large noise (by default, 0); and finally seed, a seed, included for repeatability when using random perturbations.

For instance, the following piece of code simulates and depicts a corneal topography with $N = 15$ rings, $N_A = 128$ points per ring, with a diameter of 8 mm and introducing a relatively large perturbation in the distribution of radii with ringRadiiPerturbation = 0.7. The results are plotted in Figure 5 (left).

```
dataset = simulateData(rings = 15, pointsPerRing = 128, diameter = 8,
                       ringRadiiPerturbation = 0.7)
plot(dataset$x,dataset$y,pch=20,cex=0.5,asp=1)
```

A highly distorted corneal topography can be generated by using the parameters related to ellipticity and mire displacement. For example, the piece of code below simulates and plots a fairly irregular corneal topography, which includes mires drift of 2mm in the direction $-30°$, with non-uniform displacements, and elliptic rings with axes ratio of 1.2 and orientated at $45°$. The resulting data set is depicted in Figure 5 (right).

```
dataset = simulateData(maximumMireDisplacement = 2, mireDisplacementAngle = -30,
                       mireDisplacementPerturbation = 1.2, ellipticAxesRatio= 1.2,
                       ellipticRotation = 45)
```
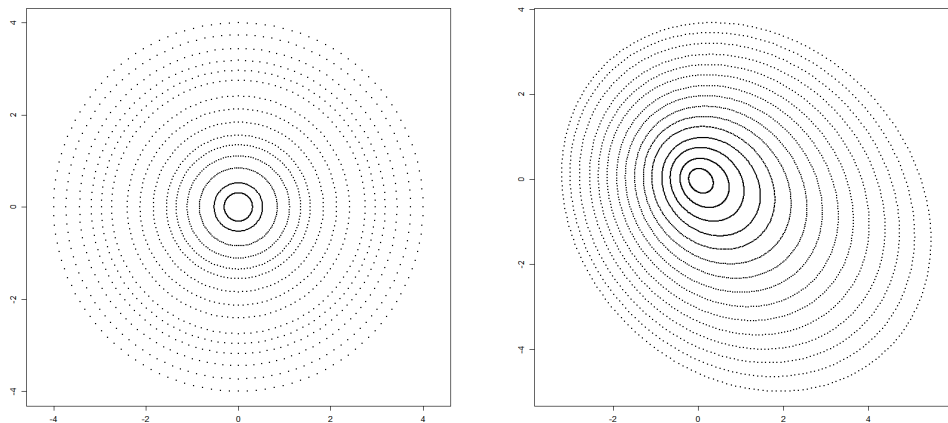

Figure 5: Simulated corneal topographies using simulateData with different parameters: (left) rings with non-uniform radii distribution; (right) adding mires drift and ellipticity.

More examples of using the simulateData() function can be found in the **rPACI** vignette "Simulating corneal datasets".

A Shiny interface to rPACI

The package presented in the previous section requires a basic knowledge of the R language. However, many practitioners might find this a serious obstacle to overcome. For this reason, a shiny interface to **rPACI** has been deployed. **Shiny** is an R package used to develop interactive web applications, so that users do not need to interact with R code, nor to install any software. The rPACI app can be found at https://admaldonado.shinyapps.io/rPACI/.

This app provides a GUI (Graphical User Interface) organized in different tabs, allowing to perform the three types of analyses above described, as well as to simulate corneal topographies. Moreover, the web app contains a 'Home' tab, to welcome users and offer a guided tour through the app; a 'Help' tab, which summarizes the app features; and an 'About' tab, which includes information about the package developers. Links to the **rPACI** vignettes, CRAN and GitHub repositories are also available in the shiny app.

Depending on the number of corneas to analyze at a time (i.e., one or more), the user can select the 'Analyze one patient' or 'Analyze multiple patients' tab. If the first option is chosen, a sub-menu appears, where the user can select between 'Single measurement' and 'Repeated measurements'. The former choice analyzes a cornea based on a single measurement, whereas the latter analyzes a cornea

based on repeated measurements over time to see its evolution. These three sections allow to read external files (in any format supported by **rPACI**) and also have the option to use demo data to check out how the app works.

The 'Single measurement' menu (accessed through the 'Analyze one patient' tab) contains a browser button, which allows to load one file only, and a check-box to use the available demo data sets. If the check-box is checked, a drop-down menu appears, which allows to select a particular demo data set. This menu contains six different demo data sets, which are in fact real measurements, three of them corresponding to keratoconic corneas (filenames starting with 'K'), one of them to suspicious keratoconic cornea (filename starting with 'S'), and two of them to normal corneas (filenames starting with N). When a file is selected, the app returns a plot depicted by plotSingleCornea() and a table containing the Placido indices. Figure 6 shows a screenshot of the app when the demo file 'K01.txt' is selected.
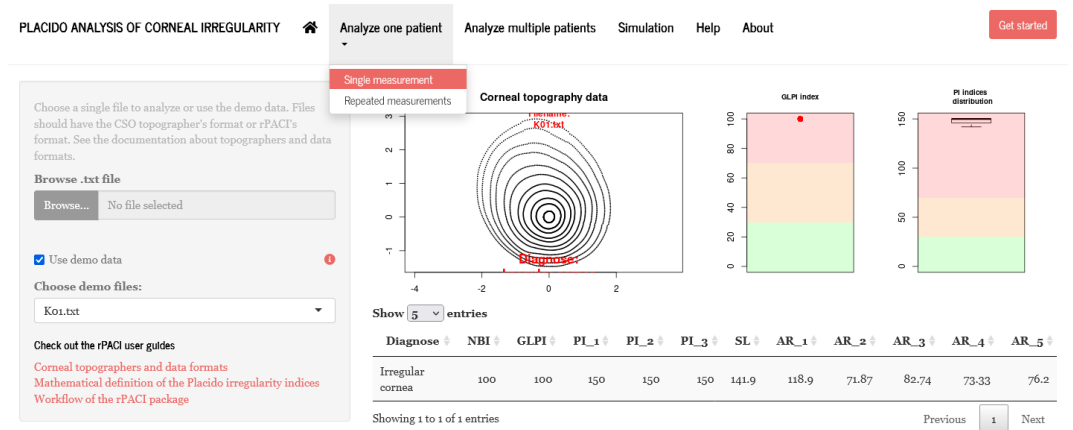

Figure 6: Screenshot of the rPACI shiny app, performing a single measurement corneal analysis. The demo data set K01.txt is chosen.

The 'Repeated measurements' menu (accessed through the 'Analyze one patient' tab) contains a browser button that allows to load several files, whose names account for the chronological order, and a check-box to use the available demo data sets. If the check-box is ticked, the app uses a set of three simulated corneal topographies that model a patient's cornea at three different moments. This menu returns a plot depicted by analyzeEvolution() and a table containing the Placido indices of each data set. Figure 7 shows a screenshot of the app when the demo data is selected.

Figure 7: Screenshot of the rPACI shiny app, performing a repeated measurements corneal analysis. The available demo data is chosen for illustrative purposes.

The 'Analyze multiple patients' tab contains a browser button that allows to load several files, a check-box to use the available demo data sets and two mutually exclusive radio-buttons to chose the

display of the results. If the check-box is marked, the analysis is performed on six available demo data sets. The radio-buttons allow to visualize either the aggregated results (displaying a barplot of the diagnosis values) or the individual plots of each file, in which case a drop-down menu that allows to select a specific case appears. Moreover, this menu returns a table containing the Placido indices of the selected files. Note that the table is interactive, as the rows can be sorted according to a specific column and the number of entries to display can be changed. Figure 8 shows a screenshot of the app when the demo data is selected.
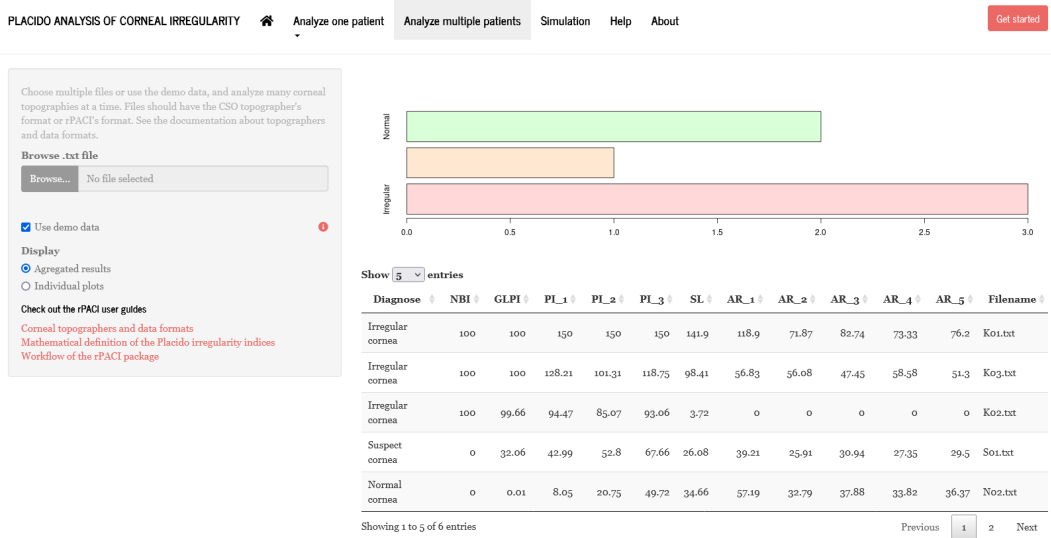
Figure 8: Screenshot of the rPACI shiny app, performing an analysis on multiple corneas simultaneously. The available demo data is chosen for illustrative purposes.

Finally, the 'Simulation' tab allows to simulate a corneal topography data set, specifying up to ten different parameters. The simulated data set can be downloaded, using the button designed for that purpose, and the parameters can be reset to the default values. If the simulated data set is downloaded, the file is saved as a '.txt' file in rPACI format, containing a header with the chosen simulation parameters. This file can later be loaded into the app to perform any kind of analysis. Figure 9 shows a screenshot of the app performing a corneal topography simulation.
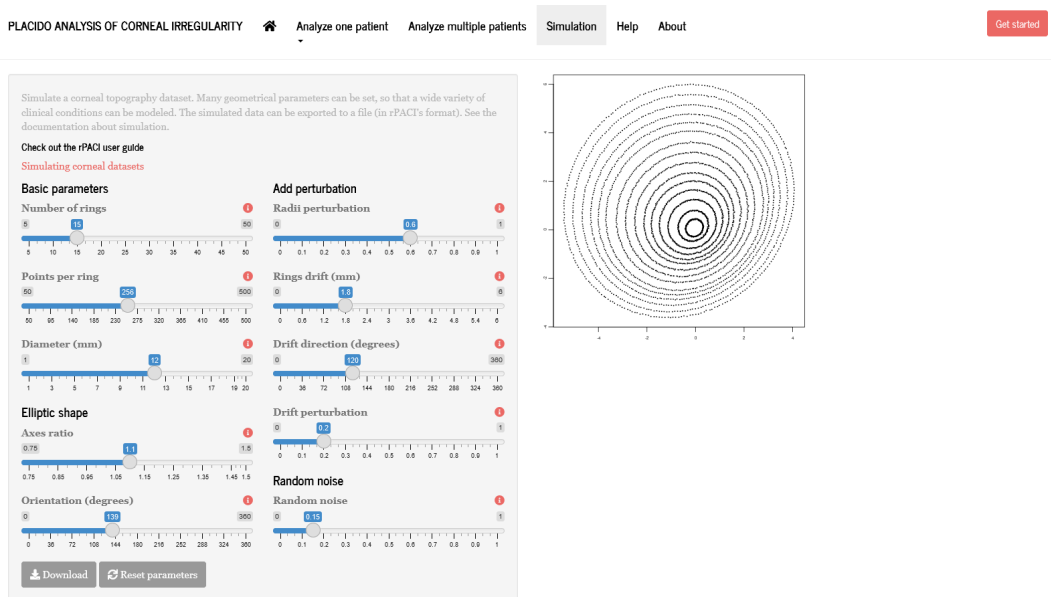

Figure 9: Screenshot of the 'Simulation' tab in the rPACI shiny app.

Summary

Diagnose of corneal diseases, especially keratoconus, is still an important research and clinical problem. Most of the corneal topographers in clinics are Placido disk devices, and several keratoconus indices for these devices exist. In this paper, we have presented the R package **rPACI**, whose aim is to provide practitioners or researchers in the field of ophthalmology with several easy-to-use functions to handle corneal data as provided by a Placido disk corneal topographer. This package includes several indices for detecting keratoconus or other irregularities that are available in the literature and that proved to be useful and accurate in the diagnose.

As the main goal is to facilitate the users the computation and use of the indices, the R package **rPACI** was designed to be as easy-to-use as possible and to provide results (data tables and graphics) that are effortless to interpret. Additionally, a **shiny** web app was developed and openly deployed (https://admaldonado.shinyapps.io/rPACI/) so that **rPACI** can be used in any web browser, in a truly user-friendly graphical interface, without the need of installing R or **rPACI**, nor writing any R code.

In the future, we hope other indices may be included in this package or in other new R packages, facilitating even further their use, improvement, or validation by practitioners or researchers in the topic, and contributing to new advances in the field. Also, we would like that this package and its **shiny** GUI can serve as an inspiration for other researchers in the field of healthcare to make their research results more accessible to clinicians and other researchers.

Acknowledgments

Bibliography

P. A. Accardo and S. Pensiero. Neural network-based system for early keratoconus detection from corneal topography. *Journal of Biomedical Informatics*, 35:151–159, 2002. URL https://doi.org/10.1016/S1532-0464(02)00513-0. [p]

J. L. Alió. *Keratoconus: recent advances in diagnosis and treatment*. Springer, 2016. URL https://doi.org/10.1007/978-3-319-43881-8. [p]

J. L. Alió and M. H. Shabayek. Corneal higher order aberrations: a method to grade keratoconus. *Journal of Refractive Surgery*, 22:539–545, 2006. URL https://doi.org/10.3928/1081-597X-20060601-05. [p]

W. Alkhaldi, D. R. Iskander, A. M. Zoubir, and M. J. Collins. Enhancing the standard operating range of a Placido disk videokeratoscope for corneal surface estimation. *IEEE Transactions on Biomedical Engineering*, 56(3):800–809, 2009. URL https://doi.org/10.1109/TBME.2008.2005997. [p]

J. Bühren, D. Kook, G. Yoon, and T. Kohnen. Detection of subclinical keratoconus by using corneal anterior and posterior surface aberrations and thickness spatial profiles. *Investigative Ophthalmology and Visual Science*, 51:3424–3432, 2010. URL https://doi.org/10.1167/iovs.09-4960. [p]

G. M. Castro-Luna, A. Martínez-Finkelshtein, and D. Ramos-López. Robust keratoconus detection with Bayesian network classifier for Placido-based corneal indices. *Contact Lens and Anterior Eye*, 43 (4):366–372, 2020. URL https://doi.org/10.1016/j.clae.2019.12.006. [p]

W. Chang, J. Cheng, J. J. Allaire, Y. Xie, and J. McPherson. *Shiny: web application framework for R*, 2020. URL https://CRAN.R-project.org/package=shiny. R package version 1.5.0. [p]

J. Cheng and M. J. Druzdzel. AIS-BN: an adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks. *Journal of Artificial Intelligence Research*, 13:155–188, 2000. URL https://doi.org/10.1613/jair.764. [p]

U. de-Sanctis, C. Loiacono, L. Richiardi, D. Turco, B. Mutani, and F. M. Grignolo. Sensitivity and specificity of posterior corneal elevation measured by Pentacam in discriminating keratoconus/subclinical keratoconus. *Ophthalmology*, 115:1534–1539, 2008. URL https://doi.org/10.1016/j.ophtha.2008.02.020. [p]

J. Espinosa, D. Mas, J. Pérez, and A. B. Roig. Open-access operating algorithms for commercial videokeratographer and improvement of corneal sampling. *Applied Optics*, 52(7):C24–C29, 2013. URL https://doi.org/10.1364/AO.52.000C24. [p]

R. Fan, T. C. Chan, G. Prakash, and V. Jhanji. Applications of corneal topography and tomography: A review. *Clinical & Experimental Ophthalmology*, 46(2):133–146, 2018. URL https://doi.org/10.1111/ceo.13136. [p]

R. Fung and K. C. Chang. Weighting and integrating evidence for stochastic simulation in Bayesian networks. *Machine Intelligence and Pattern Recognition*, 10:209–219, 1990. URL https://doi.org/10.1016/B978-0-444-88738-2.50023-3. [p]

I. Issarti, A. Consejo, M. Jiménez-García, S. Hershko, C. Koppen, and J. J. Rozema. Computer aided diagnosis for suspect keratoconus detection. *Computers in biology and medicine*, 109:33–42, 2019. URL https://doi.org/10.1016/j.compbiomed.2019.04.024. [p]

I. Issarti, A. Consejo, M. Jiménez-García, E. O. Kreps, C. Koppen, and J. J. Rozema. Logistic index for keratoconus detection and severity scoring (Logik). *Computers in Biology and Medicine*, 122:103809, 2020. URL https://doi.org/10.1016/j.compbiomed.2020.103809. [p]

L. Llorente, S. Barbero, J. Merayo, and S. Marcos. Total and corneal optical aberrations induced by laser in situ keratomileusis for hyperopia. *Journal of Refractive Surgery*, 20:203–216, 2004. URL https://doi.org/10.3928/1081-597X-20040501-03. [p]

N. Maeda, S. D. Klyce, M. K. Smolek, and H. W. Thompson. Automated keratoconus screening with corneal topography analysis. *Investigative Ophthalmology and Visual Science*, 35:2749–2757, 1994. [p]

L. J. Maguire and W. M. Bourne. Corneal topography of early keratoconus. *American Journal of Ophthalmology*, 108:107–112, 1989. URL https://doi.org/10.1016/0002-9394(89)90001-9. [p]

A. Martínez-Abad and D. P. Piñero. New perspectives on the detection and progression of keratoconus. *Journal of Cataract & Refractive Surgery*, 43(9):1213–1227, 2017. URL https://doi.org/10.1016/j.jcrs.2017.07.021. [p]

S. Ortiz-Toquero, I. Fernández, and R. Martín. Classification of keratoconus based on anterior corneal high-order aberrations: a cross-validation study. *Optometry and Vision Science*, 97:169–177, 2020. URL https://doi.org/10.1097/OPX.0000000000001489. [p]

D. P. Piñero. Technologies for anatomical and geometric characterization of the corneal structure and anterior segment: A review. *Seminars in Ophthalmology*, 30(3):161–170, 2015. URL https://doi.org/10.3109/08820538.2013.835844. [p]

G. Prakash, A. Agarwal, A. I. Mazhari, G. Kumar, P. Desai, D. A. Kumar, and S. Jacob. A new, pachymetry-based approach for diagnostic cutoffs for normal, suspect and keratoconic cornea. *Eye (London)*, 26:650–657, 2012. URL https://doi.org/10.1038/eye.2011.365. [p]

Y. S. Rabinowitz. Keratoconus. *Survey of Ophthalmology*, 42:297–319, 1998. URL https://doi.org/10.1016/s0039-6257(97)00119-7. [p]

Y. S. Rabinowitz and K. Rasheed. KISA % index: a quantitative videokeratography algorithm embodying minimal topographic criteria for diagnosing keratoconus. *Journal of Cataract and Refractive Surgery*, 25:1327–1335, 1999. URL https://doi.org/10.1016/s0886-3350(99)00195-9. [p]

D. Ramos-López and A. D. Maldonado. *rPACI: Placido Analysis of Corneal Irregularity*, 2021. URL https://CRAN.R-project.org/package=rPACI. R package version 0.2.1. [p]

D. Ramos-López, A. Martínez-Finkelshtein, G. M. Castro-Luna, D. P. Piñero, and J. L. Alió. Placido-based indices of corneal irregularity. *Optometry and Vision Science*, 88(10):1220–1231, 2011. URL https://doi.org/10.1097/OPX.0b013e3182279ff8. [p]

D. Ramos-López, A. Martínez-Finkelshtein, G. M. Castro-Luna, N. Burguera-Gimenez, A. Vega-Estrada, D. P. Piñero, and J. L. Alió. Screening subclinical keratoconus with Placido-based corneal indices. *Optometry and Vision Science*, 90(4):335–343, 2013. URL https://doi.org/10.1097/OPX.0b013e3182279ff8. [p]

D. Ramos-López, A. R. Masegosa, A. Salmerón, R. Rumí, H. Langseth, T. D. Nielsen, and A. L. Madsen. Scalable importance sampling estimation of Gaussian mixture posteriors in Bayesian networks. *International Journal of Approximate Reasoning*, 100:115–134, 2018. URL https://doi.org/10.1016/j.ijar.2018.06.004. [p]

R. H. Rand, H. C. Howland, and R. A. Applegate. Mathematical model of a Placido disk keratometer and its implications for recovery of corneal topography. *Optometry and Vision Science*, 74:926–930, 1997. URL https://doi.org/10.1097/00006324-199711000-00026. [p]

J. B. Randleman, B. Russel, M. A. Ward, K. P. Thompson, and R. D. Stulting. Risk factors and prognosis for corneal ectasia after LASIK. *Ophthalmology*, 110:267–175, 2003. URL https://doi.org/10.1016/s0161-6420(02)01727-x. [p]

S. A. Read, M. J. Collins, D. R. Iskander, and B. A. Davis. Corneal topography with Scheimpflug imaging and videokeratography: comparative study of normal eyes. *Journal of Cataract and Refractive Surgery*, 35(6):1072–1081, 2009. URL https://doi.org/10.1016/j.jcrs.2009.01.020. [p]

J. J. Rowsey, A. E. Reynolds, and R. Brown. Corneal topography: corneascope. *Archives of Ophthalmology*, 99(6):1093–1100, 1981. URL https://doi.org/10.1001/archopht.1981.03930011093022. [p]

S. Samapunphong and D. Azar. Placido and elevation-based corneal topography. a review. *Ophthalmology Clinics of North America*, 11(3):311–329, 1998. URL https://doi.org/10.1016/S0896-1549(05)70059-6. [p]

M. Scutari. Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35 (3):1–22, 2010. URL https://doi.org/10.18637/jss.v035.i03. [p]

M. Scutari and R. Ness. *bnlearn: Bayesian network structure learning, parameter learning and inference*, 2019. URL https://CRAN.R-project.org/package=bnlearn. R package version 4.5. [p]

S. Yousefi, E. Yousefi, H. Takahashi, T. Hayashi, H. Tampo, S. Inoda, Y. Arai, and P. Asbell. Keratoconus severity identification using unsupervised machine learning. *PLOS ONE*, 13(11):1–11, 2018. URL https://doi.org/10.1371/journal.pone.0205998. [p]

Darío Ramos-López
Department of Applied Mathematics, Materials Science and Engineering and Electronic Technology
Rey Juan Carlos University
Calle Tulipán, s/n, 28933, Móstoles, Madrid
Spain
ORCiD: https://orcid.org/0000-0002-6127-6559
dario.ramos.lopez@urjc.es

Ana D. Maldonado
Department of Mathematics
University of Almería
Carretera Sacramento s/n, 04120 La Cañada, Almería
Spain
ORCiD: https://orcid.org/0000-0001-8253-2526
ana.d.maldonado@ual.es