

iRRA package.

Display the Region of Interest and compute the Ratio of Relevant Areas

This package allows the user to plot a ROC curve and a Region of Interest (RoI), highlighting the RoI space under the curve, and to compute the Ratio of Relevant Areas (RRA). It is also possible to confront different ROC curves' RRAs and average them out.

Functions available:

- [rra](#)
- [rra.plot](#)
- [rra.print](#)
- [rra.test](#)
- [rra.average](#)

RRA

This is the main function of the iRRA package. It builds the Region of Interest (RoI) given the coordinates of the ROC curve, the number of actual positives (AP) and actual negatives (AN) and the reference values of the performance metrics, and then it computes the Ratio of Relevant Areas (RRA). It returns a "rra_result" object, a named list of "rra_result" class. This object can be printed and plotted. Additionally, two "rra_result" object can be compared (`rra.test`) and a list of "rra_result" objects can be averaged out by `rra.average`.

USAGE

```
rra(roc_x, roc_y, AP, AN,  
    precision = FALSE, c_precision = "pop", p_precision = -1,  
    recall = FALSE, c_recall = "pop", p_recall = -1,  
    fm = FALSE, c_fm = "pop", p_fm = -1,  
    npv = FALSE, c_npv = "pop", p_npv = -1,  
    specificity = FALSE, c_specificity = "pop", p_specificity = -1,  
    fallout = FALSE, c_fallout = "pop", p_fallout = -1,  
    nm = FALSE, c_nm = "pop", p_nm = -1,  
    j = FALSE, c_j = -1,  
    markedness = FALSE, c_markedness = -1,  
    phi = FALSE, c_phi = 0.4,  
    ncost = FALSE, c_ncost = "uses_mu", lambda = c(-1), mu = 1,  
    print = TRUE, plot = TRUE, ...)
```

ARGUMENTS

`roc_x, roc_y`

The values of the ROC curve's points

`AP`

The number of actual positives. This value represents the number of positive responses (the "1" values) used to build the ROC curve. It must be greater than 0

`AN`

The number of actual negatives. This value represents the number of negative responses (the "0" values) used to build the ROC curve. It must be greater or equal to 0

`precision, recall, fm, npv,
specificity, fallout, nm,
j, phi, markedness, ncost`

| | |
|---|---|
| | Indicate which metrics the user wants to use as reference values for the generation of the Region of Interest |
| <code>c_precision, c_recall, c_fm, c_npv, c_specificity, c_fallout, c_nm</code> | The reference values. It must be between 0 and 1. It can also indicate that the reference value is selected using a uniform random policy ("uni") method (<code>c_... = "uni"</code>). If <code>c_... = "pop"</code> it means that a "uni" method with $p(m)=AP/n$ is used ("Proportion Of Positives" policy) |
| <code>p_precision, p_recall, p_fm, p_npv, p_specificity, p_fallout, p_nm</code> | The unified probabilities used by the "uni" method |
| <code>c_j</code> | The reference value for Youden's J. It must be between -1 and 1 |
| <code>c_markedness, c_phi</code> | The reference values for Phi and Markedness. It must be between 0 and 1 |
| <code>c_ncost</code> | The reference value for the Normalized Cost. It must be between 0 and the NC value selected with the "pop" method. This value will define the cost reduction index (μ). The default value, "uses_mu", indicates that the user wants to use the μ value directly |
| <code>lambda</code> | The value of False Negative and False Positive cost ratio as $(c_{FN} / (c_{FN}+c_{FP}))$. It must be between 0 and 1. Indicating two values means that the user wants to consider a range of lambda values |
| <code>mu</code> | The reduction cost index value. It's used only if <code>c_ncost = "uses_mu"</code> |
| <code>print</code> | If the user wants to print the result. For more information, check rra.print |
| <code>plot</code> | If the user wants to plot the ROC curve and the RoI. For more information, check rra.plot |
| <code>...</code> | Other arguments for rra.print and rra.plot . |

DETAILS

The Region of Interest (RoI) represents the points in the ROC space that have a better performance value than the reference values.

Every performance metrics corresponds to a specific border of the RoI. It is possible to use multiple metrics and different methods, but it is important to keep in mind that some border could be always greater than others within the ROC space. In this case some borders will obscure the others.

Additionally, some special values will be not very significant. For instance, a recall reference value of 1 will result in a non-existent RoI. Its RRA will be 0 unless the ROC curve is perfect ($AUC = 1$). On the other hand, if recall is equal to 0 the RoI will correspond to the ROC space, therefore the RRA will be equal to the AUC of the ROC curve.

The default value for phi, 0.4, represents a medium-strong association between a model and actual positiveness.

A "rra_result" object contains the points of the ROC curve, the coordinates of the RoI and the RoI under the curve, the RRA value and the list of the performance metrics considered.

ERRORS

The function will stop if `roc_x` and `roc_y` has different length or have values greater than 1 or lesser than 0. It will also stop if the other parameters have invalid values

NOTE

Note that the `precision("uni")` border will be $y=x$ for every $p(m)$. For this reason, using `c_precision = "uni", p_precision = (0<p<1)` will generate the same border as using `c_precision = "pop"`. This is also true for the NPV border.

RRA.PLOT

This function will plot a "rra_result" object obtained by the [rra](#) function. It will draw the RoI space together with the ROC curve. It will also highlight the RoI under the curve.

USAGE

```
rra.plot(rra_result, ADD = FALSE,
         cROI = "grey", cUnder = "light blue",
         type = "l", diagonal= TRUE,
         xlim = c(0,1), ylim = c(0,1), asp = 0,
         xlab = "FP/AN", ylab = "TP/AP",
         main = "ROC curve and RoI", ...)
```

ARGUMENTS

| | |
|--|--|
| <code>rra_result</code> | A "rra_result" object obtained by rra |
| <code>ADD</code> | If the user wants to keep the elements he already drawn |
| <code>cROI</code> | The color of the RoI space. By default, it's grey. It is used as a normal <code>col</code> parameter of any plot function |
| <code>cUnder</code> | The color of the RoI space under the ROC curve. By default, it's light blue. It is used as a normal <code>col</code> parameter of any <code>plot.default</code> function |
| <code>diagonal</code> | If the user wants to plot the $y = x$ diagonal |
| <code>type, xlim, ylim, asp, xlab, ylab, main</code> | These are normal <code>plot.default</code> parameters but with a default value |
| <code>...</code> | Others <code>plot.default</code> parameters that can be used to personalize the ROC curve, such as <code>lwd</code> |

NOTE

Note that the RoI is plotted as a polygon. It could hide other plotted structures if `ADD = TRUE`. In this case, consider using something like `adjustcolor("color_name", alpha.f=0.5)` to give some transparency to the polygons.

If `main = ""` or `NULL`, `xlab = ""` or `NULL` or `ylab = ""` or `NULL`, the margins will adapt so that as little space as possible is wasted

RRA.PRINT

Taking as input a "rra_result" object, it will print the RoI's borders information and the resulting Ratio of Relevant Areas (RRA)

USAGE

```
rra.print(rra_result)
```

ARGUMENTS

| | |
|-------------------------|----------------------------------|
| <code>rra_result</code> | The "rra_result" object to print |
|-------------------------|----------------------------------|

DETAILS

This function will print the meaning of the Region of Interest selected; it will highlight what performance values are matched by all the points within the RoI space.

If `rra_result` represents the AUC of the ROC curve it will just print the resulting Area Under the Curve.

RRA.TEST

This function tests two ROC curves and check which one has the best RRA. It can get as input either two "rra_result" objects or two ROC curves based on the same data set (AN and AP values are the same for both curves)

USAGE

```
rra.test(...)
## Default S3 method:
rra.test(roc_x_1, roc_y_1, roc_x_2, roc_y_2, AP, AN, ...)
## S3 method for class 'rra_result'
rra.test(rra_result_1, rra_result_2, plot = TRUE,
         col1 = "black", col2 = "red",
         cUnder1 = "light blue", cUnder2 = "yellow", lty1 = "dashed", ...)
## S3 method for class 'list'
rra.test(list1, list2 = c(), AP, AN, plot = TRUE, legend = FALSE,
         colBest = "black", lwdBest = 2, ltyBest = "solid",
         cUnderBest = "light blue", lwdOther = 1, ltyOther = "dashed",
         addcUnderOthers = FALSE, ...)
```

ARGUMENTS

| | |
|----------------------|---|
| <code>roc_x_1</code> | The x values of the first ROC curve's points |
| <code>roc_y_1</code> | The y values of the first ROC curve's points |
| <code>roc_x_2</code> | The x values of the second ROC curve's points |
| <code>roc_y_2</code> | The y values of the second ROC curve's points |

| | |
|-----------------|---|
| AP | The number of actual positives. This value represents the number of positive responses (the "1" values) used to build the ROC curve. It must be greater than 0 |
| AN | The number of actual negatives. This value represents the number of negative responses (the "0" values) used to build the ROC curve. It must be greater or equal to 0 |
| rra_result_1 | The first "rra_result" object to test |
| rra_result_2 | The second "rra_result" object to test |
| plot | If the user wants to plot the two ROC curves and their RoIs |
| col1 | The color of the first ROC curve when plotted |
| col2 | The color of the second ROC curve when plotted |
| cUnder1 | The color of the first RoI space under the curve |
| cUnder2 | The color of the second RoI space under the curve |
| lty1 | The style of the "worst" curve. It corresponds to lty of the plot.default function |
| list1 | Either a list of "rra_result" object or a list of vectors containing the x values of the ROC curves |
| list2 | A list of vectors containing the y values of the ROC curves. It's empty by default as it is not used if list1 is a list of "rra_result" object |
| legend | If the user wants to plot a legend of the drawn ROC curves |
| colBest | Line's color of the ROC curve with the best RRA value |
| lwdBest | Line's width of the ROC curve with the best RRA value |
| ltyBest | Line's style of the ROC curve with the best RRA value |
| cUnderBest | Color of the RoI under the curve with the best RRA value |
| lwdOther | Line's width of the other ROC curves |
| ltyOther | Line's style of the other ROC curves |
| addcUnderOthers | If the user wants to draw the RoI under the other curves |
| ... | Other parameters that can be set. |

Especially:

- For the default method all the parameters for [rra](#) and the method for class "rra_result"
- For the method for class "rra_result" the parameters for [rra.plot](#) to modify the appearance of the ROC curve
- For the method for class "list" all the parameters for [rra](#) and for [rra.plot](#) (except for col, lwd, lty and cUnder, more informations in the Details section)

DETAILS

It is possible to plot the two ROC curves and their Rols.

The method for class 'list' will return a vector of the indices of the parameters' list ordered by their RRA values

It's possible to test any "rra_result" object, but when two curves are generated by different data sets (AN and AP values have different values for the two curves), or when the RoI is computed by different performance metrics, the user will be warned.

[rra.plot](#) parameters `col`, `lwd`, `lty` and `cUnder` are overwritten by `colBest`, `lwdBest`, `ltyBest` and `cUnderBest` in the method for class "list". It is possible to use both but it will give an error if used at the same time.

The color of the other curves in `rra.test.list` is randomly selected by a list of colors.

By default the RoI under the curve of the other curves will not be drawn to make the plot easier to understand. Consider that, if the "rra_result" object considered have different RoI and RoI under the curve (different data-set or different reference metrics), one may want to set `AddcUnderOthers=TRUE`

RRA.AVERAGE

This function averages the RRA values of multiple "rra_result" objects. It will also returns the median value(s) and the standard deviation.

USAGE

```
rra.average(rra_results, print = TRUE)
```

ARGUMENTS

`rra_results` The list of "rra_result" objects. Note that `rra_results` must be a "list" object
`print` If the user wants to print the result

DETAILS

This function returns the average value together with the standard deviation and the median value(s).

If `print = TRUE` it will print all three values