Communication Papers of the of the 17th Conference on Computer Science and Intelligence Systems pp. 241–249

DOI: 10.15439/2022F32
ISSN 2300-5963 ACSIS, Vol. 32

Visually Enhanced Python Functions for Clinical Equality of Measurement Assessment

Mauro Nascimben
Università del Piemonte Orientale,
Dept. of Health Sciences,
Novara, Italy
Enginsoft SpA, Padua, Italy
Email: m.nascimben@enginsoft.com

Lia Rimondini
Università del Piemonte Orientale,
Dept. of Health Sciences,
CAAD, Corso Trieste 15,
Novara, Italy
Email: lia.rimondini@med.uniupo.it

***Abstract*—Equivalence testing requires specific procedures usually provided by specialized statistical software. The proposed package includes customized methods to assess biomedical equivalence and focuses on translating the outcomes into visual reports. The functions are coded in an object-oriented framework, contain improved plots or novel graphs to facilitate interpretation of the results, and are accompanied by console textual outputs to support users with additional explanations. Special attention has been devoted to verifying the preliminary assumptions of the statistical tests with automatic routines. The current module covers four aspects of biomedical statistics (equivalence, Bland-Altman and ROC analyses, effect size, and confidence intervals interpretation), offering these methodologies to the biomedical community as accessible stand-alone functions. The manuscript defines software's functions and innovations with examples and theoretical explanations.**

I. Introduction

COMPARATIVE statistical tests could not address the interchangeability of measurements obtained from different laboratory devices or the similarity between two treatments. For example, the output values returned by a new and an old laboratory system require specific statistical analysis to demonstrate that the outcomes from the two machines are equivalent [1]. In comparative inference, the lack of a significant effect does not necessarily mean equality. Analyzing the equivalence means reversing the null hypothesis of standard biostatistical testing by validating the alternative hypothesis of no difference between measurements. The importance of this topic is particularly relevant for the medical sector, especially for the biopharmaceutical industry, with guidelines for therapeutic equivalence between drugs established by regulatory agencies like USA Food and Drug Administration (i.e., FDA) [2]. According to FDA, therapeutic equivalence implies that the two drugs have the same clinical effect on patients and follow the same safety profile. The FDA requires *two one-sided tests* procedure (i.e., TOST) to prove that formulations of two drugs are bioequivalent. The TOST approach determines, for a certain significance level α, if the $(1-\alpha)\times 100$ confidence interval of the average difference between drugs falls inside the regulatory boundaries $\pm\delta$. The equivalence implies that the efficacy of the new therapy is within δ units from the efficacy of a drug taken as reference, usually already on the market. FDA recommends a 90% confidence interval to determine biosimilarity, even if recently scholars suggested raising the interval range to 95% [3]. Thanks to the simplicity of TOST to prove similarity, the methodology started to be applied in other domains like medicine and chemistry [4].

The *sensitivity* (aka true positive fraction) and *specificity* (aka true negative fraction) assessment is another kind of equivalence analysis between diagnostic tests characterized by dichotic outcomes. For their estimation, a cross table is initially assembled containing the frequencies of a laboratory outcome versus the truth "disease status" in a population of subjects. Alternatively, the frequency table could be employed to contrast one lab measurement versus a gold standard. The continuous laboratory outcomes are categorized into "positives" or "negatives" based on a threshold: subjects with lab values above the threshold (or below, depending on the analysis) are labeled as positives, meaning the lab test detected the disease. Conversely, all the others are the negatives from the laboratory results. The same nomenclature is used to distinguish subjects' actual status: those carrying the illness (positives) or not (negatives). The frequency table contains the terms "false positives" (the lab procedure erroneously identified the illness in certain samples) and "false negatives" (the lab test incorrectly labeled a few samples as disease-free), also known as type I and type II errors, respectively. The sensitivity and specificity of a lab test can be deducted from the frequency cross table and describe the validity of the diagnostic examination, and they are assumed to be independent of the prior probability of having a disease [5]; however, they depend on the threshold selected to categorize the laboratory outcomes. A series of thresholds could be picked out to circumvent this limitation, thus obtaining different sensitivity and specificity pairs for each cut-off point. The *receiver operating characteristic* (i.e., ROC) curve shows over a graph the sensitivity and (1- specificity) values corresponding to each threshold. The area under the ROC curve is a metric to judge the efficacy of two laboratory tests in determining the same disease or the inter-observer variability. This latter situation is

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska Curie grant agreement No 860462.

critical in medical exams involving the interpretation of the outcomes where diagnostic procedures are dependent on the investigator's skills and experience (for example, microscopic examination of cytologic samples or radiographic imaging). In addition, a certain degree of disparity between laboratory data revolves around different protocols. Consequently, the same test results might not be in perfect accordance if carried out in different hospitals or inside the same hospital by different operators.

Another pillar in the investigation of the agreement between measurements is the *Bland-Altman* analysis [6] (i.e., BA). The approach verifies if the difference between observations is contained inside acceptable agreement limits. In the original method, the two limits correspond to the $\overline{m}\pm1.96\times sd_m$, where $\overline{m}$ is the mean difference, and sd_m is its standard deviation. They approximate the 2.5^{th} and 97.5^{th} percentiles of the distribution of the differences, theoretically enclosing 95% of the differential values. Given the differential measurements obtained from a reference method and a new device $(New-Ref)$, approximately 95% of the time, the measurements from the new machine should be $\overline{(New-Ref)}-1.96\times sd$ units below the reference and $\overline{(New-Ref)}+1.96\times sd$ units above the reference. Before employing BA analysis, researchers should check the assumptions that the differences are normally distributed and exhibit constant variance. Moreover, studies usually include the degree of uncertainty in estimating the limits of agreement by reporting their confidence intervals. The 95% confidence interval around the agreement limits $\overline{(New-Ref)}\pm1.96\times sd$ can be computed as $\pm1.96\times SE$ where $SE=sd\times\sqrt{3/n}$ is the standard error of the limits and n the number of samples. Sampling errors might cause the agreement limits fluctuations incorporated by the confidence intervals; they restrict the initial agreement limits, but they will scale down as the sample size increases. The decision to accept the agreement intervals is taken according to clinical and biological objectives because BA analysis does not provide conclusions for statistical inference.

Confidence intervals (i.e., CI) play a fundamental role in addressing bioequivalence but can also highlight situations where a new treatment is *non-inferior* or *superior* to an existing one. The non-inferiority test requires the identification of a lower boundary $-\delta$, needed to verify if the CI obtained by the difference between treatments remains above it. A treatment is considered "as good as" the reference method if this happens. A drug's superiority to an available one is determined if the difference between treatments' CI lies above zero. In general, CIs express a certain level of probability that by randomly sampling a population an infinite number of times, one can obtain the true population parameter inside the interval. Closely connected to CI is the concept of quantifying the amount of difference between treatments to facilitate efficacy comparisons (aka effect size [7]). Standardizing the size of treatment effects produces unit-free measures that identify analogies between biomarkers and allow meta-analysis.

A. Aim of the Proposed Software Tools

Python is a general-purpose programming language widely adopted by the data science community. Its intuitive syntax and universality allow scholars to deal with various aspects of quantitative analysis. However, as for other non-commercial software, it relies on the efforts of the users to create libraries of functions able to solve specific data analysis tasks. Another popular free program for data science, for certain aspects complementary to python, is R, a statistically centered software. R might be preferred for deep statistical analysis over python, especially for creating visually interpretable statistical graphs. Alternatively, commercial software provides valuable statistical analysis tools. The present work presents a set of statistical functions in an object-oriented python library to deal with several aspects of equivalence testing. The importance is two-fold: provide specialized or improved functions usually found in other environments and share source code for future reusability by the scientific community. This latter aspect has been emphasized by the European Union's efforts for the "open science" paradigm [8].

II. Python functions

This manuscript section explains the essential functions and innovations included in the package referencing the relevant theoretical parts but accompanied by textual descriptions rather than formulas. This approach has been preferred to focus on the graphical interpretation of the outcomes. However, only a selected number of plots could be included in the current document, and only a few console outputs were discussed. The present library of visual functions has been coded in an object-oriented fashion; for didactic purposes, code snippets have been inserted throughout the document accompanied by full-length imports so users can match functions to the code organized in the Github folders. Code examples do not incorporate all function inputs, accepting the tacit defaults. Each function also provides textual outputs in the console to strengthen the analysis' conclusions. One of the key features is the preference for probability density distributions to visualize continuous data representations over a range of values rather than histograms: this avoids the process of selecting bin widths. Apart from standard python libraries, the module requires a few external dependencies: Numpy, Scipy, Pandas, Seaborn, and Matplotlib. Table I outlines the methodologies subdividing the procedures into four macro-areas as described in the "Introduction."

A. Bland-Altman Analysis

The proposed implementation controls the preliminary assumptions required to run this investigation. Indeed, calculating the limits of agreement depends on the prerequisite that the measurements' differences follow a normal homoscedastic distribution [9]. The function's code silently checks these assumptions and forces the user to meet this specification. Normalcy is assessed by the Shapiro-Wilks test, while homogeneity of variance is by Levene statistics. A practical example has been provided to illustrate the application of the

TABLE I
OVERVIEW OF THE METHODOLOGIES IMPLEMENTED IN THE PYTHON FUNCTIONS

Macro-area	Folder	Operations	Characteristics
Equivalence	EIS	TOST	Independent or paired two one-sided t-tests
	EIS	TOST	Fixed margin $\delta = f \times \sigma_{Ref}$ equivalence test
	EIS	TOST	Modified Wald with maximum likelihood estimator
	EIS	TOST	Paired inputs, sample size and statistical power dedicated functions
	EIS	TOST	Heteroscedastic inputs, sample size and statistical power dedicated functions
	EIS	Non-inferiority	Measurement vs. parameter or two-sample statistics
	EIS	Superiority	Measurement vs. parameter or two-sample statistics
	EIS	ROPE	Region of practical equivalence by equally-tailed or highest density intervals
ROC	ROC	2 × 2 freq. table	Confusion matrix and derivation of 34 performance indexes
	ROC	Radar plots	Circular graphs to compare cross-table's indexes (single/paired/bars) between two treatments
	ROC	ROC visual interpr.	ROC computation, Youden index, k-index, MID, non-parametric CI
	ROC	ROC Statistics	DeLong and Venkatraman independent or dependent methods
	ROC	Ranking plots	False positive rate vs. true positive rate with statistics and precision vs. recall with AUC
Effect size and CI	ES	Cohen's d	Independent or paired inputs, non-overlapping indexes
	ES	Re-testing	Repeated measurements' minimal detectable change
	ES	Responsiveness	Guyatt coeff., standardized response mean, effecct size, normalized ratio, reliable change index
	ES	Exploratory stats	Equivalence estimation based on CI analysis between biomarkers
	CI	Margin value study	Stacked CI representations to study the optimal equivalence margin
	CI	Paired Cat's Eyes	Biased or unbiased representation of two biomarkers
	CI	Car's Eye vs. p value	Single biomarker visual two-tailed analysis of CI vs. p significance
Bland-Altman	EQU	BA analysis	Revisited graphical interpretation, approximated and exact limits, min. detectable change
	EQU	Regress. diagnostics	Residuals interpretation, spread-location plot with Cook distances, influential points graphs
	EQU	Inherent imprecision	Graphical inherent imprecision with Chebyshev interval adjustment

Bland-Altman analysis. Two variables simulating the data of two different treatments were generated randomly by sampling two Gaussian distributions: the first variable represents a new treatment (`var1`), while the second is the reference method (`var2`). As summarized below, the initialization of `eq_BA` with default parameters creates an object whose methods incorporate the BA evaluation procedures:

```
from equiv_med.EQU import eq_BA
BA=eq_BA.BA_analysis(var1, var2)
BA.run_analysis()
BA.minimal_detectable_change()
```

The output of `run_analysis` is a graph exhibiting a novel interpretation of the Bland-Altman plot, as shown in Fig. 1. The typical illustration produced by statistical programs contains less information and is featured in Figure 12 of [9]. In the peculiar design supplied by the python function, each differential value is shown as a gray bar at the bottom of Fig. 1; the probability distribution overlays the graph, and the data range is the dark green horizontal dashed line. The light green dashed vertical line references the zero while the red dashed vertical line is the mean difference between measurements (aka the *bias*). Theoretically, if two measurements are equivalent, their difference should be zero. At the bottom, the standard deviation (i.e., SD) and standard error (i.e., SE) of the mean difference are visualized as horizontal bars. The SD line encloses 68% of the differences between measurements, providing the spread of the central portion of differences. The blue dashed vertical lines are the default limits of agreement at 1.96, with shaded areas representing their confidence intervals. The "BA Prop" rectangle at the top shows the effective percentage of values encompassed by the approximated limits of agreement (theoretically expected 95%, actual 94%); the rectangle length is scaled to show the proportion of actual data exceeding computed limits. Indeed, the "BA Prop" rectangle does not overcome the computed limits of 1% in length, and its horizontal boundaries are inside the blue vertical dashed lines. The orange dashed vertical lines are the exact limits

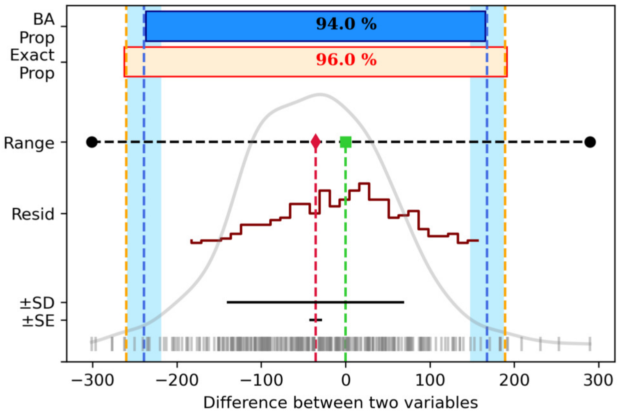

Fig. 1. Alternative design of the Bland-Altman plot as found in the `BA_analysis` class, using function `run_analysis`.

of agreement obtained by the procedure of [10]. The “Exact prop” rectangle shows the proportion of values falling inside the exact limits of agreement, with the rectangle length scaled to highlight the actual data behavior compared to the computed limits. The last information portrayed by the figure is the residuals of the linear regression between measurement values. The residuals histogram is expected to gather values around zero, and dispersion around it may emphasize the incomplete agreement between treatment measurements. The graph is accompanied by textual statistical information:

- Identification of a systematic difference between treatments (also called a fixed bias) or not by one-sample t-test
- The proportion of values in the two measurements that fall outside the approximated or exact limits of agreement

The second function in the code above allows users to detect the `minimal_detectable_change` parameter from the width of the approximated and exact limits of agreement. The minimal detectable change is the most negligible modification not attributed to an instruments measurement error. If the parameter values are less than a “minimal clinically important change” deduced from literature or clinical practice, the methods are in accordance with each other.

The sample size is a critical element in BA analysis for biomarker compliance: the class `eq_BA` offers two features to help researchers study this aspect for the exact limits of agreement (methods `exact_Bound_sample_size` and `exact_Bound_assurance`). The first function establishes the sample size by building the two-sided equal-tailed interval, given a certain CI width around the limits and significance level (i.e., $\alpha = 0.05$). Users may vary the width input parameter to adjust the required sample size. The second function supports the users in setting up the sample size, relating the theoretical to the actual probability of getting the desired CI width.

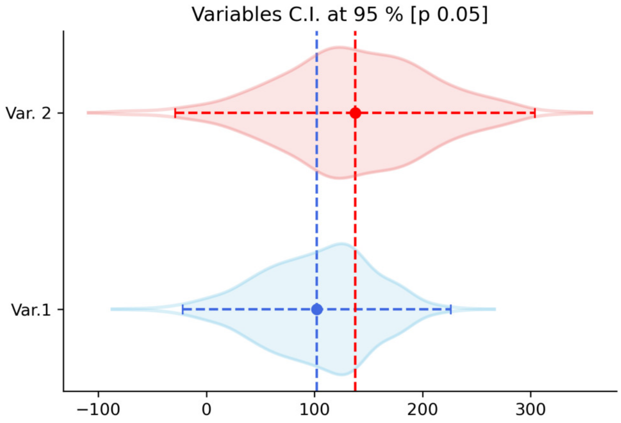

Fig. 2. Comparison between Cat’s-Eye plots in `run_ce_unbiased` function.

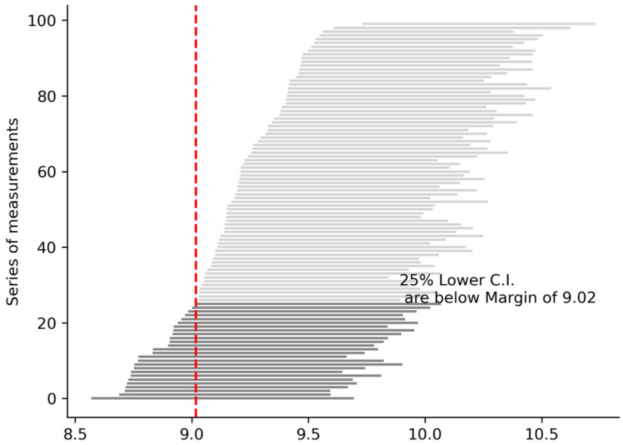

Fig. 3. Stacked CI representations of `decision_margin` function.

B. Visual Representations of Confidence Intervals

Interval estimation is directly related to p-values of null hypothesis statistical significance testing and helps interpret the precision of effect size [11]. CIs are advantageous in meta-analysis to compare data from previous studies, or in the case of longitudinal studies, they provide quickly interpretable insights. The proposed python function computes CI visual analysis creating a Cat’s-Eye plot accompanied by statistical information. The function call `eq_CatEyes` is summarized below, keeping the same random variable characteristics to simulate two measurements as in the previous example:

```
from equiv_med.CI import eq_Cateyes
ce=eq_Cateyes.Cat_Eye_2var(var1, var2)
ce.run_ce_unbiased(95)
ce.single_cat_eye(var1,95)
```

The line of code with `run_ce_unbiased` verifies if the 95% CI of the first measurement is contained in the second one and vice-versa at default $\alpha = 0.05$. The console output is “At 5.0% probability: The first variable C.I. is entirely inside the C.I. of the second variable (open interval)”. If the two CI match, the function inspects the interval type by checking the inclusion of the endpoints (open interval) or not (close interval). Calling `run_ce_unbiased`, users can plot the Cat’s-Eyes of the two measurements, with eye shape depicting the probability density function of the data mirrored vertically (Fig. 2). The pupil of the eye, marked as a dot, is the mean, with horizontal lines providing references to evaluate the CI range and vertical dashed lines to feature the difference between averages. Alternatively, function `run_ce` shows the eyes with standard gaussian density estimation. Another function, `single_cat_eye`, allows users to investigate the plausibility of confidence intervals at different α values.

A different class, `Id_margin`, contains a method `decision_margin` that illustrates the positioning of a series of simulated CIs by bootstrap statistics built knowing the average and coefficient of variation of a biomarker (Fig. 3). The CIs are sorted and plotted together with a value acting as an equivalence margin: it might help examine the behavior of a specific regulatory boundary during equivalence testing.

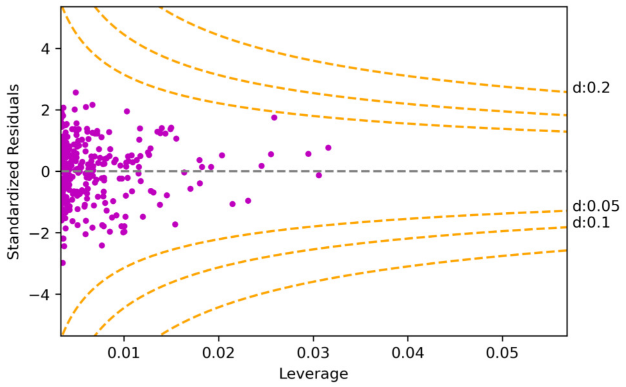

Fig. 4. Residuals vs. leverage with superimposed Cook distances of `run_diagnostic` function in `eq_Regr`.

C. Regression Diagnostics

The correlation between two treatments could be examined through linear regression using `eq_Regr`. Despite not being an equivalence assessment methodology, linear regression in the presence of correlation might expose a relationship between measurements.

```
from equiv_med.EQU import eq_Regr
regr=eq_Regr.Regr_diagn(var1 ,var2)
regr.run_diagnostic([0.05 ,0.1 ,0.2])
```

Among the plots produced by `run_diagnostic`, there is a visual determination of Cook's distances over the residuals versus leverage graph. Users input a list of possible distances plotted as dashed orange lines: points that overcome a certain distance might be considered "highly influential" on regression outcomes. This investigation of scattered values far from zero on the y-axis and with high leverage provides an understanding of which points have a high impact on the linear model. In addition, if the graph returns stable residuals as a function of leverage, it might be perceived as an indicator of homoscedasticity. In automatic, the `run_diagnostic` method also apprises the user if the residuals are normally distributed by performing Jarque-Bera statistics: this test evaluates skewness and kurtosis to classify gaussianity or not. Another prerequisite is the independence of the errors, and the function displays the Durbin-Watson test result. Further method users can apply to study influential values is `influential_points` which shows DIFITS and DFBETAS in relation to empirical thresholds.

D. Acceptance Limits based on Inherent Imprecision

A method measuring values on a continuous scale might be characterized by a certain degree of variability in quantifying an underlying unknown true amount. The imprecision in defining the ground truth is associated with the "random error," usually measured by the coefficient of variation (i.e., CV). The CV may catch uncertainty in the repeatability or reproducibility of results. In such situations, estimating the random error (and consequently the imprecision) could be possible with the `ICI_bounds` class's method `run_ICI`:

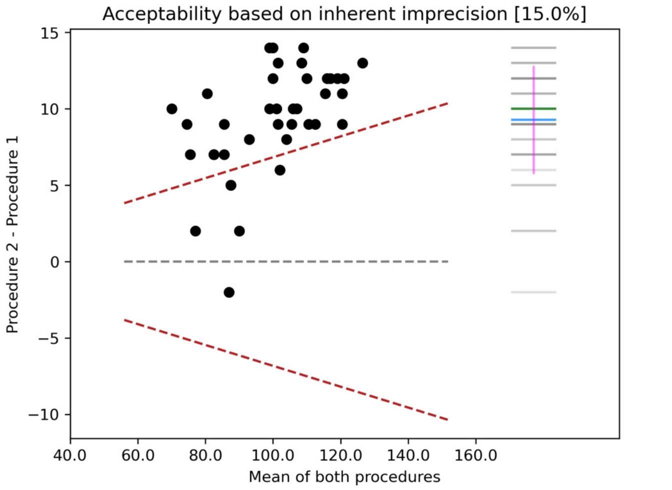

Fig. 5. Plot comparing imprecision of two measurement methods by `eq_ICI` function.

```
from equiv_med.EQU import eq_ICI
ici=eq_ICI.ICI_bounds(var1 ,var2)
ici.run_ICI(2,4)
```

The python code produces the graph in Fig. 5: the number of observations by both methods has been reduced to 40 for a less crowded and more readable plot. The mean of both measurements is shown over the x-axis, while on the y-axis the difference between them. The two dashed lines in red are the acceptance limits computed as $bias \pm z \times CV_{mean}$, with $bias = 0$ and $z = 1.96$. The $CV_{mean} = \sqrt{CV^2_{procedure1} + CV^2_{procedure2}}$ requires knowledge of the CVs related to each procedure or instrument. It is a number available after carrying out laboratory experiments or from previous literature. In the example, it has been set $CV_{procedure1} = 2$ and $CV_{procedure2} = 4$. The python routine checks if the differences between measurements are normally distributed, and if not, adjusts the σ value according to the Chebyshev interval $1 - \frac{1}{z^2}$. If the Chebyshev adjustment is performed, the limits of agreement are in red otherwise, if normality is found the limits are in orange. The title also reports the number of inliers inside bounds as a percentage. On the right side of Fig. 5, users can observe the frequency distribution of the differences between measurements as grey horizontal lines. Further lines detail the median (in green), and the mean (in blue) plotted together with one standard deviation extension from the mean difference as a vertical bar (in violet). The function also produces textual outputs that explain the operations executed on the data in detail.

E. Standardized Mean Difference and Indexes of Non-overlapping

The standardized mean difference (aka Cohen's d) exposes effect size about two normally distributed measurements. Cohen d computation undertakes different formulas depending

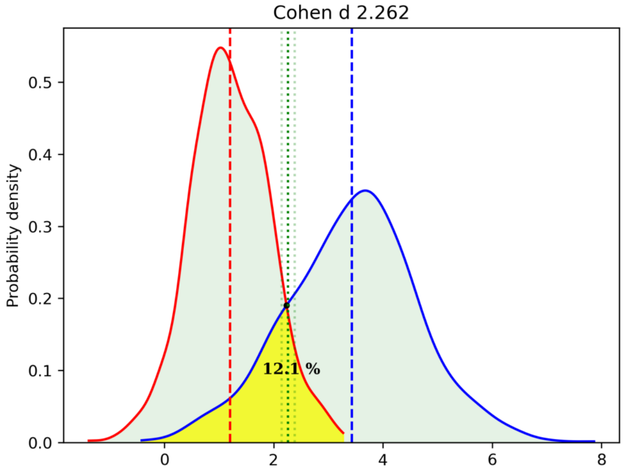

Fig. 6. Overlapping of two measurement methods by `plotting` function of `Cohen_es`. The Cohen's d CI are plotted as green dotted vertical lines.

on input values, whether they are paired or not, and whether variances are homogeneous. In two groups designs with equal variances, it is $\frac{method_1 - method_2}{sd_{pooled}}$, and it summarizes the number of standard deviations the means of two groups differ. However, when the sample size is small, Cohen d is corrected to obtain an unbiased version, called Hedges g. During the initialization of the class `Cohen_family`, the user can decide the experimental design of the two input variables (paired or not, default is not). The function `Cohen_es` automatically checks the prerequisites and selects the proper formulation, possibly applying the small sample size correction factor using the Gamma function [12].

```
from equiv_med.ES import Cohen_family
D_meas=Cohen_family.Cohen_es(var1 ,var2)
D_meas.nonoverlap_measures()
D_meas.plotting()
```

The literature on Cohen's d computation in case of unequal variances reports different approaches for determining the right effect size. In the default implementation with equality of variances, computations follow [12], with results in the case of heteroskedasticity personalizable using formulas suggested by [13], [14], [15]. Among the methods of class `Cohen_family`, the `nonoverlap_measures` returns three indexes upon verifying that input data are normally distributed and with equal variability:

- Cohen's U_3, also called "percentile standing." Indeed, effect size might be interpreted as the average percentile standing of the average experimental measurement relative to the average control.
- Cohen's U_2, as U_3 is quantified using the cumulative density function, but of $d/2$ rather than d as in U_3
- Cohen's $U_1 = \frac{2 \times U_2}{U_2}$ is the non-overlap percentage between the measurement areas subtended by the probability density functions.

These indexes complete the information supplied by d. With `plotting`, a visual interpretation of the data is shown as in Fig. 6. The two distributions' overlapping area is colored in yellow, with the overlap percentage added near the intersection point. Vertical dashed lines are the means of the two groups, while vertical dotted lines are the Cohen's d and its CI estimated via the "non-centrality" method.

Another class, called `Retesting`, calculates the *minimal detectable change* for repeated measurements using the same instrument on different occasions [16]. It tests the consistency of the scores by automatically switching between Pearson product-moment or Spearman correlation depending on the gaussianity of the data. This peculiar function has been inserted in the same folder as the Cohen's d, because there is a relation between d and the coefficient of correlation $r = \frac{d}{\sqrt{d^2+a}}$, with $a = 4$ if the two input measurements have same length. So theoretically, the correlation could be inferred from d, although the current implementation calculates the correlation indexes directly. The class `Responsiveness` contains other metrics to characterize devices' properties in case of repeated measurements recorded from two instruments over time.

F. Equivalence, Non-inferiority and Superiority

Analytical biosimilarity testing by TOST (`EIS` folder) executes two one-sided t-tests to verify the positioning of the mean difference between measurements in relation to the regulatory boundaries $\pm\delta$. In statistical terms the equivalence tests implemented in the python library aim to check:

$$H_0 : \overline{m_1} - \overline{m_2} \leq -\delta \quad or \quad \overline{m_1} - \overline{m_2} \geq \delta$$

$$H_1 : -\delta < \overline{m_1} - \overline{m_2} < \delta$$

In situations where measurements are independent, the function `run_Tost_indep` could be employed, while for paired biomarkers `run_Tost_dep`. When biomarkers do not have equal variances in these two functions, degrees of freedom are computed by the Satterthwaite formula, and the t-test is replaced by Welch's t-test. Assumption of normally distributed input data is checked automatically by Shapiro-Wilks statistic. The python package also contains two equivalence tests that tackle the biosimilarity problem from a slightly different point of view. In `Tost_Alt` class, the method `run_TOST_T` performs the similarity procedure of [17], fixing the margin at $\delta = f \times \sigma_2$ where f is a multiplication factor, and σ_2 is the standard deviation of the reference method. The f multiplier should be selected to accommodate statistical power based on sample size. The authors of [17] suggested $f = 1.5$. Inside the same `Tost_Alt` class, the method `run_TOST_MW` provides the methodology studied in [18] to control the type I errors. In standard TOST, the type I error rate is restrained by $\alpha = 0.05$, while the authors introduced a modified Wald test to assess the standard error by the maximum likelihood estimator. This operation should better control type I errors in repeated measurements involving samples of small size. The section dedicated to bioequivalence contains two additional procedures: paired

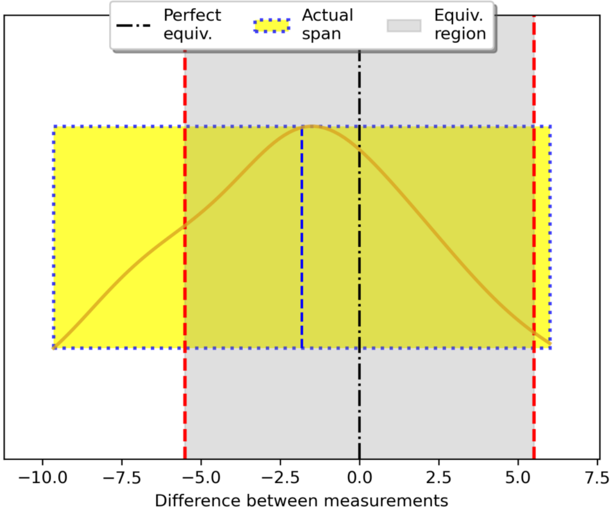

Fig. 7. Plot showing equivalence region and the actual CI extension of the difference between biomarkers, using `Tost_paired`.

measurements could be investigated with the methodology of [19] (class `Tost_paired`, Fig. 7), while in the case of heteroscedasticity, the TOST methods in `WS_eq` perform the Welch-Satterthwaite as suggested in [20]. These classes also include specialized function calls to calculate the minimal desired sample size and related statistical power. In Fig. 7, the red vertical dashed lines are the user-defined equivalence margins (set to ±5.5). As remarked in the "Introduction," this value should be deducted from literature or clinical experience. The yellow rectangle represents the extent of the input data, and it has a length that overcomes the limits, thus not allowing to establish equivalence between the datasets. Inside the yellow rectangle, the probability distribution is shown together with a vertical dashed blue line depicting the mean difference of biomarker values.

```
from equiv_med.EIS import Standard_Tost as ST
from equiv_med.EIS import Tost_NCP,Tost_WS
from equiv_med.EIS import Tost_Alt
res1=ST.EQU(var1 ,var2 ,−5.5 ,5.5)
res1.run_Tost_indep()
res1.run_Tost_dep()
res2=Tost_Alt.TOST_T(var1 ,var2)
res2.run_TOST_T()
res2.run_TOST_MW()
res3=Tost_NCP.Tost_paired(var1 ,var2 ,5.5)
res3.run_tost()
res4=Tost_WS.WS_eq(var1 ,var2 ,5.5)
res4.run_TOST()
```

As a final remark, equivalence testing could be integrated into standard inferential statistics analysis pipelines. Indeed, equivalence assessment can clarify null-hypothesis significance tests. When $p > \alpha$, biomakers are classified as "not different" in traditional inference, but only equivalence determines their exchangeability. Conversely, it might be possible to find a significant difference ($p < \alpha$) during traditional inference and, at the same time, equivalence inside certain boundaries.

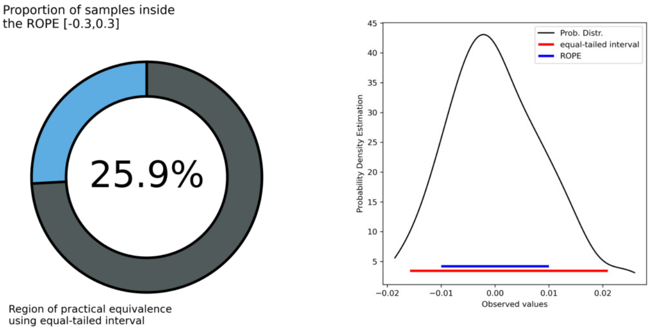

Fig. 8. Region of practical equivalence as shown by `plot_rope` of class `ROPE`. The right plot is visualized only if the sample size is larger than 50.

Non-inferiority analysis establishes that the efficacy of a new therapy is not lower than δ units than the current one:

$$H_0 : \overline{m_1} - \overline{m_2} \leq -\delta$$

$$H_1 : \overline{m_1} - \overline{m_2} > -\delta$$

Superiority evaluates if there is a difference between measurements by usually fixing $\delta = 0$:

$$H_0 : \overline{m_1} - \overline{m_2} \geq \delta$$

$$H_1 : \overline{m_1} - \overline{m_2} < \delta$$

Non-inferiority (`non_inferiority`) and superiority (`superiority`) tests were implemented following [21] as methods of the class `IoS`.

```
from equiv_med.EIS import Inf_or_Sup as IS
res5=IS.IoS(var1 ,var2)
res5.non_inferiority(ni_bound=0.1)
res5.superiority(sup_bound=0)
```

The region of practical equivalence (i.e., ROPE, developed inside `ROPE` class) has been introduced in [22] as a Bayesian probabilistic framework that does not rely on statistical significance. This approach considers the data sample as a probabilistic representation of the underlying real population of values. The user selects the region of practical equivalence, and it corresponds to the "null" hypothesis. The functions check the percentage of samples that fall inside the ROPE; this proportion is called the credibility interval, and it could be built using the highest density principle (`rope_hdi`) or as an equal-tailed interval (`rope_calc`). The intervals obtained from these two processes are the same when dealing with symmetric distributions, but skewed distributions might show different extensions; the python functions warn the user about this possibility. The credibility interval conceptualizes the idea that points comprehended inside it are more credible representatives of the data than external points, and it could be set to 95% or restricted to 89%. When the percentage of credibility interval within the ROPE is sufficiently low, the ROPE "null" hypothesis is rejected. Users can visualize the test result using `plot_rope`, as shown in Fig. 8.

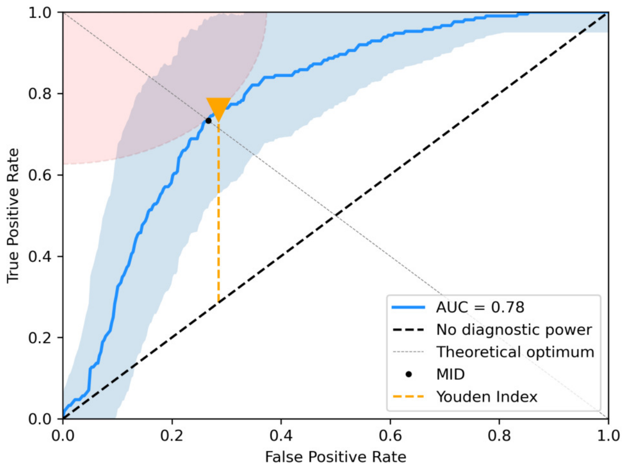

Fig. 9. ROC with CI, Youden index and K-index as shown by `plot_roc_youden`.

G. ROC Analysis

Diagnostic procedures can be evaluated in terms of ROC curves without proving any distributional assumption. Using the function `plot_roc_youden` of the class `Roc_youden`, users can visualize a ROC curve as in Fig. 9. The curve displays different threshold values determining the positivity to a medical test and the trade-off between true positive and false positive fractions. The more a ROC curve is close to the upper-left corner of the box enclosing the plot, the better the test's diagnostic power. The function automatically checks the side of the curve with the Mann-Whitney U rank test and consequently adapts computations. In the following examples, the results of a medical test are the first input variable (`test_res`) while `true_labels` represent the disease status of the subjects:

```
from equiv_med.ROC import Roc_youden as Ry
roc=Ry.Youden_Roc(test_res,true_labels)
roc.plot_roc_youden()
```

In ROC curves, it is fundamental to represent also the CIs: for their computation reckoned by non-parametric technique, `plot_roc_youden` exploits the algorithm in [23]. In addition, the plot contains the area under the curve (i.e., AUC), an accuracy metric often applied to compare different methodologies, while the Gini index is shown as textual output on the console. The orange triangle marks the Youden index, the point of maximal effectiveness of a medical test. At the Youden level, the sensitivity and specificity are balanced, thus highlighting the optimum cut-off point for the procedure. The Youden point might be close to the theoretical optimum, as happens in Fig. 9. Another index included in the graph is the K-index, represented as the pale red circle sector centered on the upper-left corner. It is the distance between the best result (the upper-left corner) and the optimum identified by the Youden index. The smaller this quarter-circle is, and better the medical test. Moreover, any ROC point that enters the K-index space is preferable to the threshold selected to determine the positives of the test. The black point "MID" is the *minimal important difference* detected by the anchor method. The MID value might also be called "minimally important change," even if it has been suggested to refer to MID only for between-subjects differences [24]. The MID value could help examine thresholds of treatments relating them to clinical improvements in health patient status.

The python package also offers four classes to statistically compare the ROC curves of two biomarkers, acquired as independent or repeated experiments. These statistics require two ROCs having the same direction:

- *DeLong* techniques in fast version [25], accomplished relating the Heaviside function to the samples mid-ranks
  - `DeLong_dependent`
  - `DeLong_independent`
- *Venkatraman* methods, performing pointwise comparison [26], [27]. This procedure requires the exchangeability assumption.
  - `Venkatraman_dependent`
  - `Venkatraman_independent`

During statistical equivalence, the ROC direction is detected by Mann-Whitney U. Intriguingly, there is a direct relation between ROC's AUC and Mann-Whitney U being $AUC = \frac{U}{n_0 \times n_1}$, with n_0 and n_1 number of negative and positive cases.

The class `Ranking_plots` allows users to visualize the precision vs. recall plot, including the calculation of area under this curve, and the true positive vs. false positive rates graph, containing the Kolmogorov-Smirnov statistic in the standard and truncated forms. Both provide an interpretation of the distance between true positive vs. false positive rate lines, but the truncated formula is less sensitive to noise.

H. The 2x2 Frequency Table and Performance Indexes

Dichotomous outcomes of diagnostic tests are generally summarized by a two-by-two frequency table, also known as two-class confusion matrix (displayed calling `frequency_plot`). Multiple performance indexes could be derived from the frequency table: these attributes are computed on a single threshold rather than several thresholds like for ROC. The python library contains the class `Frequency_table`, which calculates 34 indexes. These performance indexes support the validation of a new test against a gold standard. The `Radar` class offers comparisons of performance indexes from one or two instruments in two types of charts: a radar plot with or without overlapping input indexes or a circular paired bar graph.

```
from equiv_med.ROC import Frequency_table as ft
from equiv_med.ROC import Radars
t1=ft.Freq_table(test_res1,true_labels)
out1=t1.performance_indexes()
t2=ft.Freq_table(test_res2,true_labels)
out2=t2.performance_indexes()
rd=Radars.Radar_plots(indexes_list)
rd.radar_plots(out1,out2,overlapping=True)
```

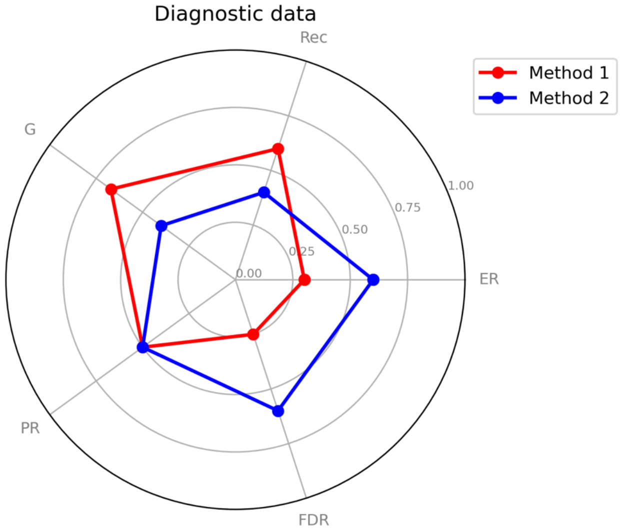

Fig. 10. Radar graphs comparing the indexes of two diagnostic tests using `radar_plots` of `Radars` class.

In Fig. 10, the radar illustrates five indexes pre-selected by the user passing the list of names during `Radar_plots` initialization. The function automatically abbreviates indexes names; in the example of Fig. 10, “Error Rate” is “ER,” “Recall” as “Rec,” “G” is “G measure,” “Prevalence” as “PR,” and “False Discovery Rate” is “FDR.”

III. Conclusions

A python library for visual understanding of medical-related statistical tests targeting several aspects of bioequivalence has been presented. It offers a free alternative to commercial software. Functions are highly automated and produce enhanced graphs to facilitate the interpretation of the output parameters. Minimal working examples were included to aid in reproducing the results. Future versions will expand and improve the implemented methodologies maintaining the spotlight on producing visual insights.

Appendix

The source code of the functions described in the document (current version 0.11) has been uploaded to GitHub (https://github.com/m89p067/equiv_med) and archived on Zenodo (https://zenodo.org/record/6504217). Installation of the package directly from GitHub through pip.

References

[1] S. C. Gad, *Safety evaluation of pharmaceuticals and medical devices: international regulatory guidelines*. Springer Science & Business Media, 2010.

[2] F. Home, “Orange book: approved drug products with therapeutic equivalence evaluations,” *US Food Drug Adm*, 2013.

[3] S.-C. Chow and S. J. Lee, “Current issues in analytical similarity assessment,” *Statistics in Biopharmaceutical Research*, vol. 13, no. 2, pp. 203–209, 2021. doi: 10.1080/19466315.2020.1801497

[4] A. Munk, J. Gene Hwang, and L. D. Brown, “Testing average equivalencefinding a compromise between theory and practice,” *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, vol. 42, no. 5, pp. 531–551, 2000.

[5] J. A. Hanley and B. J. McNeil, “The meaning and use of the area under a receiver operating characteristic (roc) curve.” *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[6] J. M. Bland and D. Altman, “Statistical methods for assessing agreement between two methods of clinical measurement,” *The lancet*, vol. 327, no. 8476, pp. 307–310, 1986.

[7] G. Cumming, “The new statistics: Why and how,” *Psychological science*, vol. 25, no. 1, pp. 7–29, 2014.

[8] C. O’Carroll, B. Rentier *et al.*, “*Evaluation of research careers fully acknowledging open science practices-rewards, incentives and/or recognition for researchers practicing open science*,” Publication Office of the Europen Union, Tech. Rep., 2017.

[9] J. M. Bland and D. G. Altman, “Applying the right statistics: analyses of measurement studies,” *Ultrasound in Obstetrics and Gynecology: The Official Journal of the International Society of Ultrasound in Obstetrics and Gynecology*, vol. 22, no. 1, pp. 85–93, 2003.

[10] S.-L. Jan and G. Shieh, “The bland-altman range of agreement: Exact interval procedure and sample size determination,” *Computers in Biology and Medicine*, vol. 100, pp. 247–252, 2018. doi: https://doi.org/10.1016/j.compbiomed.2018.06.020

[11] G. Cumming, *Understanding the new statistics: Effect sizes, confidence intervals, and meta-analysis*. Routledge, 2013.

[12] J.-C. Goulet-Pelletier and D. Cousineau, “A review of effect sizes and their confidence intervals, part i: The cohensd family,” *The Quantitative Methods for Psychology*, vol. 14, no. 4, pp. 242–265, 2018.

[13] G. Shieh, “Confidence intervals and sample size calculations for the standardized mean difference effect size between two normal populations under heteroscedasticity,” *Behavior research methods*, vol. 45, no. 4, pp. 955–967, 2013.

[14] M. Delacre, D. Lakens, C. Ley, L. Liu, and C. Leys, “Why hedges g*s based on the non-pooled standard deviation should be reported with welchs t-test,” May 2021. [Online]. Available: psyarxiv.com/tu6mp

[15] D. Cousineau, “Approximating the distribution of cohens dp in within-subject designs,” *Quant. Methods Psychol*, vol. 16, pp. 418–421, 2020.

[16] Y. Shou, M. Sellbom, and H.-F. Chen, “Fundamentals of measurement in clinical psychology,” in *Reference Module in Neuroscience and Biobehavioral Psychology*. Elsevier, 2021. ISBN 978-0-12-809324-5

[17] Y. Tsong, X. Dong, and M. Shen, “Development of statistical methods for analytical similarity assessment,” *Journal of biopharmaceutical statistics*, vol. 27, no. 2, pp. 197–205, 2017.

[18] Y.-T. Weng, Y. Tsong, M. Shen, and C. Wang, “Improved wald test for equivalence assessment of analytical biosimilarity,” *International Journal of Clinical Biostatistics and Biometrics*, vol. 4, no. 1, pp. 1–10, 2018.

[19] G. Shieh, “Assessing agreement between two methods of quantitative measurements: Exact test procedure and sample size calculation,” *Statistics in Biopharmaceutical Research*, vol. 12, no. 3, pp. 352–359, 2020. doi: 10.1080/19466315.2019.1677495

[20] G. Shieh, S.-L. Jan, and C.-S. Leu, “Exact properties of some heteroscedastic tost alternatives for bioequivalence,” *Statistics in Biopharmaceutical Research*, pp. 1–10, 2021.

[21] S. Wellek, *Testing statistical hypotheses of equivalence*. Chapman and Hall/CRC, 2002.

[22] J. K. Kruschke and T. M. Liddell, “The bayesian new statistics: Hypothesis testing, estimation, meta-analysis, and power analysis from a bayesian perspective,” *Psychonomic bulletin & review*, vol. 25, no. 1, pp. 178–206, 2018.

[23] P. Martínez-Camblor, S. Pérez-Fernández, and N. Corral, “Efficient nonparametric confidence bands for receiver operating-characteristic curves,” *Statistical Methods in Medical Research*, vol. 27, no. 6, pp. 1892–1908, 2018.

[24] H. C. De Vet, R. W. Ostelo, Terwee *et al.*, “Minimally important change determined by a visual method integrating an anchor-based and a distribution-based approach,” *Quality of life research*, vol. 16, no. 1, pp. 131–142, 2007.

[25] X. Sun and W. Xu, “Fast implementation of delongs algorithm for comparing the areas under correlated receiver operating characteristic curves,” *IEEE Signal Processing Letters*, vol. 21, no. 11, pp. 1389–1393, 2014.

[26] E. S. Venkatraman and C. B. Begg, “A distribution-free procedure for comparing receiver operating characteristic curves from a paired experiment,” *Biometrika*, vol. 83, no. 4, pp. 835–848, 1996.

[27] E. Venkatraman, “A permutation test to compare receiver operating characteristic curves,” *Biometrics*, vol. 56, no. 4, pp. 1134–1138, 2000.