Case Study of Genetic Algorithms in Metrology: Assessment of Inter-laboratory Comparisons

Romain Coulon [1*]

[1]Bureau International des Poids et Mesures, Pavillon de Breteuil, Sèvres, Cedex, F-92312, France

Abstract. This study reviews conventional consensus estimation methods, including mean-based, median-based, and pooling-based approaches, and evaluates their performance under challenging scenarios involving outliers and deviations from normality. While traditional methods such as the weighted mean and weighted median often fail to handle extreme values and non-Gaussian distributions, advanced techniques like the Monte Carlo Median (MCM) and Power Moderated Mean (PMM) offer improved robustness. The Genetic Algorithm (GA), a novel optimization-based approach, demonstrates exceptional resilience to outliers. To facilitate its application, the GA is made available through the Python package consensusGen, accessible via the Python Package Index and GitHub. This ensures that practitioners and researchers can easily implement the GA in their consensus estimation tasks, benefiting from its superior robustness and precision.

1 Introduction

In an inter-laboratory comparison, a set of measurements is represented by $X = (x_1, x_2, \dots, x_i, \dots, x_N)^{\mathrm{T}}$ with corresponding standard uncertainties $U = (u_1, u_2, \dots, u_i, \dots, u_N)^{\mathrm{T}}$. To determine a reference value μ that reflects the consensus of the dataset, a common approach involves aggregating the measurements using the Weighted Mean (WM), where the weights are defined as $w_i^{(\mathrm{WM})} = u_i^{-2}$. The reference value and its standard uncertainty are then calculated as follows:

$$\hat{\mu}^{(\mathrm{WM})} = \frac{\sum_{i=1}^{N} w_i^{(\mathrm{WM})} x_i}{\sum_{i=1}^{N} w_i^{(\mathrm{WM})}}, \tag{1}$$

$$u\left(\hat{\mu}^{(\mathrm{WM})}\right) = \left(\sum_{i=1}^{N} w_i^{(\mathrm{WM})}\right)^{-1/2}. \tag{2}$$

The standard Weighted Mean (WM) estimator is known to be sensitive to extreme values in a dataset, making it non-robust in such scenarios. A more robust consensus value can be derived by modifying the weight parameters to include a between-laboratory variance τ^2, often referred to as "dark uncertainty" [1]. In this approach, the weights are redefined as $w_i = 1/(u_i^2 + \tau^2)$. Estimation of τ^2 is typically achieved using optimization procedures, such as those based on the Birge ratio [2–6]. These procedures adjust τ to satisfy the following condition:

$$\frac{1}{N-1}\sum_{i=1}^{N} \frac{\left(x_i - \hat{\mu}^{(\mathrm{WM})}\right)^2}{u_i^2 + \tau^2} = 1. \tag{3}$$

The consensus value remains unchanged in this approach, but its uncertainty, $u\left(\hat{\mu}^{(\mathrm{WM}^*)}\right)$, is adjusted to ensure consistency with the dataset based on the Chi-squared test.

Another strategy, denoted as WM**, involves implementing a null hypothesis test to identify and reject potential outliers [7]. This approach is applied on the WM estimator when the Chi-squared test fails under the following condition:

$$P\left(\chi^2_{N-1} > \sum_{i=1}^{N} \frac{\left(x_i - \hat{\mu}^{(\mathrm{WM})}\right)^2}{u_i^2}\right) < 0.05, \tag{4}$$

where χ^2_{N-1} is the Chi-squared statistics with $N-1$ degrees of freedom.

To detect discrepant measurements, one of the following criteria can be used:

$$\left|x_i - \hat{\mu}^{(\mathrm{WM})}\right| > 2\sqrt{(1 - 2w_i)u^2(x_i) + u^2(\hat{\mu}^{(\mathrm{WM})})}, \tag{5}$$

or

$$\left|x_i - \hat{\mu}^{(\mathrm{WM})}\right| > 2\sqrt{u^2(x_i) + u^2\left(\hat{\mu}_{-i}^{(\mathrm{WM})}\right)}, \tag{6}$$

where $u^2\left(\hat{\mu}_{-i}^{(\mathrm{WM})}\right)$ is the variance of the consensus value computed excluding measurement i from the dataset.

The WM** procedure ensures a minimum uncertainty value $u\left(\hat{\mu}^{(\mathrm{WM}^{**})}\right)$ by rejecting outliers. However, WM* is less precise but not destructive, as no measurements are excluded. A hybrid approach combining $WM*$ and $WM**$ is also sometimes employed and called the power-moderated mean (PMM) [8]. This approach is actively employed for key comparisons piloted by the section 2 of the Consultative Committee for Ionizing Radiation (CCRI) of the International Committee for Weights and Measures [9].

* Corresponding author: author@email.org

$\hat{\mu}^{(\mathrm{WM}*)}, \hat{\mu}^{(\mathrm{WM}**)}$and $\hat{\mu}^{(\mathrm{PMM})}$ estimators rely on the Chi-squared statistic, which describes the distribution of the sum of squared standard normal random variables. As a result, they assume that the measurements follow a normal distribution, though real-world measurement statistics may deviate from this assumption [10,11].

As an alternative to mean-based estimators, the median offers inherent robustness against extreme values. To address the specific challenges of measurement dataset (X, U), the weighted median (WMD) has been proposed by J. W. Muller [12], incorporating weights $w_i^{(\mathrm{WMD})} = u_i^{-2}$. The weighted median is defined as:

$$\hat{\mu}^{(\mathrm{WMD})} = \underset{h}{\operatorname{argmin}} \left(\sum_{i=1}^{N} w_i^{(\mathrm{WMD})} |x_i - h| \right), \quad (7)$$

and its associated uncertainty is calculated as:

$$u\left(\hat{\mu}^{(\mathrm{WMD})}\right) = \frac{1.9}{\sqrt{N-1}}$$

$$\underset{h}{\operatorname{argmin}} \left(\sum_{i=1}^{N} w_i^{(\mathrm{WMD})} \left| \left| x_i - \hat{\mu}^{(\mathrm{WMD})} \right| - h \right| \right). \quad (8)$$

Another approach to estimating the median uses Monte Carlo sampling, treating the measurement as random variables drawn from a normal distribution, $X^{(r)} \sim \mathrm{NORMAL}(X, U^2)$. The Monte-Carlo median (MCM) is then computed as the mean of medians from each resampled dataset $X^{(r)} = \left(x_1^{(r)}, x_2^{(r)}, \ldots, x_i^{(r)}, \ldots, x_N^{(r)}\right)^{\mathrm{T}}$. The Monte Carlo median estimator [7] is given by:

$$\hat{\mu}^{(\mathrm{MCM})} = \frac{1}{M} \sum_{r=1}^{M} \mathrm{median}\left(X^{(r)}\right), \quad (9)$$

and its variance is estimated as:

$$u^2\left(\hat{\mu}^{(\mathrm{MCM})}\right) = \frac{1}{M-1} \sum_{r=1}^{M} \left(\mathrm{median}\left(X^{(r)}\right) - \hat{\mu}^{(\mathrm{MCM})}\right)^2. \quad (10)$$

This Monte Carlo sampling approach forms the foundation of the genetic algorithm originally introduced in [13] and described in the next section. By leveraging the bootstrapping technique, the genetic algorithm refines the estimation process, optimizing robustness and accuracy when determining consensus values.

2 Method

2.1 Preprocessing of the resampled measurement data

In this new approach, the resampled data $X^{(r)}$ are pooled into a single vector:

$$Y = \left(x_1^{(1)}, \ldots, x_N^{(1)}, \ldots, x_i^{(r)}, \ldots, x_N^{(M)}\right)^{\mathrm{T}} = \left(y_1, \ldots, y_j, \ldots, y_Z\right)^{\mathrm{T}}, \quad (11)$$

where $Z = N \times M$. This technique, known as the *linear pooling procedure* (LP), was first formalized by M. Stone [14] and may trace its origins back to Pierre-Simon de Laplace [15]. LP serves as a foundation for generating the initial population in the genetic algorithm (GA). While LP considers only the resampled values $y_j = x_i^{(r)}$, the GA extends this approach by incorporating the provenance of each data point within the pooled dataset. This is achieved through a vector of labels $L = (\mathrm{A}, \mathrm{B}, \mathrm{C}, \ldots)$, representing the N participants (e.g., "A", "B", "C"). These labels are stored in a $1 \times Z$ genome vector $G = \left(g_1, \ldots, g_j, \ldots, g_Z\right)^{\mathrm{T}}$, where the labels g_j for each y_j is given by $g_j = L_{j \bmod N+1}$.

Each individual in the dataset is therefore described by its value $y_j = x_i^{(r)}$ (its *phenotype*) and its corresponding label g_j (its *genome*). This pairing, formally represented as $\left(y_j, g_j\right)$, creates a structured mapping for every data point $j \in \{1, .., Z\}$, where $Z = N \times M$. This structured representation enables the GA to evaluate consensus values while preserving information about the data's origin.

First, values in Y and G are sorted by ascending order of Y to produce $Y^{(0)}$ and $G^{(0)}$, where

$$y_j^{(0)} < y_{j+1}^{(0)}, \forall j \in \{1, .., Z-1\}. \quad (12)$$

The sorted dataset $\left\{Y^{(0)}, G^{(0)}\right\}$ serves as the initial population for the genetic algorithm (GA).

A GA belongs to as a specific class of optimization algorithms inspired by evolution biology [16,17]. It iteratively applies evolutionary processes to a population of individuals, each characterized by their *phenotype* (e.g., $y_j^{(0)}$) and *genome* (e.g., $g_j^{(0)}$). At each generation t, a new population $t+1$ is created from the current one through selection mechanisms based on individual fitness.

The selection process follows Darwinian principles, often summarized as "survival of the fitness". The fitness of each individual is evaluated using a *fitness function*, which reflects how well the individual satisfies the problem's objectives. This function typically incorporates information from the genome and drives the evolution toward an optimal solution.

In this approach, the objective is to derive a consensual reference value from the pooled and bootstrapped data $Y^{(0)}$, using the corresponding genomes $G^{(0)}$. To achieve this, the genetic algorithm (GA) discourages inbreeding while promoting crossover within the population $\left\{Y^{(t)}, G^{(t)}\right\}$. Fitness function F_j is designed to penalize genetic similarity, fostering diversity between individuals in terms of their genomes.

2.2 Fitness Function Based on Cosine Similarity

For each individual $j = \{1, \ldots, Z^t - 1\}$ at generation t, the fitness function F_j is given by:

$$F_j = 1 - \mathrm{cosine}\left(g_j^{(t)}, g_{j+1}^{(t)}\right), \quad (13)$$

where $\mathrm{cosine}\left(g_j^{(t)}, g_{j+1}^{(t)}\right)$ measures the similarity of genomes between adjacent individuals based on frequency vectors $v(g)$. Each frequency vector $v(g) = \left(P_g(L_1), \ldots, P_g(L_i), \ldots, P_g(L_N)\right)^{\mathrm{T}}$ counts occurrences of each gene in L_i (participant label) in the genome g. The cosine similarity is defined as:

$$\text{cosine}\left(g_j^{(t)}, g_{j+1}^{(t)}\right) = \frac{v\left(g_j^{(t)}\right)^{\mathrm{T}} v\left(g_{j+1}^{(t)}\right)}{\sqrt{v\left(g_j^{(t)}\right)^{\mathrm{T}} v\left(g_j^{(t)}\right)} + \sqrt{v\left(g_{j+1}^{(t)}\right)^{\mathrm{T}} v\left(g_{j+1}^{(t)}\right)}}. \quad (14)$$

For example:

- If $g_j^{(t)}$ and $g_{j+1}^{(t)}$ are identical, $\text{cosine}\left(g_j^{(t)}, g_{j+1}^{(t)}\right) = 1$ (e.g., cosine(ABC, BCA) = 1), resulting in $F_j = 0$, and the individual has no chance of breading.
- If $g_j^{(t)}$ and $g_{j+1}^{(t)}$ have no genes in common, $\text{cosine}\left(g_j^{(t)}, g_{j+1}^{(t)}\right) = 0$ (e.g., cosine(ABC,EFG) = 0), resulting in $F_j = 1$, and the individual survives.
- For partial overlap (e.g., cosine(ABC, BGA) = 0.67), F_j falls between 0 and 1, with survival depending on whether F_j exceeds a user-defined threshold $S \in [0,1]$.

2.3 Breeding and Phenotype Update

If breeding is allowed ($F_j > S$), the new genome at generation $t+1$ combines genes from the parents:

$$g_j^{(t+1)} = \text{CONCATENATE}\left(g_j^{(t)}, g_{j+1}^{(t)}\right). \quad (15)$$

The phenotype $y_{j+1}^{(t)}$ is generated as a random mixture of parental phenotypes:

$$y_j^{(t+1)} = a y_j^{(t)} + (1-a) y_{j+1}^{(t)}, \quad (16)$$

where $a \sim \text{uniform}(0,1)$.

2.4 Final Population and Reference Estimation

After W iterations, the GA yields a final population $(Y^{(W)}, G^{(W)})$. The consensus value $\hat{\mu}^{(\text{GA})}$ and its uncertainty $u\left(\hat{\mu}^{(\text{GA})}\right)$ are derived as:

$$\hat{\mu}^{(\text{GA})} = \frac{1}{Z^{(W)}} \sum_{j=1}^{Z^{(W)}} y_j^{(W)} \quad (17)$$

and

$$u^2\left(\hat{\mu}^{(\text{GA})}\right) = \frac{1}{N(Z^{(W)}-1)} \frac{Z^{(W)}}{Z^{(0)}} \sum_{j=1}^{Z^{(W)}} \left(y_j^{(W)} - \hat{\mu}^{(GA)}\right)^2. \quad (18)$$

The weight for each participant is proportional to the frequency of their original genome in $G^{(W)}$:

$$w_i = \frac{P_{G^{(W)}}(L_i)}{Z^{(W)}}, \forall i \in \{1,..,N\}, \quad (19)$$

with its standard uncertainty:

$$u(w_i) = \frac{\sqrt{P_{G^{(W)}}(L_i)}}{Z^{(W)}}. \quad (20)$$

The uncertainty of the deviation for each participant, $d_i^{(\text{GA})} = x_i - \hat{\mu}^{(\text{GA})}$, is given by:

$$u^2\left(d_i^{(\text{GA})}\right) = (1-w_i)u_i^2 + u^2\left(\hat{\mu}^{(\text{GA})}\right) + \sum_{i=1}^{N} x_i^2 u^2(w_i). \quad (21)$$

Unlike conventional methods, the GA explicitly incorporates the uncertainty of participant weights, $u(w_i)$, providing a comprehensive evaluation of uncertainty associated with deviations from the reference. This robust framework ensures greater reliability in the consensus estimation while accounting for variability in individual contributions.

3 Results

To demonstrate the application of the Genetic Algorithm (GA), a dataset with seven measurements was analysed:

- Measurement values (X): $\{10.1, 11, 14, 10, 10.5, 9.8, 5.1\}$,
- Standard uncertainties (U): $\{1, 1, 1, 2, 1, 1.5, 3\}$,
- Participant labels (L): {A, B, C, D, E, F, G}.

In this dataset, point C (14) is moderately above the other values and precise, while point G (5.1) is significantly lower with higher uncertainty.

The GA was executed with the following parameters:

- Number of generations(W): 1,
- Resampled values (M): 1 000 000,
- Breeding threshold (S): 0.02.

The initial pooled dataset $Y^{(0)}$ from the linear pooling (LP) reflects the distribution of the resampled measurements and is represented by the blue distribution in Fig. 1. This distribution includes equal contributions from all participants.

After applying one generation of the GA ($Y^{(1)}$) with selection and breeding, the resultant distribution is displayed as the orange distribution in Fig. 1. The GA procedure reshapes the distribution toward a monomodal distribution inherently centred around its main mode.

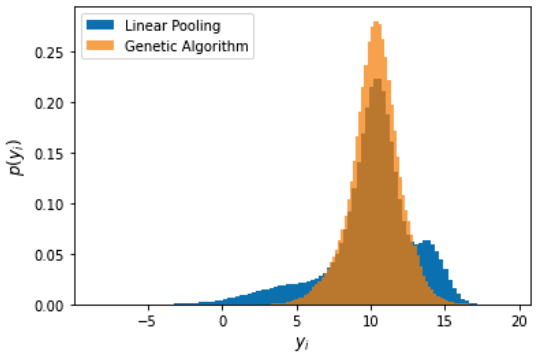

Fig. 1. Distribution of the values $\boldsymbol{Y^{(0)}}$ and $\boldsymbol{Y^{(1)}}$.

The GA-based approach demonstrates robustness against the influence of extreme points, particularly for outliers like C (14) and G (5.1) (see Fig. 2). This robustness is evident when analyzing the weights assigned to each participant in the final generation (here $W = 1$), as shown in Fig. 3.

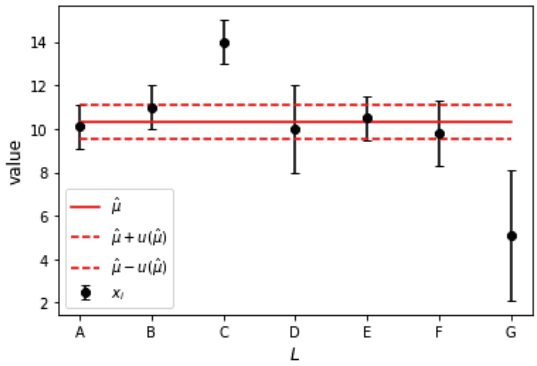

Fig. 2. The measurements and reference value in red line with standard uncertainties in dash lines.

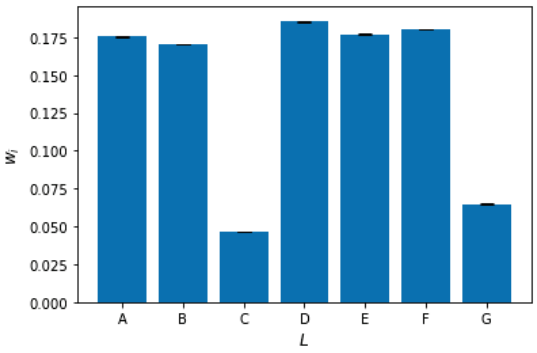

Fig. 3. Weights assigned to each measurement in the estimation of the reference value.

The GA selection process significantly reduces the population size Z with each generation, reflecting the algorithm's filtering strength. Fig. 5 illustrates this reduction for the given example, where the initial population size decreases to 62.3 % after the first generation, and only 9 % of the original population remains by the seventh generation. The threshold S allows for control over the "killing rate" between generations. A higher S value leads to stricter selection and faster population reduction, while a lower S retains more data. Whatever the tuning of S, the first selection step results in a substantial improvement of the precision and shifts the reference value toward a better consensus compared to the linear pooling (LP) method. Beyond the first step, additional selection rounds yield minimal improvement in consensus quality while significantly degrading precision due to excessive loss of data (see Fig. 4). In this example, no notable gains are observed in the reference value after the first generation, indicating that further iterations are unnecessary. Based on these observations, one or two generations of the GA is recommended for most applications. For fine-tuning, adjusting the threshold S can control the tradeoff between consensus quality and data retention.

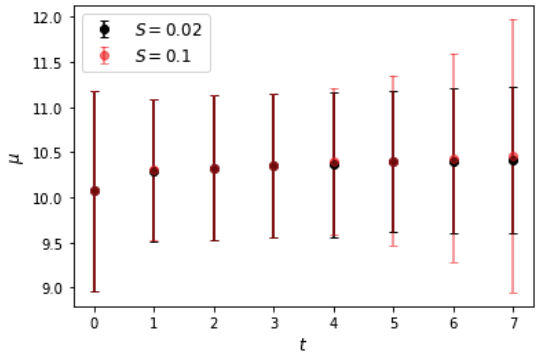

Fig. 4. Estimated reference value as a function of the evolution step.

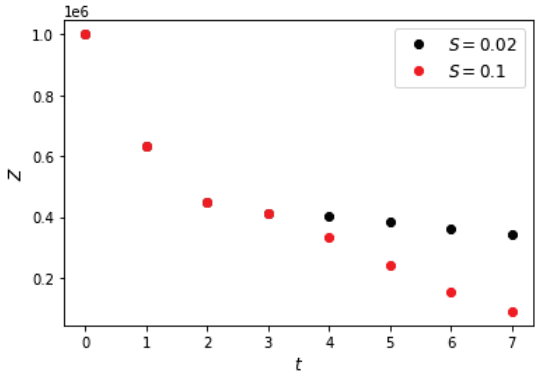

Fig. 5. Population size as a function of the evolution step.

It is important to note that this statistical approach serves as an assessment tool and does not replace the necessary scientific discussions on extreme values, which are overseen by an international committee responsible for comparison exercises.

4 Conclusion

Among all consensus estimators, the Genetic Algorithm (GA) emerged as the new reliable and versatile estimator with exceptional robustness against outliers. For practical applications, the GA is readily available as the Python package **consensusGen**. It can be accessed on the Python Package Index at https://pypi.org/project/consensusGen/ and its development repository on GitHub at https://github.com/RomainCoulon/consensusGen. This accessibility ensures that researchers and practitioners can seamlessly integrate the GA into their workflows, leveraging its robust performance for consensus estimation tasks.

References

1. M. Thompson and S. L. R. Ellison, Accreditation and Quality Assurance **16**, 483 (2011)
2. O. Bodnar and C. Elster, Metrologia **51**, 516 (2014)
3. R. DerSimonian and N. Laird, Control Clin Trials **7**, 177 (1986)
4. R. C. Paule and J. Mandel, J Res Natl Bur Stand (1934) **94**, 197 (1989)

5. A. A. Veroniki, D. Jackson, W. Viechtbauer, R. Bender, J. Bowden, G. Knapp, O. Kuss, J. P. Higgins, D. Langan, and G. Salanti, Res Synth Methods **7**, 55 (2016)
6. R. T. Birge, Physical Review **40**, 207 (1932)
7. M. G. Cox, Metrologia **39**, 589 (2002)
8. S. Pommé and J. Keightley, Metrologia **52**, S200 (2015)
9. R. Coulon, C. Michotte, S. Courte, M. Nonis, T. Ziemek, J. Marganiec-Gałązka, E. Lech, P. Saganowski, M. Czudek, and A. Listkowska, Metrologia **60**, 06001 (2023)
10. D. C. Bailey, R Soc Open Sci **4**, 160600 (2017)
11. R. Coulon, C. Michotte, and V. Gressier, in *MathMet Conference* (2022)
12. J. W. Muller, *Rapport BIPM-2000/6, Weighted Medians* (Sèvres, 2000)
13. R. M. Coulon and S. Judge, Metrologia (2021)
14. M. Stone, The Annals of Mathematical Statistics **32**, 1339 (1961)
15. M. Bacharach, J Am Stat Assoc **74**, 837 (1979)
16. D. B. Fogel, IEEE Trans Neural Netw **5**, 3 (1994)
17. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley Longman Publishing Co., Inc.75 Arlington Street, Suite 300 Boston, MAUnited States, 1989)