A machine learning approach to identify candidate reference genes based on RNA-seq data

Edian F. Franco[1,2], Dener Maués[1], Ronnie Alves[3], Luis Guimarães[1], Vasco Azevedo[4], Artur Silva[1], Preetam Ghosh[5], Jefferson Morais[2], and Rommel T. J. Ramos[1,2,*]

[1]Institute of Biological Sciences,Laboratory of Biological Engineering, Federal University of Para, Belem,Pará,Brazil
[2]Department of Computer Science,Computer Science Postgraduate Program (PPGCC), Federal University of Para, Belem,Pará,Brazil
[3]Vale Technology Institute, Belem, Para, Brazil.
[4]Institute of Biological Sciences, Federal University of Minas Gerais-UFMG,Belo Horizonte, Minas Gerais, Brazil.
[5]Department of Computer Science, Virginia Commonwealth University,Richmond, VA, USA
[*]rommelthiago@gmail.com; rommelramos@ufpa.br

ABSTRACT

Reference genes (RGs), also called Housekeeping genes (HKGs), are essential for gene expression based studies performed through Reverse Transcriptase-polymerase Chain Reaction (RT-qPCR). These genes are related with the basic cellular processes that are essential for cell maintenance, survival and function. Thus, RGs should be expressed in all cells of an organism regardless of the tissue type, cell state or cell condition. High-throughput technologies, including RNA sequencing (RNA-seq), are used to study and identify these types of genes. RNA-seq is a high-throughput method that allows the measurement of gene expression profiles in a target tissue or an isolated cell. Moreover, machine learning methods are routinely applied in different genomics related areas to enable the interpretation of large datasets, including those related to gene expression.This study reports a new machine learning based approach to identify candidate HKGs *in silico* from RNA-seq gene expression data. The approach enabled the identification of stable RG candidates in RNA-seq data from *Corynebacterium pseudotuberculosis* and *Escherichia coli* strains. These genes showed stable expression under different stress conditions as well as low variation index and fold changes. Furthermore, some of these genes were already reported in the literature as RGs or RG candidates for the same or other bacterial organisms, which reinforced the accuracy of the proposed method. We present a novel approach based on clustering algorithms and machine learning methods that can identify stable reference genes from gene expression data with high accuracy and efficiency. As a result of our approach, we present ClustREFGenes a web tool to identify reference genes candidates *in silico* through clustering techniques using RNA-seq data. The method used validated reference genes set to identify new references genes candidates based on Euclidean distance. The tool is freely available at http://computationalbiology.ufpa.br/ClustREFGenes/

Introduction

Reference Genes are constitutive genes required for the maintenance of basic cellular functions. Thus, RGs are expressed in all cells of an organism under both normal and patho-physiological conditions[1]. However, some RGs are expressed at relatively constant rates in most non-pathological situations, although in recent times different studies have reported variations in the reference genes under different experimental treatments, time variations or cell types[2–5]. Hence, the expression of RGs is used as reference point in the expression levels of other genes in analyzing gene expression datasets; the key criterion in choosing references genes is based on low variance in control and experimental conditions[6].

Nevertheless, reports have shown the importance of an accurate prediction of references genes for satisfactory normalization of gene expression data obtained by the method of reverse transcription quantitative real-time PCR (RT-qPCR). A guideline called the Minimum Information for Publication of Quantitative Real-Time PCR Experiments (MIQE) established the fact that accurate study design would require between three and five good RGs[7,8].

The gene expression data is essential to understand many biological processes in different organisms in relation to its environment. RNA-Seq is an important technique frequently used in recording gene expression data that allows for capturing an accurate picture of the molecular processes within the organisms and understanding the interaction between genes or other genomic elements[9–12]. Several gene expression data analysis methods are available to identify RGs, including machine learning (ML) methods[13], which enable the collection and identification of valid, new, potentially usable and understandable patterns and knowledge based on a dataset, which can lead to the detection and identification of possible RG candidates[14,15]. ML methods have also been applied in different genomic areas to enable the interpretation of large datasets, including those related

to gene expression[15].

Some studies related to RGs and HKGs identification have been conducted using ML approaches with conventional classifiers (e.g., neural networks and support vector machine (SVM)). Some approaches predict housekeeping genes based on Fourier transform analysis of time series gene expression data in combination with SVM; this method has been used to identify 510 HKGs in human genomic data[16]). Studies have classified RGs and HKGs based on physical and functional characteristics using exon length and chromatin compactness measurements as properties with the Naive Bayes classification algorithm to identify new HKG and RG candidates in humans[17]. These studies used eukaryotic data, which demonstrates the viability of identifying these genes in other organisms, including prokaryotes, using ML methods.

Within machine learning methods, clustering algorithms are based on unsupervised learning and are used to cluster objects based on the intrinsic information contained in the data and their relationships[18]. Clustering algorithms seek groups that are a) compact, with the members of each cluster as close as possible to one another, and b) separated, in which the clusters are as far apart as possible[19]. These methods may be applied for the identification of expression patterns of gene groups, including references genes[20].

Clustering algorithms use different methods or techniques to discover clusters. The most popular methods applied to the bioinformatics domain are the partitioned, hierarchical, density-based, and grid-based methods[18,21]. Internal metrics are used to evaluate the clustering methods and seek the set of groups that best adapt to the natural partitions[22]. These metrics evaluate cluster compactness, separation and robustness using information intrinsic to the data[23].

This study reports an unsupervised machine learning based approach to identify reference gene candidates based on RNA-seq data. The analysis is supported by intrinsic evaluation metrics of clustering to evaluate and validate our method. We employed data from expression studies of *Corynebacterium pseudotuberculosis*[24] and *Escherichia coli*[25] that were exposed to different levels of stress to test our method.

Methods

Proposed Pipeline for HKG identification

The proposed approach consists of three stages (Fig 1):

1. Pre-clustering- wherein the dataset is pre-processed and evaluated to identify optimal number of clusters using evaluation metrics (SD validity (SDbw) and Dunn index) and stability (average portion of non-overlap measure, and average distance between means measure).

2. Clustering - data clustering algorithms are applied in this stage to define genes with the same expression profiles, the similarities in the data are measured using algorithms based on the Euclidean distance calculated on the gene expression profiles.

3. Post-Clustering - Reference Gene candidates present in the dataset are identified in this stage. First, a square Euclidean n x n matrix is calculated (where n is the number of genes) based on the proximity of reference genes previously described in the literature, with the genes set under each stress condition. Then, the quartile metric of the distance matrix is set as the cut-off threshold on the distance matrices to identify the genes closest to the reference genes.

Pre-Clustering

RNA-seq data

The RNA-seq data used in this study was gathered from the study by Pinto and colleagues (2014), wherein *Corynebacterium pseudotuberculosis* strains CP1002 and CP258 were subjected to different stress levels (acid, osmotic and heat stress). Then, genes related to bacterial survival were evaluated during host infection under simulated conditions (i.e., acid, osmotic and heat stress) and a control condition using the RNA-seq method in the SOLiD System sequencing platform[24].

The RNA-seq data was processed through a reference-based approach using the CLC Genomics Workbench software and considering alignments with at least 70% identity in 80% of the read lengths using the following *Corynebacterium pseudotuberculosis* strains: CP258, isolated from a sheep (CP003540.2)[26], and CP1002, isolated from a horse (CP001809.2)[27].

The third dataset was obtained from Berghoff and colleagues, where they examined the *E.coli* response to the chemotherapeutic drug mitomycin C, which they investigated at early and late time-points by RNA-seq[25].

Core genome

For the CP258 and CP1002 genomes, the genes belonging to the core genome was obtained using the Pan-Genome Analysis Pipeline (PGAP) software[28] which enabled us to identify the set of genes shared between the *Corynebacterium pseudotuberculosis* genomes which were then designated as possible reference genes. Thus, 38 *Corynebacterium pseudotuberculosis* strains

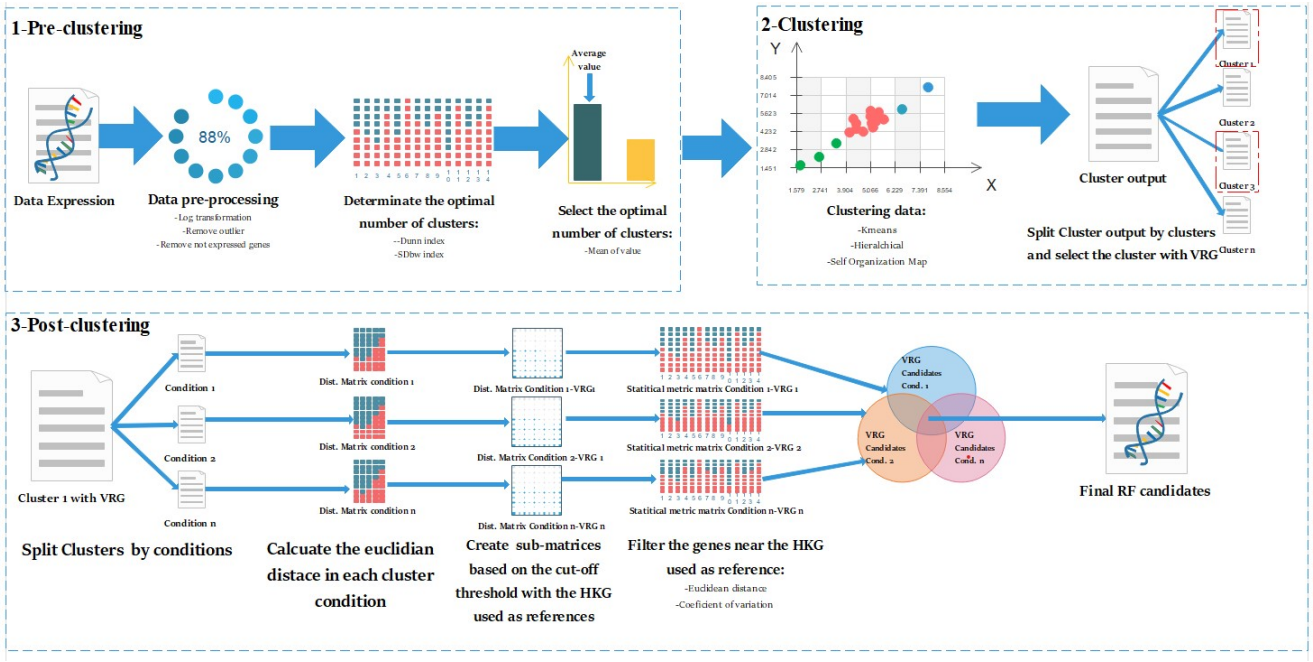

Figure 1. Flowchart of the approach used to identify candidate references gene candidates using expression data.

were used (12 from horses and 26 from sheep) with 90% coverage, 90% identity and an e-value of 1E-05 as parameters. The strains were divided into two groups according to the biovars (equi and ovis).

The genes obtained in the core genome were compared with the genes of *C. pseudotuberculosis* strains 258 and 1002 to identify the commonly expressed genes in the data gathered from the experiment by Pinto and colleagues[24]. In the *E. coli* dataset all the genes from the genome was used in this study.

Data pre-processing

The raw RNA-Seq data were normalized by reads per kilobase of exon model per million mapped reads (RPKM), which was defined according to equation 1:

$$RPKM_g = \frac{r_g.10^9}{fl_g.R} \tag{1}$$

wherein r_g is the number of reads mapped to a particular gene (g) region, and the feature length fl_g is the number of nucleotides in a mappable region of a gene, and R is defined as the total number of reads from the sequencing run of that sample, $R = \sum g \in G^{r_g}$ is the total count for the $ijth$ sample[29]. The RPKM values are transformed into $(\log_2 + 1)$, thereby reducing data asymmetry and the outlier effects on the dataset[30].

Moreover, the data-sets was processed to find and remove the extreme values and outlier expression values to improve the accuracy of the clustering algorithms following the methodology shown in[12].

Determining the optimal number of clusters

All expression datasets were evaluated using the SD validity (SDbw) index and Dunn index as defined below; these indices can individually predict the optimal number of clusters in each data set. Alongside, some other indices were also evaluated, such as, Silhoutte index and connectivity, to selected the better index to according the data-sets.

SDbw index This validation index definition is based on the criteria of compactness and separation between clusters. The index is optimized for data sets that include compact and well-separated clusters. The compactness of the data set is measured by the cluster variance whereas the separation is measured by the density between clusters[31,32]. The index is computed by the equation 2:

$$SDbw(q) = Scat(q) + Density.bw(q) \tag{2}$$

The first term, Scat(q), signifies the average compactness of q clusters (i.e., intra-cluster distance) and is computed by equation 3:

$$Scat(q) = \frac{\frac{1}{q}\sum_{k=1}^{q} \|\sigma^{(k)}\|}{\|\sigma\|} \tag{3}$$

where σ is the vector of variances for each variable in the data set (i.e., for p variables, $\sigma = (VAR(V_1), VAR(V_2), ..., VAR(V_p)))$ and $\sigma^{(k)}$ is the variance vector for each cluster C_k (i.e., $\sigma^{(k)} = (VAR(V_1^{(k)}), VAR(V_2^{(k)}), ..., VAR(V_p^{(k)}))$.

he second term $Density.bw(q)$, is the inter-cluster density. It evaluates the average density in the region among clusters in relation to the density of the clusters and it is calculated using the equation 4.

$$Density.bw(q) = \frac{1}{q(q-1)} \sum_{i=1}^{q} \left(\sum_{j=1, i\neq j}^{q} \frac{density(u_{ij})}{max(density(c_i), density(c_j))} \right), \tag{4}$$

where u_{ij} is the middle point of the line segment defined by the clusters' centroids c_i and c_j and the $density(u_{ij})$ is calculated using the equation 5,

$$density(u_{ij}) = \sum_{l=1}^{n_{ij}} f(x_l, u_{ij}), \tag{5}$$

where n_{ij} is the number of tuples that belong to the cluster C_i and C_j, $f(x_l, u_{ij})$ is equal to 0 if $d(x, u_{ij}) >$ Stdev (Standard deviation) and 1 otherwise. Stdev is the average standard deviation of the clusters.

Dunn index The Dunn index was used to identify compact clusters with good separation. The index is defined as[33]:

$$\min_{1\leq i\leq c} \left\{ \min_{\substack{1\leq j\leq c \\ j\neq i}} \left\{ \frac{s(X_i, X_j)}{\max\limits_{1\leq j\leq c} \{\Delta(Xk)\}} \right\} \right\} \tag{6}$$

wherein $\delta(X_i, X_J)$ defines the distance between clusters X_i and X_j (intercluster distance), $\Delta(X_k)$ represents the intracluster distance of clusterX_k, and c is the cluster number in the partition[33].

The indices were implemented with a variation in the number of clusters in the $2-20$ range for each of the selected algorithms. The data sets were evaluated using cluster internal metrics according to equations 2 and 6.

The R suite[34] was used to implement this metric with the specialized packages NbClust[35], and clValid[36]. These packages enable estimations of the optimal number of clusters present in the different data sets using the aforementioned evaluation metrics. In the subsequent analysis, we used the average of the number of clusters from each evaluation metric to serve as the best number of clusters to generate the results.

Clustering

Selection of clustering algorithms

To identify reference gene candidates based on RNA-seq data, the clustering algorithms were used on the expression profiles of the genes to find similarities between the genes in the datasets[37]. Three data clustering strategies were explored on the gene expression data: partitioning (k-means), hierarchical (agglomerative) and neural network (Self-organizing map, SOM); these algorithms are all distance based and are popular in exploring gene expression dataset[12,21].

We used the number of clusters (k), based on the optimal number of clusters identified in the previous step for all the algorithms selected. The k-means algorithm was executed based on the Euclidean distance between the genes. For the hierarchical algorithm, we used the agglomerative (down-up) method with the Ward technique. The SOM was implemented with a the height and the neuron weight according the optimal cluster number. The Weka data mining software[38] and R language (with the specialized packages factorextra and SOM) were used to perform the clustering analysis[39].

Post-Clustering

List of validated references genes (VRGs)

To select the references gene candidates list, we identified a set of validated references genes described in the literature to use as references points on the clusters to select the new candidates based on their proximity with these genes. This generic list is formed of genes that were used in bacterial expression studies[20].

A total of 9 genes *(ftsZ, gap, gyrA, gyrB, recA, secA, rho, rpoA and rpoB)* identified in the literature as VRGs were selected as the generic list for all the bacterial genomes used in the study. The selection was based on the criteria of[2,4,25], who identified a set of reference genes for bacterial studies through a literature review and then selected the genes validated by two or more RT-PCR studies.

After applying each clustering algorithm, we selected only those clusters where one or more VRGs were present; this allows to reduce the search space and select the genes that have greater proximity to the VRGs with respect to their Euclidean distance.

Distance matrix calculation for each expression condition

The distance matrix of each gene of the cluster was calculated with respect to the VRGs based on the Euclidean distance, which is a metric commonly used to assess dissimilarity. Thus, the Euclidean distance between points X and Y in a Euclidean p-space is calculated according to equation 7:

$$d_{euc}(X,Y) = \left(\sum_{j=1}^{p} (x_j - y_j)^2 \right)^{\frac{1}{2}} \tag{7}$$

The Euclidean distance was calculated for each dataset (i.e., strain) under each stress condition based on the gene expression values.

$$d_{euc} \rightarrow A \subseteq B \tag{8}$$

wherein A is a strain and B is a specific stress condition. The distance matrix A caters to a specific stress condition to which the microorganism has been subjected and hence is logically a subset of the overall matrix B which represents all the considered stress conditions. Based on the Euclidean distance matrices, a heuristic method was used to identify the genes closest to the VRGs and also the genes whose gene expression levels did not vary significantly under the different stress conditions to enable the selection of candidate reference genes.

Creation of sub-matrices based on the cut-off threshold

Sub-matrices were created to select the closest genes based on the Euclidean distance matrix using the second distance quartile $(q2)$ as a cut-off threshold because this threshold included the elements close to the VRGs. Then, the filter $y \leq q_2$ (y is any gene in the matrix) was applied if under a specific stress condition, the gene belongs to the sub-matrix. Algorithm 1 was implemented for this purpose, thereby enabling the creation of a sub-matrix for each stress condition separately.

Algorithm 1: To create distance sub-matrices based on the established cut-off threshold

input : Distance matrix: $d_{euc}[m_i, n_j]$
output : Matrix of Quartile 2: $M_{q1}[m_i, n_j]$

```
/* calculation of the second quartile of the input matrix */
```
$Q_1 = (d_{euc} + 1)/4$;
```
/* Creation of the output matrix */
```
for *each* y_i *in* d_{euc} **do**
  if $y_i \leq Q_2$ **then**
    $Write\ y_i$ in $M_{q1}[m_i, n_j]$
  end
end
return M_{q1}

Where m_i are the VRG used as references and n_j are the genes in the distance matrix. The distances of each gene present in the sub-matrices of the different conditions in relation to the VRGs (Fig 2) were next analyzed to identify the closest genes as possible reference genes candidates.

If y is an VRG with constant expression and little variation in all stress conditions C_1,C_2,C_2...C_n and A is a gene with little variation in its expression level under all conditions and is close to y in relation to (q_2), then A is a possible candidate reference gene for the strain under consideration in Figure 2. The set of genes obtained using this approach was identified for each clustering algorithm.

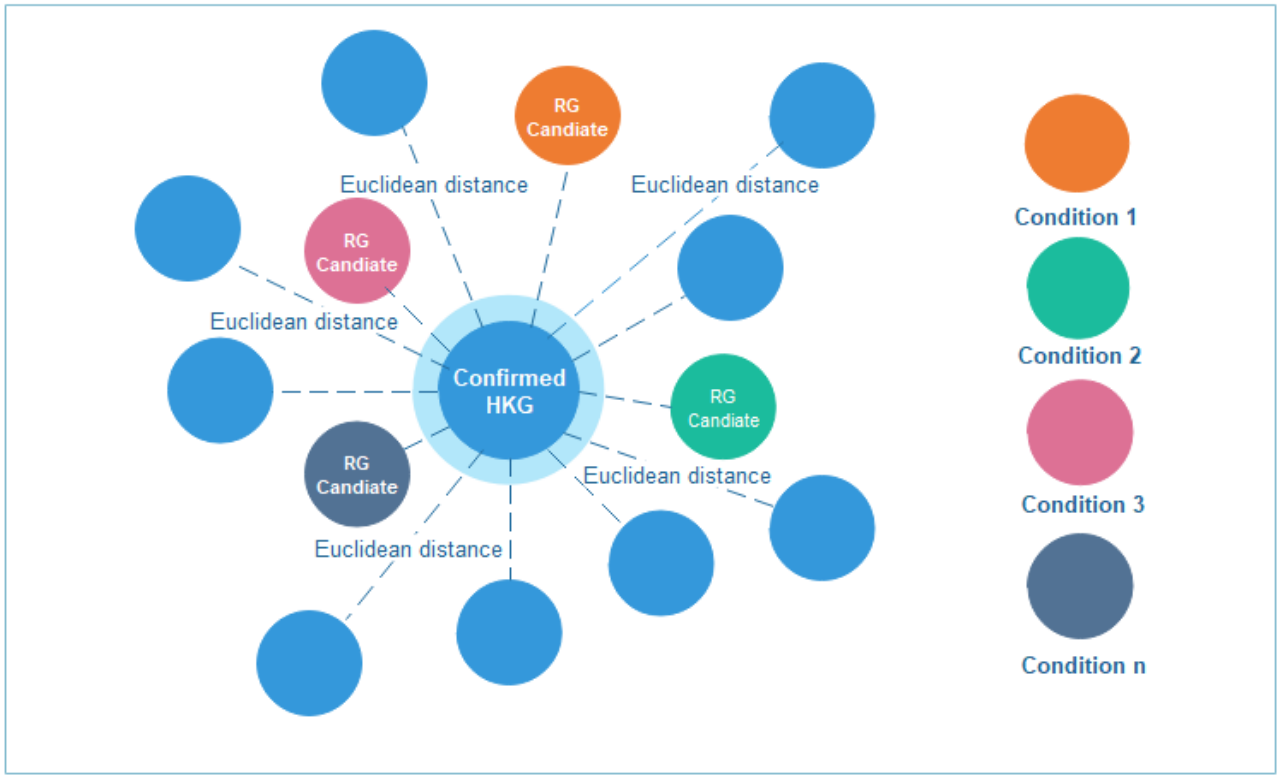

Figure 2. Example of the distances of each gene present in the sub-matrices of the different conditions in relation to the VRG

Filtering and Validation of possible candidate reference genes

To filter the genes identified as possible RG candidates, a stability test was performed based on the coefficient of variation (CV), which was used as a statistical metric to compare the degree of variation between genes regardless of their mean expression levels[40], as defined by the equation 9 :

$$CV = \frac{S}{X} \tag{9}$$

where S is the standard deviation of gene expression and X is the mean gene expression level under the different stress conditions. This method enabled the selection of genes with the lowest expression variation by setting $CV < 15$ as the threshold. Another metric used to evaluate candidates was the the standard deviation (SD); we only selected genes with an $SD < 1$[40].

These statistical metrics enable the identification of genes with stable expression levels, low coefficients of variation and low standard deviation between strains as candidate RGs.

Approach for accuracy test

To test the accuracy of the approach, we selected the list of essential and conditional genes (where, conditional genes are the ones that have not been widely adopted by existing essential gene databases), using the *E-Coli* and *Mycobacterium tuberculosis*, a phylogenetically close organism of the *C. pseudotuberculosis*. The datasets were obtained from[41]. We performed a scoring test to check how many of these reported essential or conditional genes can be identified by our approach using the same set of 9 VRGs as mentioned before,using the equation 10:

$$Accuracy = \frac{TGE + TGC + VRG}{TGI} \tag{10}$$

where TGE is the number of essential genes identified, TGC is the number of conditional genes identified, VRG is the number of validated reference genes and TGI is the total of genes identified by the approach. This test allows us to verify the ability of our method in identifying the possible reference genes. A high score obviously suggests that our candidate genes can be classified as an essential or conditional gene, and possess greater probabilities of being an actual housekeeping or reference gene[42].

ClustREFGenes web tool

ClustREFGenes is developed in Python and R as back-end programming languages. PHP, JavaScript and Flask are used as the front-end programming language. The tool received as input a genes expression matrix and the list of VRGs used as references.

ClustREFGenes can normalize expression matrix to $log_2 + 1$, in case the user requires it. The maximum number of clusters used to determine the optimal number can be specified by the use.

Results and discussion

Pre-Clustering

PGAP[28] was used to identify the genes of the core genome based on the pangenomic approach. We identified 1285 genes for strain Cp-258 belonging to the core genome for the set of 12 genomes of the biovar equi, 1191 of which (55% of total) showed gene expression in the strain. For the 26 genomes of biovar ovis, 1072 of the 1116 genes belonging to the core genome (54% of the total genome) were expressed in strain 1002. For the *E. coli* MG1655 dataset, we used all genes present in the genome to compare the results between core genome and whole genome Table 1. We eliminated the genes with null expression from the dataset.

The expression data were transformed into $log_2 + 1$, which allowed us to obtain a dataset closer to real values by suppressing outliers. Thus, we observed that clustering algorithms based on distance showed good performance in defining the border separating the clusters.

Outliers and extreme values were detected used the Interquartile range formula $[Q_1 - k(Q_3 - Q_1), Q_3 + k(Q_3 - Q_1)]$ where k is a constant and $IQR = Q_3 - Q_1$. The IQR was implemented used the InterQuartileRangeFilter from Weka[38]; this filter uses k=*3*, to define the genes as an outlier and $K = 3 * 2$ to define an instance as extreme value, and the formula guarantees that at least 50% values are considered non-outliers. We got as final data sets: 1173 genes for Cp258, 1109 genes for Cp1002 and 4051 genes for E.Coli MG1655.

Genomes	Core Genome Genes	Genes of the core genome expressed in the genome	Genes with null values, outliers and extreme value	Final Dataset
Cp-258	1285	1191	18	1173
Cp-1002	1114	1110	1	1109
*E.Coli-MG1655**	4293	0	188	4051

Table 1. Data processing to obtain the final genome datasets

* In E.coli dataset, the whole genome was used.

To verify the distribution of the genes belonging to the reference genes list, we performed a Principal Component Analysis (PCA) Fig 3, where we defined three classes: 0- Genes used as reference genes; 1- the essential genes in each genome and 2- the other genes in the genomes. This analysis allowed us to verify the proximity between the genes used as a reference and the essential genes, as well as the data for the projection of the clusters. The essential genes set, was obtained from[41], using *Mycobacterium pseudotuberculosis* as the reference for *C. pseudotuberculosis*, as they are phylogenetically related, while for *E.coli* we used the same genome.

Through the distribution of the PCA, we were able to verify the proximity between the RG and the essential genes. Consequently, we can infer that there may be co-expression between these genes and for this reason we could identify new RG candidates through the expression and the proximity-based clusters that these form with the candidate genes[20,43].

The data-sets were evaluated using the indices: SDdw and Dunn using Nbclust[35] and Clvalid[36] packages in R[34] to identify the optimal cluster number for each strain. We took the average of both metrics as the optimal number of clusters using expression data for genes as input. To select the optimal number of clusters we implemented the metrics with the number between 2 to 20 clusters and Table Table 2 shows the results for each strain and the different algorithms.

Clustering

To select the best clustering algorithms, several tests were performed with different algorithms, such as those based on statistical models i.e., Expectation maximization (EM)[44] and those based on disinfection i.e., DBSCAN[45]. DBSCAN could not identify the clusters present in the data and the EM algorithm demanded a lot of memory to process the data. The algorithms that had the best performances were the ones based on the partition clustering such as K-means, the ones based on connectivity clustering such as hierarchical, and the ones based on the artificial neural networks clustering such as Self Organization Map. These three

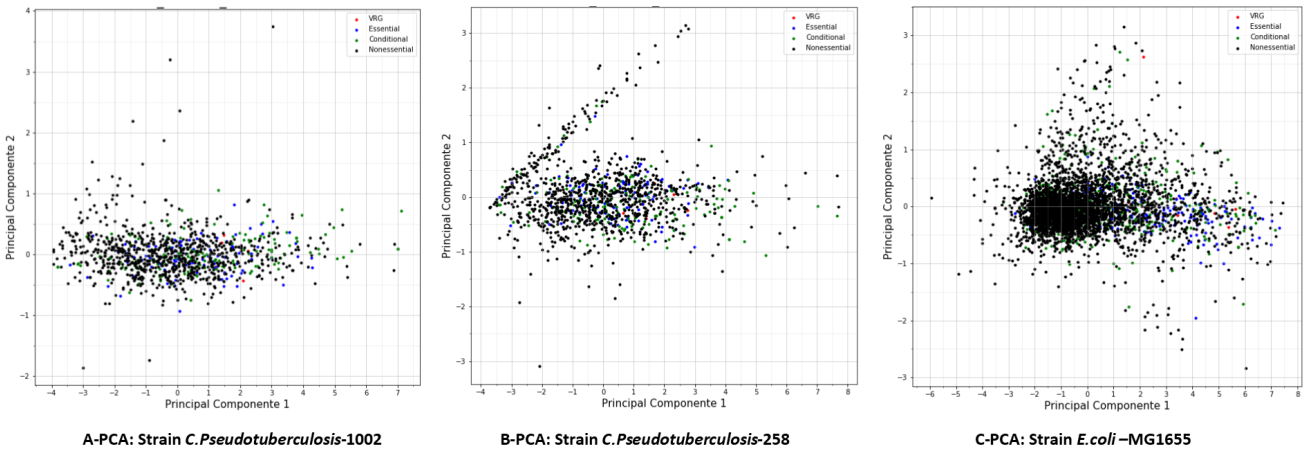

Figure 3. Principal Component Analysis (PCA) with the distribution of the genes belonging to the reference genes list.

Indexes	Hierarchical algorithm			Kmeans algorithm			SOM algorithm		
	Cp-258	**Cp-1002**	***E.coli***	**Cp-258**	**Cp-1002**	***E.coli***	**Cp-258**	**Cp-1002**	***E.coli***
SDbw	20	10	20	18	11	20	11	18	12
Dunn	15	5	2	9	17	13	5	6	16
Average	**17**	**7**	**11**	**13**	**14**	**16**	**12**	**12**	**14**

Table 2. Evaluation indices to get the optimal number of clusters for each strain.

algorithms were selected to identify the clusters with a similar profile within the data-sets and each of these algorithms use different methodologies to define the clusters.

The clustering algorithms were implemented based on the average of number of cluster results from the cluster evaluation metrics using Weka and R, and the shows the sizes of the clusters formed by the different algorithms. Fig 4 shows the graphical distribution of the clustering algorithms for each strain.

Post-Clustering

After clustering, the list of VRGs for each of the genomes were adapted, based on the genes present in that specific genome. Table 3 shows the list of VRGs adapted for each genome. This allowed selecting only the clusters where one or more VRGs were present. For each of these clusters with VRGs, we calculated the Euclidean distance to identify the genes that are closest to each VRG.

Adapted validated references genes list	
Cp-258	gap, gyrA, gyrB, recA, rho, rpoA, rpoB and secA
Cp-1002	ftsZ, gap, gyrA, gyrB, recA, rho, rpoA, rpoB and secA
E.coli	ftsZ, gyrA, gyrB, rho,recA, rpoB, and secA

Table 3. Adapted validated references genes list. Adapted reference genes list for each genome based on[4] where all the genes have been tested in three or more studies. In addition for E-coli strain, we used the three genes used as references from[25]

.

In order to select the cutoff of the Euclidean distance and the best variation coefficient to identify the possible candidates to RGs, we experimented with different distance cut-offs and CVs. For this, we developed a cross-validation test based on the different combinations of using only the VRGs, where we verify if using a specific cutoff these chosen VRGs could find the other VRGs in the data set. With this test we tried different combinations of cutoff (Euclidean distance: 1st quartile, median and mean and with CV: $<= 10, <= 15$ and $<= 20$; other metric tried was the M measure from[46]). With this test, we determined that the best cutoff for both metrics was the 1st quartile of the Euclidean distance (Q1), coefficient of variation less than 15 ($CV <= 15$) and the standard deviation less that 1 $SD < 1$. With these cutoffs, we could identify the genes that have greater

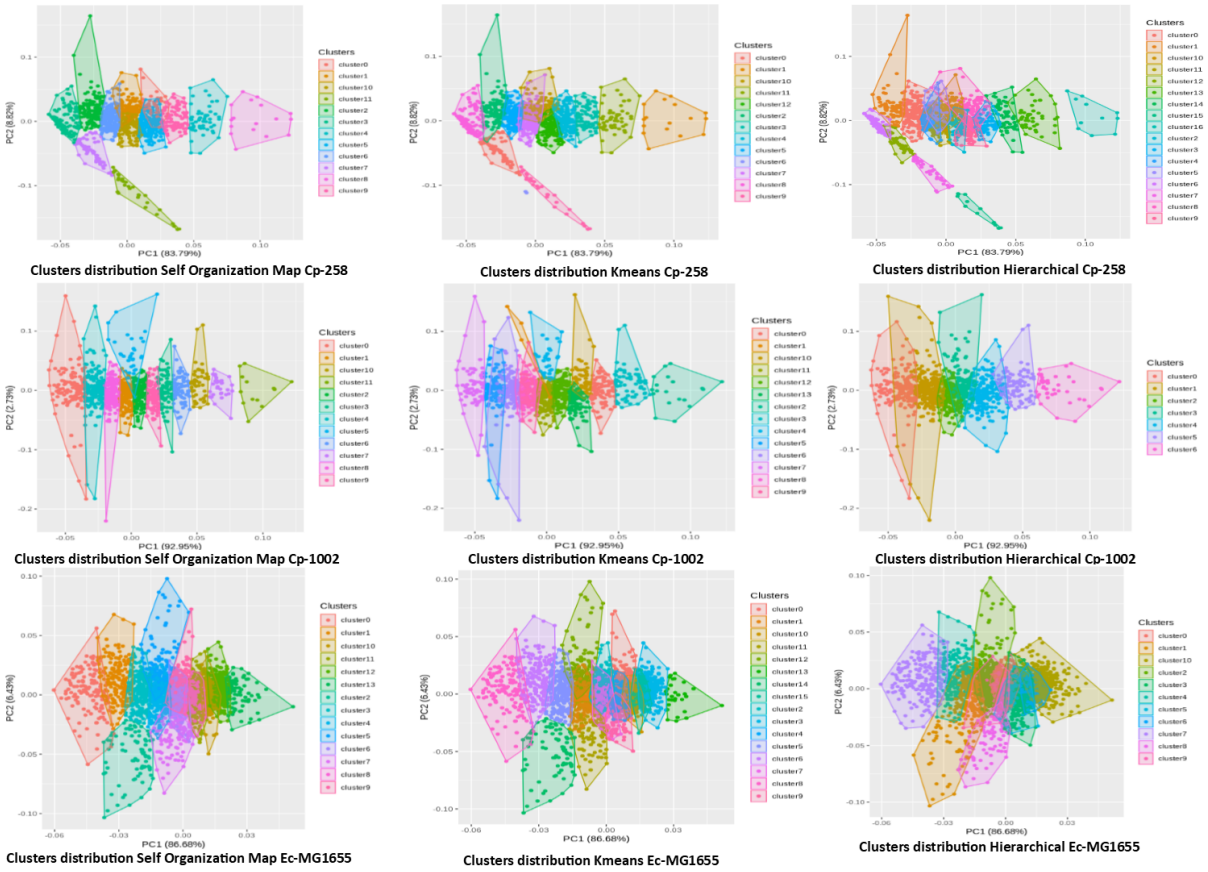

Figure 4. Cluster size for each dataset and algorithm. The graphical distribution of the clusters in each data-set with the different algorithms applied. These points are projections of each genome on the first two principal components (PCs)

proximity to the VRGs and greater stability which is a desired characteristic for the reference genes.

Table 4. Final candidate reference genes list using each algorithm and genome. The table shows the final number of candidates for reference genes selected by each algorithm in each genome

Genomes	Hierarchical	K-means	SOM
Cp-258	52	58	73
Cp-1002	125	76	52
E.Coli-MG1655	83	46	41

With this cutoff, we could get a final list of potential candidates for reference genes, for each data set and each algorithm. The Table 4 shows the number of genes selected as possible reference genes. The complete final list of genes for each algorithm and genome can be obtained from . For each gene, we got its respective sequence, and annotated them using the GO FEAT[47] web tool, which allowed us to get the functional annotation of each sequence, and to verify its ontology in the genome. Fig 5 shows the main subsystems in which the genes are present. The annotation list can be consulted in .

As seen in Table 4, the hierarchical and Self Organization Map algorithms showed relative stability when defining the candidate genes; this can be seen in all the three genomes that were studied. The K-means algorithm showed greater difficulty in defining the final list of genes in the Cp-258 genome, where it selected almost twice as many candidates (94 genes) as compared to the other genomes; this is because of the characteristic of the data in this genome.

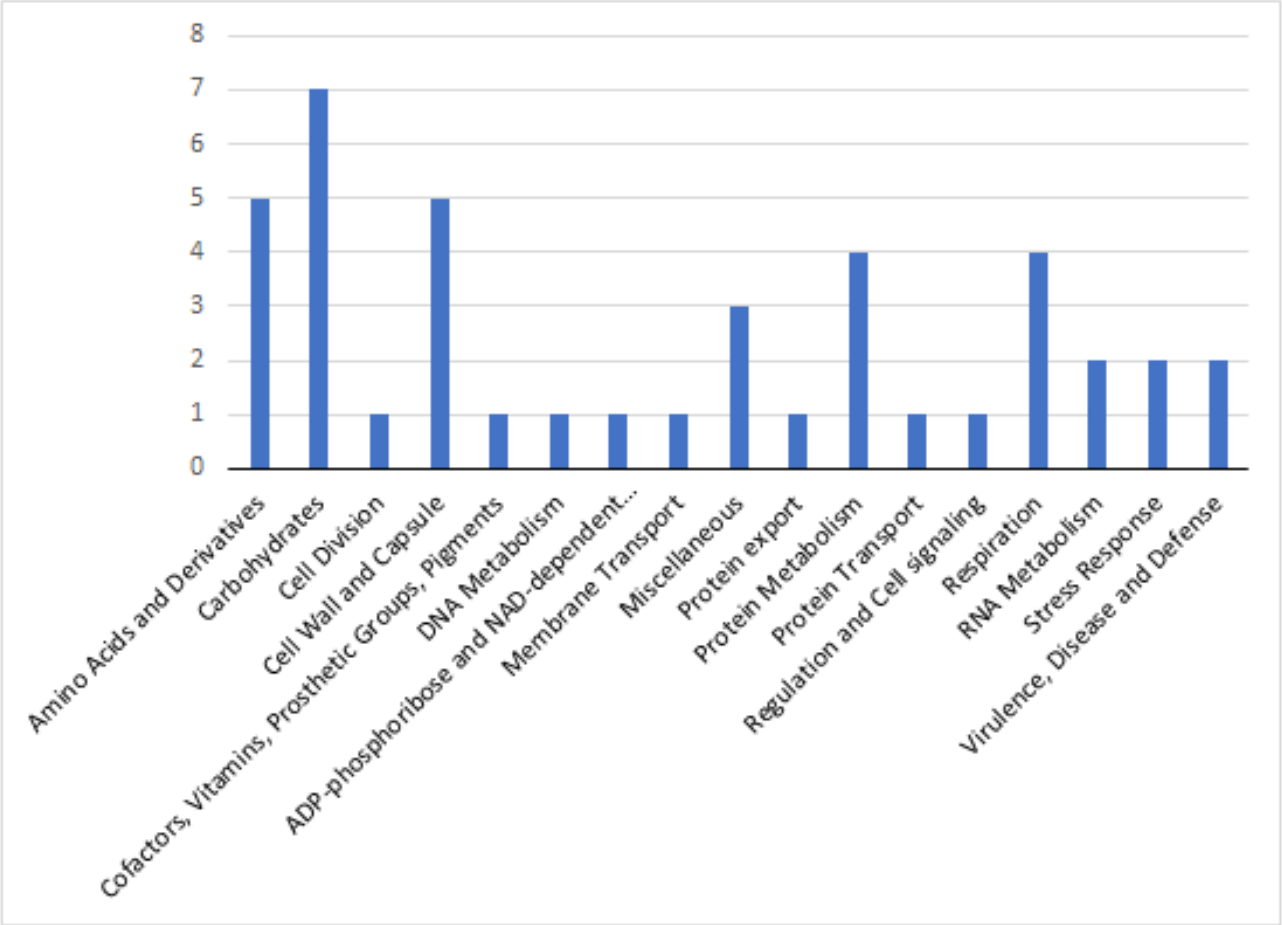

Figure 5. Main subsystems in which the genes are present.

In the results of the final list of the candidates of reference genes, we could verify the different methodologies that the algorithms used to define the clusters within the data set. We saw these characteristics in the size of the clusters defined by each algorithm, and the selection of the final genes based on the cardinality of the final set of genes for each algorithm.

The genes selected as possible candidates shows relative stability in the different stress conditions to which they were subjected, and reinforces the possibility that they can be used as reference genes.

Results from the accuracy test.

In order to develop this test, we excluded from the data sets of Cp-258 and Cp-1002, all the genes that were identified as hypothetical proteins, because they could not be classified in the categories we defined earlier for serving as reference (i.e., essential, conditional and non-essential). Table 5 shows the accuracy of the different algorithms and genomes in relation to the number of essential and conditional genes selected.

Algorithms	***C. pseudotuberculosis Strain 258***			***C. pseudotuberculosis Strain 1002***			***E.Coli Strain MG1655***		
	Total Genes	**Essential/ Conditional Genes**	**Acurracy**	**Total Genes**	**Essential/ Conditional Genes**	**Accuracy**	**Total Genes**	**Essential/ Conditional Genes**	**Accuracy**
Hierachical	33	24	72.73%	89	54	60.67%	83	47	56.62%
kmeans	38	25	65.78%	58	41	70.68%	46	29	63.04%
SOM	46	29	63.04%	37	31	83.78%	41	25	60.97%

Table 5. Acurracy results. Result of the accuracy test of the different genomes and algorithms, using the list of essential genes as reference

As can be seen in the results, the genome Cp-258 and Cp-1002 obtained a better accuracy in the selection of a greater number of candidate genes. In relation to the genome of *E. Coli*, it can be observed how the accuracy decreases in relation to the other genes; we can also observe that fewer essential genes were selected. This is due in part to the inversely proportional relation that a higher genome size has a smaller amount of essential genes.

Table 6 shows the final list of genes and their classification in the genomes that were used as a reference. For this list, we merged the results of each of the algorithms. Genes were classified as essential, conditional and non-essential. We also identified the genes used as VRG in the approach and those that were identified as possible RGs.

C. pseudotuberculosis Strain 258		*C. pseudotuberculosis Strain 1002*		*E.Coli Strain MG1655*	
adk*	aceF	aroK*	qcrA**	bamD*	cof
argD*	ackA	asd*	rplC**	efp*	cyoB
aspB*	cat1	cobG*	rpsD**	gatB*	cyoC
ccdA*	cfp30B	ctaB*	rpsE**	gatD*	fnr
ctaC*	cobF	ctaC*	rpsL**	hfq*	focA
emrB*	corA	fusA*	trpS**	nusA*	gatA
glgX*	csdA	gatB*	ubiE**	polA*	gcvT
glpR*	dprE1	glf*	ackA	prc*	gnd
gltA*	etfA	gpmA*	ag84	prfB*	mprA
hemY*	etfB	ileS*	ahpC	ptsN*	nuoA
infB*	fxsA	infC*	apt	rimP*	nuoE
ispG*	gltX1	leuS*	clpP1	rplP*	nuoJ
ligA*	hflX	mapB*	cobB	rpsC*	nuoM
pgk*	ldh	ndk*	ctaF	rpsO*	panD
pimB*	marR3	nusB*	deoR	rpsU*	pck
ppc*	mntR	pgk*	entD	slyD*	pepD
relA*	nudF	ppa*	fagD	ychF*	pldB
tal*	pgsA	ppc*	ftsW1	eno**	ptsH
xerD*	plsC1	pspA*	lutA	fmt**	rcsB
dapA**	rimJ	rfe*	merR3	hemB**	rfbB
fmt**	rrmA	ribD*	mgtE2	holA**	rplA
galU**	rsmD	rnc*	moaB	infC**	rplK
glmS**	sigH	rplR*	mscL	lptD**	rpoZ
guaB2**	slpA	rplT*	nagD	lptE**	sspA
lysS**	tig	rpoC*	nifU	lpxC**	tolC
metG**	upp	rpsI*	pspA1	lpxD**	xseB
proS**	gap**◇	rpsM*	rrmA	plsB**	ybaB
purQ**	gyrA**◇	rpsS*	sigH	rplO**	ybbN
rplB**	gyrB**◇	tatA*	ubiA	rpmD**	ybeX
rplD**	recA**◇	thrS*	whiB	rpoC**	ydgJ
rpsC**	rho**◇	tpiA*	yjiN	rpsB**	yeaD
secY**	rpoA**◇	atpB**	ftsZ	rpsM**	ygiB
valS**	rpoB**◇	cysS**	gap**◇	rpsN**	ylaC
	secA**◇	dapA**	gyrA**◇	secY**	rho◇
		efp**	gyrB**◇	topA**	ftsZ**◇
		hemA**	recA**◇	amn	gyrA**◇
		hisS**	rho**◇	aspC	rpoB**◇
		ilvE**	rpoA**◇	bcp	secA**◇
		lysS**	rpoB**◇		
		metG**	secA**◇		
		murG**			

Table 6. List of genes used in the accuracy test Final list of genes used for the accuracy test. The genes are classified *Essential Genes, **Conditional Genes and ◇ Genes used as VRG in the approach.

The result of our methodology was compared with the previously published software of selecting invariable reference genes named moose2[25]. In this approach, we used the same VRGs as used in our methodology and we also evaluated the final gene

list with the same genes used in article[25]. The results show a lower accuracy with respect to the essential and conditional genes selected by moose2, in three of the four datasets that were used to execute the program. Table 7 shows the results of this test.

Moose2 Program				
Genomes	**VRG**	**Total Genes**	**Essential/Conditional Genes**	**Accuracy**
C. pseudotuberculosis Strain 258	8	56	21	37.50%
C. pseudotuberculosis Strain 1002	5	31	17	54.84%
E.Coli Strain MG1655	7	39	11	28.21%
E.Coli Strain MG1655 Paper result	6*	33	10	30.30%

Table 7. Accuracy test results using Moose2 program. Final results of the accuracy test using Moose 2[25]. In this test, the dataset with the genes used to test our approach was used and we also evaluated the list of genes from the article[25]

ClustREFGenes web tool

ClustREFGenes was developed to be executed in any modern internet browser. Also, it has a clean and easy-to-use graphic interface. It's not required any kind of installation of any tool or software and users can execute projects without previous registration. The tool is freely available at http://computationalbiology.ufpa.br/ClustREFGenes/

Conclusion

Identifying reference genes are essential for expression studies and RT-PCR analyses. The proposed approach based on clustering algorithms and methods identified housekeeping genes from gene expression data generated by high-throughput sequencing platforms and could be adapted to other studies for establishing control genes for gene expression analyses.

Our method showed that it can adequately identify candidates with a high probability of being reference genes, because of its stability in the expression at different levels of stress that were studied here.

The proposed methodology can be used with the set of genes of the core genome or with the whole genome, however, using the core genome allows improving the accuracy of the genes that are selected as candidates.

An important point is the selection of the genes that will be used as VRGs. The result of the algorithms will be directly proportional to the accuracy of these genes. We suggest using only those VRGs that are validated by two or more studies, preferentially by laboratory experiments.

We only tested our methodology and pipeline in prokaryote data, but our approach is also extendable for use in the eukaryotic genome. As future work, we will implement the eukaryotic version of the house keeping gene identification software.

Additional information

S1 File. Cluster sizes: The file shows the sizes of the clusters formed by the different algorithms.

S2 Files. Complete Genes List: The complete final list of genes for each algorithm and genome..

S3 Files. The genes annotation lists.

S4 Files. Genomes Data-sets.

Data availability

The data an the codes generated and analyzed during the current study, are publicly available in: https://github.com/edianfranklin/ClustREFGenes

Author contributions statement

EF: Designed the pipeline and performed the machine learning experiments, developed and performed early tests on first version of a sorting algorithm, analyzed the data and wrote the final version of the manuscript. DM: Designed the pipeline, developed and released the local and web versions of ClustREFGenes, analyzed the data and wrote the final version of the manuscript.RA:Conceived, supervised and performed the machine learning experiments LG: Supervised and performed genomic and bioinformatics experiments. VA: Supervised the different aspects of this work. AS:Supervised the different aspects of this work.PG: Supervised the different aspects of this work and wrote the final version of the manuscript.JM: Conceived, designed and supervised the project. RR: Conceived, designed and supervised the project and wrote the final version of the manuscript. All authors contributed to the writing of the paper. All authors read and approved the final manuscript.

Acknowledgments

The present study was conducted with support from the Partnerships Program for Education and Training (Programa de Alianças para la Educação y la Capacitação – PAEC) Agreement between the Organization of American States (OAS), the Coimbra Group of Brazilian Universities (Grupo Coimbra de Universidades Brasilei-ras – GCUB), the Brazilian Coordination for the Improvement of Higher Education Personnel (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES) and the Brazilian National Council for Scientific and Technological Development (Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq). Biological Engineering Laboratory, Federal University of Pará (Universidade Federal do Pará – UFPA). Biological Networks Lab, Virginia Commonwealth University, VA, USA.

Funding

EF and DM was supported by a master degree grant provided by the Brazilian Coordination for the Improvement of Higher Education Personnel (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES) (CAPES DS-001).

Competing interests

The authors declare that they have no competing interests.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable

References

1. Carvalho, D. M. *et al.* Reference genes for RT-qPCR studies in Corynebacterium pseudotuberculosis identified through analysis of RNA-seq data. *Antonie van Leeuwenhoek* **106**, 605–614, DOI: 10.1007/s10482-014-0231-3 (2014).
2. Zhou, Z., Cong, P., Tian, Y. & Zhu, Y. Using RNA-seq data to select reference genes for normalizing gene expression in apple roots. *PLoS ONE* **12**, 1–17, DOI: 10.1371/journal.pone.0185288 (2017).
3. Cusick, K. D., Fitzgerald, L. A., Cockrell, A. L. & Biffinger, J. C. Selection and evaluation of reference genes for reverse transcription-quantitative PCR expression studies in a thermophilic bacterium grown under different culture conditions. *PLoS ONE* **10**, 1–23, DOI: 10.1371/journal.pone.0131015 (2015).
4. Rocha, D., Santos, C. & Pacheco, L. Bacterial reference genes for gene expression studies by RT-qPCR: survey and analysis. *Antonie van Leeuwenhoek* **108**, 685–693, DOI: 10.1007/s10482-015-0524-1 (2015).
5. Greer, S., Honeywell, R., Geletu, M., Arulanandam, R. & Raptis, L. Housekeeping genes; expression levels may change with density of cultured cells. *J. Immunol. Methods* **355**, 76–79, DOI: 10.1016/j.jim.2010.02.006 (2010).
6. Dheda, K. *et al.* Validation of housekeeping genes for normalizing RNA expression in real-time PCR. *BioTechniques* **37**, 112–119 (2004).
7. Taylor, S., Wakem, M., Dijkman, G., Alsarraj, M. & Nguyen, M. A practical approach to RT-qPCR-Publishing data that conform to the MIQE guidelines. *Methods* **50**, S1—-S5, DOI: 10.1016/j.ymeth.2010.01.005 (2010).

8. Bustin, S. A. *et al.* The MIQE Guidelines: Minimum Information for Publication of Quantitative Real-Time PCR Experiments. *Clin. Chem.* **55**, 611–622, DOI: 10.1373/clinchem.2008.112797 (2009). 1109.1568v1.

9. Treangen, T. J. & Salzberg, S. L. Repetitive DNA and next-generation sequencing: computational challenges and solutions. *Nat Rev Genet.* **13**, 36–46, DOI: 10.1038/nrg3117.Repetitive (2013). NIHMS150003.

10. Ching, T. *et al.* Opportunities and obstacles for deep learning in biology and medicine. *J. The Royal Soc. Interface* **15**, 20170387 (2018).

11. Vieira, A. *et al.* Comparative validation of conventional and rna-seq data-derived reference genes for qpcr expression studies of colletotrichum kahawae. *PloS one* **11**, e0150651 (2016).

12. Oyelade, J. *et al.* Clustering Algorithms: Their Application to Gene Expression Data. *Bioinforma. Biol. Insights* **10**, BBI.S38316, DOI: 10.4137/BBI.S38316 (2016).

13. Maimon, O. & Rokach, L. Introduction to Knowledge Discovery and Data Mining. In *Data Mining and Knowledge Discovery Handbook*, 1–15, DOI: 10.1007/978-0-387-09823-4_1 (Springer US, Boston, MA, 2009).

14. Fayyad, U. M., Piatetsky-Shapiro, G. & Smyth, P. Knowledge Discovery and Data Mining: Towards a Unifying Framework. *Proc 2nd Int Conf on Knowl. Discov. Data Min. Portland OR* 82–88, DOI: 10.1.1.27.363 (1996).

15. Libbrecht, M. W. & Noble, W. S. Machine learning applications in genetics and genomics. *Nat Rev Genet.* **16**, 321–332, DOI: 10.1038/nrg3920 (2015). 15334406.

16. Dong, B. *et al.* Predicting housekeeping genes based on fourier analysis. *PLoS One* **6**, e21012 (2011).

17. De Ferrari, L. & Aitken, S. Mining housekeeping genes with a Naive Bayes classifier. *BMC genomics* **7**, 277, DOI: 10.1186/1471-2164-7-277 (2006).

18. Han, J., Kamber, M. & Pei.Jian. *Data Mining: concepts and tecniques* (Morgan Kaufmann, Elsevier, Walthan, MA, 2011), 3er edn.

19. Kovács, F., Legány, C. & Babos, A. Cluster Validity Measurement Techniques. *Proc. 6th Int. Symp. Hungarian Res. on Comput. Intell.* 1–11 (2005).

20. Lercher, M. J., Urrutia, A. O. & Hurst, L. D. Clustering of housekeeping genes provides a unified model of gene order in the human genome. *Nat. Genet.* **31**, 180–183, DOI: 10.1038/ng887 (2002).

21. Dalton, L., Ballarin, V. & Brun, M. Clustering algorithms: on learning, validation, performance, and applications to genomics. *Curr. genomics* **10**, 430–45, DOI: 10.2174/138920209789177601 (2009).

22. Rendón, E., Abundez, I., Arizmendi, A. & Quiroz, E. M. Internal versus External cluster validation indexes. *Int. J. Comput. Commun.* **5**, 27–34 (2011).

23. Brun, M. *et al.* Model-based evaluation of clustering validation measures. *Pattern recognition* **40**, 807–824 (2007).

24. Pinto, A. C. *et al.* Differential transcriptional profile of corynebacterium pseudotuberculosis in response to abiotic stresses. *BMC genomics* **15**, 14 (2014).

25. Berghoff, B. A., Karlsson, T., Källman, T., Wagner, E. G. H. & Grabherr, M. G. Rna-sequence data normalization through in silico prediction of reference genes: the bacterial response to dna damage as case study. *BioData mining* **10**, 30 (2017).

26. Soares, S. C. *et al.* Genome sequence of Corynebacterium pseudotuberculosis biovar equi strain 258 and prediction of antigenic targets to improve biotechnological vaccine production. *J. Biotechnol.* **167**, 135–141, DOI: 10.1016/j.jbiotec.2012.11.003 (2013).

27. Silva, A. *et al.* Complete genome sequence of corynebacterium pseudotuberculosis i19, a strain isolated from a cow in israel with bovine mastitis. *J. bacteriology* **193**, 323–324 (2011).

28. Zhao, Y. *et al.* PGAP: Pan-genomes analysis pipeline. *Bioinformatics* **28**, 416–418, DOI: 10.1093/bioinformatics/btr655 (2012).

29. Mortazavi, A., Williams, B. a., McCue, K., Schaeffer, L. & Wold, B. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. methods* **5**, 621–628, DOI: 10.1038/nmeth.1226 (2008). 1111.6189v1.

30. Liu, P. & Si, Y. Cluster Analysis of RNA-Sequencing Data. In *Statistical Analysis of Next Generation Sequencing Data*, 191–217, DOI: 10.1007/978-3-319-07212-8 (Springer, 2014).

31. Halkidi, M. & Vazirgiannis, M. Clustering validity assessment: Finding the optimal partitioning of a data set. In *Proceedings 2001 IEEE International Conference on Data Mining*, 187–194 (IEEE, 2001).

32. Charrad, M., Ghazzali, N., Boiteau, V. & Niknafs, A. Nbclust package: finding the relevant number of clusters in a dataset. *J. Stat. Softw* (2012).

33. Bolshakova, N. & Azuaje, F. Cluster validation techniques for genome expression data. *Signal Process.* **83**, 825–833, DOI: 10.1016/S0165-1684(02)00475-9 (2003).

34. R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria (2018).

35. Ghazzali, N. NbClust : An R Package for Determining the. *J. Stat. Softw.* **61** (2014).

36. Brock, G., Pihur, V. & Datta, S. clValid: An R package for cluster validation. *J Stat Softw* **25**, 1–32, DOI: 10.18637/jss.v025.i04 (2008).

37. Si, Y., Liu, P., Li, P. & Brutnell, T. P. Model-based clustering for RNA-seq data. *Bioinformatics* **30**, 197–205, DOI: 10.1093/bioinformatics/btt632 (2014).

38. Hall, M. *et al.* The WEKA data mining software. *ACM SIGKDD Explor.* **11**, 10–18, DOI: 10.1145/1656274.1656278 (2009).

39. Ross, I. & Gentleman, R. R: a language for data analysis and graphics. *J. computational graphical statistics* **5**, 299—-314 (1996).

40. de Jonge, H. J. M. *et al.* Evidence based selection of housekeeping genes. *PLoS ONE* **2**, 1–5, DOI: 10.1371/journal.pone.0000898 (2007).

41. Chen, W.-H., Minguez, P., Lercher, M. J. & Bork, P. Ogee: an online gene essentiality database. *Nucleic acids research* **40**, D901–D906 (2011).

42. Kozera, B. & Rapacz, M. Reference genes in real-time pcr. *J. applied genetics* **54**, 391–406 (2013).

43. Corrales, M. *et al.* Clustering of drosophila housekeeping promoters facilitates their expression. *Genome research* **27**, 1153–1161 (2017).

44. Moon, T. K. The expectation-maximization algorithm. *IEEE Signal processing magazine* **13**, 47–60 (1996).

45. Ester, M., Kriegel, H.-P., Sander, J., Xu, X. *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, vol. 96, 226–231 (1996).

46. Vandesompele, J. *et al.* Accurate normalization of real-time quantitative RT-PCR data by geometric averaging of multiple internal control genes. *Genome Biol.* **3**, 31–34, DOI: 10.1186/gb-2002-3-7-research0034 (2002). 1465-6906.

47. Araujo, F. A., Barh, D., Silva, A., Guimarães, L. & Ramos, R. T. J. Go feat: a rapid web-based functional annotation tool for genomic and transcriptomic data. *Sci. reports* **8**, 1794 (2018).