Computer Methods and Programs in Biomedicine 221 (2022) 106865

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

PhageCocktail: An R package to design phage cocktails from experimental phage-bacteria infection networks

María Victoria Díaz-Galián[a], Miguel A. Vega-Rodríguez[a,*], Felipe Molina[b]

[a] Escuela Politécnica, Universidad de Extremadura (https://ror.org/0174shg90), Avda. de la Universidad s/n, Cáceres, 10003, Spain
[b] Facultad de Ciencias, Universidad de Extremadura (https://ror.org/0174shg90), Avda. de Elvas s/n, Badajoz, 06006, Spain

ARTICLE INFO

Article history:
Received 16 September 2021
Revised 18 April 2022
Accepted 7 May 2022

Keywords:
Phage therapy
Phage cocktail
Phage-bacteria infection network
R package

ABSTRACT

Background and objective: Phage therapy is a resurgent strategy used in medicine and the food industry to lyse bacteria that cause damage to health or spoil a food product. Frequently, phage-bacteria infection networks have a large size, making it impossible to manually study all possible phage cocktails. Thus, this article presents an R package called `PhageCocktail` to automatically design efficient phage cocktails from phage-bacteria infection networks.

Methods: This R package includes four different methods for designing phage cocktails: ExhaustiveSearch, ExhaustivePhi, ClusteringSearch, and ClusteringPhi. These four methods are explained in detail and are evaluated using 13 empirical phage-bacteria infection networks. More specifically, runtime and expected success (fraction of lysed bacteria) are analyzed.

Results: The four methods have variations in terms of runtime and quality of the results. ExhaustiveSearch always provides the best possible phage cocktail, but its runtime could be long. ExhaustivePhi only focuses on one cocktail size, the one estimated as the best; thus, its runtime is less than ExhaustiveSearch, but it can produce cocktails with more phages than necessary. ClusteringSearch and ClusteringPhi are very fast (generally, less than one millisecond), providing always immediate results due to clustering techniques, but their accuracies can be lower, yielding cocktails with lower expected successes.

Conclusions: The larger the phage-bacteria infection network is, the more complex its analysis is. Thus, this tool eases this task for scientists and other users while designing phage cocktails of good quality. This R package includes four different methods; therefore, users may choose among them, considering their preferences in speed and accuracy of results.

1. Introduction

The discovery of penicillin in 1928 began the golden age of antibiotics for treating infectious diseases. Antibiotics frequently show a wide spectrum of activity, which makes many modern medical procedures, such as complex surgery, possible [1]. However, at the dawn of the third millennium, the decline in antibiotic discovery and the evolution of multidrug-resistant bacteria have led to a global antimicrobial resistance crisis. Additionally, the interest in maintaining a healthy intestinal microbial ecosystem balance (eubiosis) and the required preservation of bacterial species involved in fermenting foods limit the use of antibiotics. These issues have elicited renewed worldwide interest in using phages as a viable alternative approach for the clinical management of bacterial infections [2] and for reducing bacterial loads in raw and processed foods [3].

Virulent phages are viruses that specifically infect and destroy bacteria while concurrently releasing their progeny into the environment. In addition to this rapid exponential proliferation, the selective toxicity of phages spares useful microbiota in animals and foods [4]. Félix d'Herelle discovered phages in 1917 and soon speculated that they were responsible for the frequent recovery from diarrhea due to their antibacterial activity. However, the widespread availability of antibiotics and the narrow host range of phages undermined the enthusiasm for phage therapy by the 1940s [5]. Because many antibiotic discovery programs in major pharmaceutical companies are currently being discontinued [6], phage therapy merits a second chance.

* Corresponding author.
E-mail addresses: mvdiazgalian@unex.es (M.V. Díaz-Galián), mavega@unex.es (M.A. Vega-Rodríguez), fmolina@unex.es (F. Molina).

https://doi.org/10.1016/j.cmpb.2022.106865

Phage therapy is based on phage cocktails, which are combinations of phages that can lyse target bacteria [7]. The manual design of efficient phage cocktails is difficult or impossible because many Phage-Bacteria Infection Networks (PBINs) are large in size. Thus, this paper presents an R package to automatically design phage cocktails. The primary contributions of this study can be summarized as follows:

- Presentation and explanation in detail of four computer methods (ExhaustiveSearch, ExhaustivePhi, ClusteringSearch, and ClusteringPhi) to design phage cocktails considering a given PBIN.
- Clustering* methods are very fast (generally, less than one millisecond), but their accuracy can be less. Conversely, Exhaustive* methods take more time, but their accuracy is the highest possible. Additionally, as will be explained, some methods use the ϕ (Phi) estimation value.
- The four methods have been included in an R package that is available for any user.
- A complete experimental study that evaluates and compares the runtime and expected success (fraction of lysed bacteria) for the obtained cocktails.
- Analysis of the impact when additional phages are included in the cocktail, from both viewpoints: runtime and biological quality, also including additional insights.

This paper is structured in the following way. Section 2 reviews related literature. Section 3 explains the four proposed methods to design phage cocktails. Section 4 is devoted to the experiments performed in this study, which analyze the four methods from both viewpoints (runtime and expected success; fraction of lysed bacteria) and include additional insights. Finally, Section 5 concludes this study.

2. Related work

Phage therapy is typically used by formulating a "cocktail" composed of a variable number of phages. Phage cocktails are or can be useful in many applications, such as treatment of human tuberculosis [8] or respiratory infections [9], prevention of contamination in food (e.g., *Salmonella* in milk and chicken meat [10]), improvements in agriculture [11], quality improvement of drinking water [12], etc.

Although phages comprise the most prevalent and genetically diverse biological entities on Earth, the antagonistic interactions and fast coevolution with their bacterial hosts hamper the permanent prediction of which phages can infect which bacterial strains, explaining why metagenomics tools cannot stably predict these infections [13]. Thus, new empirical data [14] are continuously collected as bipartite phage-bacteria infection networks that must be analyzed to generate phage cocktails.

Additionally, because broad host range phages face "life" history trade-offs, such as lower virulence [15], long-term biocontrol strategies might necessitate time-structured phage cocktails or cycling of cocktails [16]. Thus, computer methods for designing phage cocktails are required for different purposes.

Regarding the computer methods proposed for phage cocktails, the structure of PBINs relies on the coevolutionary dynamics of phages and bacteria [17,18] and has been used recently to estimate the size of phage cocktails [19]. However, the proposed metric (ϕ) does not consider every edge on phage-bacteria networks or generate the minimum cocktail size. These are two important contributions of this study, together with the detailed explanation of four computer methods to design phage cocktails, their inclusion in an R package, and a complete experimental study, which analyzes runtime, biological quality, and the impact when additional phages are included in the cocktail.

3. Material and methods

This R package designs the best phage cocktail for a given phage-bacteria infection network. Therefore, the corresponding PBIN will be an input parameter in the form of an input host range matrix. Also, the user must select one of the four possible methods for designing the phage cocktail: ExhaustiveSearch, ExhaustivePhi, ClusteringSearch, and ClusteringPhi. Once the parameters have been supplied, the program performs preprocessing of the input matrix and, afterwards, executes the specific method chosen. This R package is freely available in CRAN (https://cran.r-project.org/package=PhageCocktail).

3.1. Input parameters

To execute PhageCocktail, some parameters must be given. The first one is the path of the input host range matrix (PBIN) in.xlsx format, whose rows and columns are the bacteria and phages, respectively. The first row must be devoted to the names/codes of the phages. In the same way, the first column must be devoted to the names/codes of the bacteria. Only 0 and 1 are considered to describe the infection relationship between involved phages and bacteria (0 = does not lyse; 1 = lyses). Thus, the infection relationship can be represented by the names of phages and bacteria and by the position of the value (0 or 1) in the matrix.

Other parameter indicates the method selected to design the phage cocktail: *ExhaustiveSearch* (Section 3.3), *ExhaustivePhi* (Section 3.4), *ClusteringSearch* (Section 3.5), and *ClusteringPhi* (Section 3.6).

The input host range matrix and the method to use are the most important input parameters, but there are others, such as the filename of the output, just when is chosen to generate the results into an output file. This output file eases the subsequent analyses. To finish, a limit value can be established. Thus, the phage cocktail size is limited to this value. This parameter is useful in situations where the runtime is high (very large PBINs). Additionally, it is important to highlight that phage cocktails lysing the same bacteria with fewer phages are preferred because large phage cocktails can produce problems such as horizontal transfer of undesired genes, dysbiosis or high manufacturing costs [19]. Thus, the limit parameter can be useful in some situations.

3.2. Preprocessing

Before explaining the four possible methods for designing phage cocktails, there is a previous common step of preprocessing. The pseudocode of this preprocessing is detailed in Algorithms 1

Algorithm 1 Pseudocode of the preprocessing for bacteria.

Input: *NumRows*: number of rows (bacteria) in the matrix.
Output: *MaxBacteria*: number of bacteria that can be lysed. *nonlysedBacteria*: list with nonlysed bacteria. *lysedBacteria*: list with lysed bacteria.

```
MaxBacteria ← NumRows
nonlysedBacteria ← ∅
lysedBacteria ← ∅
for each Bacterium ∈ NumRows do
    # Is this bacterium lysed by any phage?
    if size(Phages[Bacterium]) = 0 then
        MaxBacteria ← MaxBacteria − 1
        nonlysedBacteria ← nonlysedBacteria ∪ {Bacterium}
    else
        lysedBacteria ← lysedBacteria ∪ {Bacterium}
    end if
end for
```

(for bacteria) and 2 (for phages). In this step, all rows and columns

Algorithm 2 Pseudocode of the preprocessing for phages.

Input: *NumColumns*: number of columns (phages) in the matrix.
Output: *MaxPhages*: number of useful phages. *nonusefulPhages*: list with nonuseful phages. *usefulPhages*: list with useful phages.

```
1: MaxPhages ← NumColumns
2: nonusefulPhages ← ∅
3: usefulPhages ← ∅
4: for each Phage ∈ NumColumns do
5:     # Does this phage lyse any bacterium?
6:     if size(Bacteria[Phage]) = 0 then
7:         MaxPhages ← MaxPhages − 1
8:         nonusefulPhages ← nonusefulPhages ∪ {Phage}
9:     else
10:        usefulPhages ← usefulPhages ∪ {Phage}
11:    end if
12: end for
```

of the input host range matrix are inspected to reduce their dimensions (line 4 in Algorithms 1 and 2). The preprocessing checks if there is any zero vector in the matrix, which indicates that a bacterium is not lysed by any phage or that a phage does not lyse any bacterium (line 6 in Algorithms 1 and 2). Then, zero rows and columns are removed from the matrix, and these bacteria and phages are grouped into two lists: the list of non-lysed bacteria and the list of non-useful phages, respectively (lines 7–8 in Algorithms 1 and 2).

In addition, two more lists are created storing lysed bacteria and useful phages (line 10 in Algorithms 1 and 2). Once the matrix is preprocessed, it is ready for the next step, which depends on the method chosen for designing the phage cocktail.

3.3. ExhaustiveSearch

The first method proposed is ExhaustiveSearch, in which all useful phages will be considered. The pseudocode for this method is detailed in Algorithm 3. Using this method, the program returns

Algorithm 3 Pseudocode of the ExhaustiveSearch method.

Input: *MaxPhages*: number of useful phages. *MaxBacteria*: number of bacteria that can be lysed. *Matrix*: host range matrix. *limit*: limit of cocktail size.
Output: *Result*: vector with the designed phage cocktails and the bacteria lysed by those cocktails.

```
1: Result ← ∅
2: for i ← 1 to 7 do
3:     (PhageCocktail, LysedBacteria) ← Search(i, MaxPhages,
       MaxBacteria, Matrix)
4:     Result ← Result ∪ (PhageCocktail, LysedBacteria)
5:     if (LysedBacteria = MaxBacteria) or (i = MaxPhages) or (i =
       limit) then
6:         return
7:     end if
8: end for
```

the best phage cocktails from size 1 to 7 in an orderly manner (line 2 in Algorithm 3). Within a cocktail size, the best phage cocktail is the one that lyses the highest number of bacteria. This method will only stop before reaching size 7 if some previous cocktail size can lyse all the bacteria, if there are not more phages to use, or if the limit indicated in the calling is reached (line 5 in Algorithm 3).

The ExhaustiveSearch method is based on the Search function (line 3 in Algorithm 3). The first parameter of this function indicates the size of the search to perform. As an example, Algorithm 4

Algorithm 4 Pseudocode of the Search_Size4 function.

Input: *MaxPhages*: number of useful phages. *MaxBacteria*: number of bacteria that can be lysed. *Matrix*: host range matrix.
Output: *PhageCocktail*: final set of phages. *LysedBacteria*: final set of bacteria lysed by phages at *PhageCocktail*.

```
1: PhageCocktail ← ∅
2: LysedBacteria ← ∅
3: for i ← 1 to MaxPhages − 3 do
4:     for j ← i + 1 to MaxPhages − 2 do
5:         for k ← j + 1 to MaxPhages − 1 do
6:             for l ← k + 1 to MaxPhages do
7:                 bacteria ← lyse_bacteria(phage[i], phage[j], phage[k],
                   phage[l], Matrix)
8:                 if size(bacteria) > size(LysedBacteria) then
9:                     PhageCocktail ← {phage[i]} ∪ {phage[j]} ∪
                       {phage[k]} ∪ {phage[l]}
10:                    LysedBacteria ← bacteria
11:                    if size(LysedBacteria) = MaxBacteria then
12:                        return
13:                    end if
14:                end if
15:            end for
16:        end for
17:    end for
18: end for
```

shows the pseudocode for a search of size 4. The pseudocodes are similar for other sizes. As shown in Algorithm 4, a combinatorial search is performed, evaluating all the possible combinations of phages within a set of that size (lines 3–7 in Algorithm 4). If a combination of phages lyses a higher number of bacteria than the best one found until this moment (line 8 in Algorithm 4), then the new best phage cocktail is saved (lines 9-10 in Algorithm 4). If the new phage cocktail lyses the possible bacteria (line 11 in Algorithm 4), then the search is finished.

3.4. ExhaustivePhi

The ExhaustivePhi method works with all the useful phages as ExhaustiveSearch, but there is a previous step in which an estimator of phage cocktail size is calculated, called ϕ [19]. Therefore, ExhaustivePhi only provides the best phage cocktail of size ϕ. Algorithm 5 shows the pseudocode of this method.

Before calculating ϕ, other parameters, such as the nestedness temperature (T) and the fill (f) of the host range matrix, must be calculated. Nestedness temperature (0–100°) indicates the nestedness level, where low values are related to a nested network, while high values reflect an increase in disorder or a deviation from perfect nestedness [20]. This value is calculated by the BinMatNest program using a binary presence-absence matrix (i.e., the host range matrix; line 1 in Algorithm 5) [21]. The other parameter is fill, which is the percentage (0–100%) of successful infections throughout the matrix (line 2 in algorithm 5) [19], such as 100% means that all the phages lyse all the bacteria.

Once both parameters, nestedness temperature and fill, have been determined, ϕ is calculated (line 3 in Algorithm 5), as shown in Eq. (1):

$$\phi = log_2\left(\frac{b \cdot T}{f} + 2\right), \tag{1}$$

Algorithm 5 Pseudocode of the ExhaustivePhi method.

Input: *NumPhages*: original number of phages. *NumBacteria*: original number of bacteria. *MaxPhages*: number of useful phages. *MaxBacteria*: number of bacteria that can be lysed. *Matrix*: host range matrix. *limit*: limit of cocktail size.
Output: *Result*: vector with the designed phage cocktail and the bacteria lysed by that cocktail.

1: $Temperature \leftarrow BinMatNest(Matrix)$
2: $Fill \leftarrow 100 \cdot CountInfections(Matrix)/(NumBacteria \cdot NumPhages)$
3: $\phi \leftarrow PhiFormula(NumBacteria, Temperature, Fill)$ (Equation 1)
4: **if** $MaxPhages < \phi$ **then**
5: $\quad \phi \leftarrow MaxPhages$
6: **end if**
7: **if** $limit < \phi$ **then**
8: $\quad \phi \leftarrow limit$
9: **end if**
10: $Result \leftarrow Search(\phi, MaxPhages, MaxBacteria, Matrix)$

where b is the number of bacteria, T is the nestedness temperature, and f is the fill of the host range matrix.

However, the phage cocktail size can be less than the ϕ value for two reasons. The first occurs when the number of useful phages (*MaxPhages*) is less than ϕ (line 4 in Algorithm 5), and the second occurs when the limit value is less than ϕ (line 7 in Algorithm 5). In these cases, ϕ is readjusted. Finally, this method obtains the best phage cocktail of size ϕ (line 10 in Algorithm 5) calling the Search function (Algorithm 4).

3.5. ClusteringSearch

This method does not search the result using all the useful phages because they are previously clustered according to their ability and similarity to lyse bacteria. The pseudocode of this method is shown in Algorithm 6.

Algorithm 6 Pseudocode of the ClusteringSearch method.

Input: *NumPhages*: original number of phages. *NumBacteria*: original number of bacteria. *MaxBacteria*: number of bacteria that can be lysed. *Matrix*: host range matrix. *limit*: limit of cocktail size.
Output: *Result*: vector with the designed phage cocktails and the bacteria lysed by those cocktails.

1: $Temperature \leftarrow BinMatNest(Matrix)$
2: $Fill \leftarrow 100 \cdot CountInfections(Matrix)/(NumBacteria \cdot NumPhages)$
3: $\phi \leftarrow PhiFormula(NumBacteria, Temperature, Fill)$ (Equation 1)
4: $DistanceMatrix \leftarrow EuclideanDistance(Matrix)$
5: $Dendrogram \leftarrow HierarchicalClustering(DistanceMatrix)$
6: $NumClusters \leftarrow ElbowMethod(Matrix)$
7: **if** $NumClusters < \phi$ **then**
8: $\quad NumClusters \leftarrow \phi$
9: **end if**
10: $Clusters \leftarrow CutDendrogram(Dendrogram, NumClusters)$
11: $BestPhages \leftarrow \emptyset$
12: **for each** $Cluster \in Clusters$ **do**
13: $\quad BestPhages \leftarrow BestPhages \cup SelectBestPhage(Cluster, Matrix)$
14: **end for**
15: $ClusteringMatrix \leftarrow SelectColumns(Matrix, BestPhages)$
16: $NumClusteringPhages \leftarrow NumClusters$
17: $Result \leftarrow ExhaustiveSearch(NumClusteringPhages, MaxBacteria, ClusteringMatrix, limit)$

In this method, a distance matrix is calculated (line 4 in Algorithm 6) by considering the Euclidean distance between pairs of phages [22]. Then, using this distance matrix and Ward's method, agglomerative hierarchical clustering is performed (line 5 in Algorithm 6) [23–25]. Then, the optimal number of clusters must be determined. The objective is to define clusters such that the intracluster variation (i.e., the "Within-cluster Sum of Squares", WSS) is minimized. For this task, a heuristic method, known as the elbow method, is performed (line 6 in Algorithm 6). This method compares different values for the number of clusters, considering the WSS for each one, and selects (as the optimal number of clusters) the one that is the bend in the knee (or elbow) of the curve relating the values for the number of clusters with their corresponding WSS [26–28]. If the number of clusters is less than ϕ, then the number is readjusted to be equal to ϕ (lines 7–9 in Algorithm 6).

After dividing the phages into clusters (line 10 in Algorithm 6), the next step is the selection of only one phage (the most virulent phage) per cluster (lines 11–15 in Algorithm 6). Thus, this method reduces the number of phages to study considerably, and the combinatorial search is also considerably reduced. Then, the method returns the best phage cocktails from size 1 to 7 in an orderly manner as in ExhaustiveSearch (lines 16–17 in Algorithm 6). The primary difference between ClusteringSearch and ExhaustiveSearch is the reduction in the number of phages to study thanks to the clustering process.

3.6. ClusteringPhi

This method is a combination of ExhaustivePhi and ClusteringSearch. The pseudocode of this method is shown in Algorithm 7.

Algorithm 7 Pseudocode of the ClusteringPhi method.

Input: *NumPhages*: original number of phages. *NumBacteria*: original number of bacteria. *MaxBacteria*: number of bacteria that can be lysed. *Matrix*: host range matrix. *limit*: limit of cocktail size.
Output: *Result*: vector with the designed phage cocktail and the bacteria lysed by that cocktail.

1: $Temperature \leftarrow BinMatNest(Matrix)$
2: $Fill \leftarrow 100 \cdot CountInfections(Matrix)/(NumBacteria \cdot NumPhages)$
3: $\phi \leftarrow PhiFormula(NumBacteria, Temperature, Fill)$ (Equation 1)
4: $DistanceMatrix \leftarrow EuclideanDistance(Matrix)$
5: $Dendrogram \leftarrow HierarchicalClustering(DistanceMatrix)$
6: $NumClusters \leftarrow ElbowMethod(Matrix)$
7: **if** $NumClusters < \phi$ **then**
8: $\quad NumClusters \leftarrow \phi$
9: **end if**
10: $Clusters \leftarrow CutDendrogram(Dendrogram, NumClusters)$
11: $BestPhages \leftarrow \emptyset$
12: **for each** $Cluster \in Clusters$ **do**
13: $\quad BestPhages \leftarrow BestPhages \cup SelectBestPhage(Cluster, Matrix)$
14: **end for**
15: $ClusteringMatrix \leftarrow SelectColumns(Matrix, BestPhages)$
16: $NumClusteringPhages \leftarrow NumClusters$
17: **if** $limit < \phi$ **then**
18: $\quad \phi \leftarrow limit$
19: **end if**
20: $Result \leftarrow Search(\phi, NumClusteringPhages, MaxBacteria, ClusteringMatrix)$

The first steps of the method calculate ϕ (lines 1–3 in Algorithm 7), which is the size of the phage cocktail to design. Then, the phage clustering process is performed (lines 4–15 in Algorithm 7), which considerably reduces the number of phages

Table 1
13 phage-bacteria infection networks used in the experimentation.

Reference	Year	Bacteria		Phages		T (°)	Fill (%)	ϕ
		Origin	No.	Origin	No.			
Gunathilaka et al. [29]	2017	Laboratory	12	Sewage	29	10.2	49.4	2
Magare et al. [30]	2017	Air	5	Air	25	31.0	15.2	3
Romero-Suarez et al. [31]	2012	Walnut	16	Walnut	26	12.6	71.6	2
Sajben-Nagy et al. [32]	2012	Laboratory, mushroom	34	Mushroom	16	12.8	36.9	3
Wandro et al. [33]	2019	Human feces	15	Sewage	22	5.8	72.4	1
Vu et al. [34]	2019	Vegetable, seafood, livestock	31	Prophage	42	11.5	21.7	4
Murphy et al. [35]	2013	Dairy	20	Dairy	24	37.3	31.5	4
Brady et al. [36]	2017	Beehive	40	Beehive	57	7.7	40.7	3
Jackel et al. [37]	2017	Laboratory	113	Prophage	19	11.6	6.3	7
Petsong et al. [38]	2019	Livestock	47	Livestock	36	10.0	14.7	5
Gencay et al. [39]	2019	Pork meat	72	Prophage	41	13.7	22.2	5
Korf et al. [40]	2019	Clinical, poultry	64	Poultry, sewage	50	17.3	18.5	5
Mathieu et al. [41]	2020	Feces	75	Feces	167	2.8	6.1	5

to study. Its steps are as follows: calculating the distance matrix (line 4), performing agglomerative hierarchical clustering (line 5), calculating the optimal number of clusters (lines 6–9), dividing the phages into clusters (line 10), and selecting the best phage (the most virulent one) per cluster (lines 11–15).

Then, calling the Search function (Algorithm 4), this method returns the best phage cocktail with size ϕ, considering the phages selected in the clustering process (lines 16–19 in Algorithm 7).

4. Results and discussion

In this section, the four methods to design phage cocktails are evaluated. They are compared from two viewpoints: quality of the phage cocktails and runtime. The following subsections detail the experimentation performed.

4.1. Datasets

The four methods were evaluated using 13 empirical phage-bacteria infection networks. These PBINs are described in Table 1 and were chosen to have variability in terms of nestedness temperature (T), fill, ϕ, origin of the bacteria and phages, and number of bacteria and phages. Thus, many diverse solution scenarios are reproduced when designing phage cocktails. Nestedness temperature and fill indicate how the corresponding host range matrix is, while ϕ and the number of phages and bacteria can be used as estimators of how long the execution might take. Most PBINs are from the last few years, although there are also some from as long ago as 2012.

4.2. Experimental settings

The software was programmed in R using RStudio. More specifically, 4.0.4 R version and 1.4.1106 RStudio version. Experiments were performed on a computer with a CPU Intel Core i7-8700 at 3.20 GHz and 32 GB of RAM under the Windows 10 Pro operating system. The R package can be freely downloaded from CRAN (https://cran.r-project.org/package=PhageCocktail).

4.3. Runtime evaluation

Table 2 shows the median and quartile deviation of the runtime (in milliseconds) used by the four methods when designing the phage cocktail for each phage-bacteria infection network. To assure the statistical reliability of the results, 31 independent runs were performed for every experiment.

Table 2
Runtime evaluation (in milliseconds, $median_{\pm quartile\_deviation}$) of the four methods for designing the phage cocktails of the 13 phage-bacteria infection networks.

Reference	ExhaustiveSearch	ExhaustivePhi
Gunathilaka et al. [29]	$0.020_{\pm 0.0035}$	$0.035_{\pm 0.0035}$
Magare et al. [30]	$0.201_{\pm 0.0111}$	$0.036_{\pm 0.0071}$
Romero-Suarez et al. [31]	$0.109_{\pm 0.0075}$	$0.101_{\pm 0.0196}$
Sajben-Nagy et al. [32]	$0.182_{\pm 0.0093}$	$0.193_{\pm 0.0233}$
Wandro et al. [33]	$0.021_{\pm 0.0018}$	$0.034_{\pm 0.0045}$
Vu et al. [34]	$10.647_{\pm 1.4220}$	$8.036_{\pm 0.3864}$
Murphy et al. [35]	$65.330_{\pm 3.1075}$	$42.545_{\pm 1.1298}$
Brady et al. [36]	$148.767_{\pm 2.4866}$	$130.846_{\pm 2.8624}$
Jackel et al. [37]	$683.218_{\pm 5.5714}$	$314.712_{\pm 2.0703}$
Petsong et al. [38]	$7584.093_{\pm 41.1329}$	$6715.142_{\pm 69.6707}$
Gencay et al. [39]	$35425.187_{\pm 108.8692}$	$30651.298_{\pm 150.0567}$
Korf et al. [40]	$90736.713_{\pm 218.8998}$	$83227.954_{\pm 148.0530}$
Mathieu et al. [41]	$7083198.000_{\pm 170974.4162}$	$6677424.000_{\pm 81327.3691}$
Reference	**ClusteringSearch**	**ClusteringPhi**
Gunathilaka et al. [29]	$0.017_{\pm 0.0023}$	$0.034_{\pm 0.0022}$
Magare et al. [30]	$0.059_{\pm 0.0063}$	$0.036_{\pm 0.0070}$
Romero-Suarez et al. [31]	$0.044_{\pm 0.0025}$	$0.039_{\pm 0.0022}$
Sajben-Nagy et al. [32]	$0.060_{\pm 0.0045}$	$0.051_{\pm 0.0064}$
Wandro et al. [33]	$0.019_{\pm 0.0023}$	$0.028_{\pm 0.0044}$
Vu et al. [34]	$0.100_{\pm 0.0112}$	$0.043_{\pm 0.0025}$
Murphy et al. [35]	$0.205_{\pm 0.0112}$	$0.041_{\pm 0.0033}$
Brady et al. [36]	$0.086_{\pm 0.0045}$	$0.045_{\pm 0.0027}$
Jackel et al. [37]	$2.773_{\pm 0.0735}$	$0.080_{\pm 0.0071}$
Petsong et al. [38]	$0.330_{\pm 0.0092}$	$0.050_{\pm 0.0040}$
Gencay et al. [39]	$0.682_{\pm 0.0303}$	$0.067_{\pm 0.0055}$
Korf et al. [40]	$0.636_{\pm 0.0180}$	$0.067_{\pm 0.0014}$
Mathieu et al. [41]	$0.693_{\pm 0.0071}$	$0.072_{\pm 0.0030}$

In the case of high-dimensional PBINs (such as Brady et al. [36], Jackel et al. [37], Petsong et al. [38], Gencay et al. [39], Korf et al. [40], and Mathieu et al. [41]), there is an important difference in runtime between using Exhaustive* or Clustering* methods because the number of combinations that the program has to evaluate for designing the phage cocktail is higher using Exhaustive* than Clustering* methods. Exhaustive* methods look for the phage cocktail that lyses the highest number of bacteria using all useful phages, while Clustering* methods only use a small set of phages selected after the clustering process. Thus, Clustering* methods always provide immediate results (generally, in less than one millisecond).

Some differences were also observed between executing ExhaustiveSearch or ExhaustivePhi when ϕ and the number of

Table 3
Evolution of the runtime (in milliseconds, $median_{\pm quartile\_deviation}$) for ExhaustiveSearch and ClusteringSearch when designing phage cocktails of increasing size for the high-dimensional PBINs.

Reference	Cocktail size	ExhaustiveSearch		ClusteringSearch runtime (ms)
		Runtime (ms)	Δ (times)	
Brady et al. [36]	1	$0.101_{\pm 0.0037}$	–	$0.020_{\pm 0.0013}$
	2	$13.940_{\pm 1.5927}$	138.020	$0.039_{\pm 0.0022}$
	3	$134.258_{\pm 3.1679}$	9.631	$0.027_{\pm 0.0015}$
Jackel et al. [37]	1	$0.044_{\pm 0.0004}$	–	$0.029_{\pm 0.0016}$
	2	$1.171_{\pm 0.0249}$	26.614	$0.250_{\pm 0.0046}$
	3	$8.813_{\pm 0.3112}$	7.526	$0.680_{\pm 0.0187}$
	4	$38.406_{\pm 0.7602}$	4.358	$0.933_{\pm 0.0197}$
	5	$107.109_{\pm 1.8927}$	2.789	$0.727_{\pm 0.0160}$
	6	$209.389_{\pm 2.1920}$	1.955	$0.316_{\pm 0.0189}$
	7	$317.326_{\pm 2.9628}$	1.515	$0.064_{\pm 0.0031}$
Petsong et al. [38]	1	$0.055_{\pm 0.0009}$	–	$0.020_{\pm 0.0015}$
	2	$3.691_{\pm 0.1145}$	67.109	$0.074_{\pm 0.0022}$
	3	$77.438_{\pm 1.3427}$	20.980	$0.117_{\pm 0.0020}$
	4	$815.534_{\pm 12.0949}$	10.531	$0.085_{\pm 0.0022}$
	5	$6684.574_{\pm 56.7644}$	8.197	$0.030_{\pm 0.0013}$
Gencay et al. [39]	1	$0.111_{\pm 0.0014}$	–	$0.026_{\pm 0.0020}$
	2	$13.957_{\pm 1.3440}$	125.739	$0.155_{\pm 0.0051}$
	3	$270.487_{\pm 2.6962}$	19.380	$0.259_{\pm 0.0078}$
	4	$3405.080_{\pm 14.8109}$	12.589	$0.182_{\pm 0.0062}$
	5	$31724.127_{\pm 105.5093}$	9.317	$0.055_{\pm 0.0016}$
Korf et al. [40]	1	$0.124_{\pm 0.0026}$	–	$0.026_{\pm 0.0014}$
	2	$18.222_{\pm 2.0649}$	146.952	$0.147_{\pm 0.0054}$
	3	$454.034_{\pm 5.3054}$	24.917	$0.239_{\pm 0.0066}$
	4	$7125.837_{\pm 36.5867}$	15.695	$0.171_{\pm 0.0048}$
	5	$83122.426_{\pm 187.3052}$	11.665	$0.052_{\pm 0.0016}$
Mathieu et al. [41]	1	$0.321_{\pm 0.0454}$	–	$0.025_{\pm 0.0012}$
	2	$104.722_{\pm 3.6473}$	326.237	$0.162_{\pm 0.0047}$
	3	$6279.769_{\pm 107.1102}$	59.966	$0.272_{\pm 0.0039}$
	4	$240852.769_{\pm 8132.6212}$	38.354	$0.189_{\pm 0.0051}$
	5	$6831400.000_{\pm 240852.7694}$	28.363	$0.057_{\pm 0.0019}$

phages and bacteria were high. ExhaustivePhi is faster because it only returns the phage cocktail of size ϕ, while ExhaustiveSearch returns all the phage cocktails from size 1 to size 7.

Considering that a limit value can be configured, this can be used to avoid performing high cocktail size searches that could have a high computational cost. Another application of the limit value is to restrict the output in those cases in which users do not want a higher phage cocktail size. Table 3 shows the evolution of the runtime in the high-dimensional PBINs when different cocktail sizes are used. As shown, the impact in ClusteringSearch is low because the number of phages to study is reduced. However, in the case of ExhaustiveSearch, the impact is high (see column "Δ (times)" in Table 3). Regardless, the exact impact depends on the features of each PBIN, primarily its number of phages and bacteria. For example, the increment in runtime is higher in the PBIN from Mathieu et al. [41] than in the PBIN from Jackel et al. [37]. Also, due to the combinatorial procedure itself, this increment is smaller step by step (i.e., as the cocktail size increases).

4.4. Quality of the phage cocktails

Table 4 shows the size of the phage cocktails generated by the four methods, the bacteria lysed by these cocktails, and the maximum number of bacteria that could be lysed. ExhaustiveSearch is the best option because the user will obtain the best possible combination of phages (i.e., the best phage cocktail), which is the one that lyses more bacteria with the least number of phages. Therefore, when users want to be sure about getting the best result, ExhaustiveSearch must be executed.

When the *Search and *Phi methods are compared, the estimation given by ϕ is not optimal in all cases. In some cases, the same number of bacteria can be lysed with a smaller phage cocktail; this result occurs in 3 out of 13 cases in these experiments, in Gunathilaka et al. [29], Sajben-Nagy et al. [32], and Vu et al. [34]. There are PBINs whose ϕ number is higher than the maximum phage cocktail size required. Therefore, users must determine if they prefer a faster solution that is likely to be the best one (*Phi method) or the best one with absolute reliability (*Search method). Conversely, *Search methods provide the list of lysed bacteria and used phages as the phage cocktail size increases, and this information can be useful to have a better understanding and analysis.

When the Exhaustive* and Clustering* methods are compared, Table 4 shows that the Exhaustive* methods obtain better results (i.e., lyse more bacteria) in 6 out of 13 phage-bacteria infection networks (i.e., in all the high-dimensional PBINs). This comparison is described in more detail in Table 5, where ExhaustiveSearch and ClusteringSearch are compared when the phage cocktail size increases. In most cases (4 out of 6) of high-dimensional PBINs, ClusteringSearch cannot improve its solution, even by increasing the phage cocktail size, as in Brady et al. [36], Jackel et al. [37], Petsong et al. [38], and Gencay et al. [39] (see Table 5). These results indicate that the clustering process can remove phages that can later be useful for lysing new bacteria as the phage cocktail size increases. In the remaining cases (2 out of 6, PBINs from Korf et al. [40] and Mathieu et al. [41]), the number of lysed bacteria increases with the phage cocktail size, but the results are worse than in ExhaustiveSearch, even with a phage cocktail of size 2 (see Table 5). Conversely, ExhaustiveSearch always improves the result as the phage cocktail size is increased, and its result is always the best possible result. Thus, users must select between an immediate solution with Clustering* methods and a possibly better solution with Exhaustive* methods.

4.5. Additional insights

After analyzing the advantages and disadvantages of using each method, this subsection tries to provide a broader vision of the additional knowledge that is possible to achieve with this R package.

Table 4
Bacteria lysed by the phage cocktails designed with the four methods for the 13 phage-bacteria infection networks.

Reference	Cocktail size		Bacteria lysed	Maximum (%)
	ExhaustiveSearch	ExhaustivePhi		
Gunathilaka et al. [29]	1	2	12 (100.00%)	100.00%
Magare et al. [30]	3	3	4 (80.00%)	80.00%
Romero-Suarez et al. [31]	2	2	16 (100.00%)	100.00%
Sajben-Nagy et al. [32]	2	3	18 (52.94%)	52.94%
Wandro et al. [33]	1	1	14 (93.33%)	93.33%
Vu et al. [34]	3	4	20 (64.52%)	64.52%
Murphy et al. [35]	4	4	20 (100.00%)	100.00%
Brady et al. [36]	3	3	39 (97.50%)	97.50%
Jackel et al. [37]	7	7	52 (46.02%)	46.90%
Petsong et al. [38]	5	5	26 (55.32%)	59.57%
Gencay et al. [39]	5	5	64 (88.89%)	94.44%
Korf et al. [40]	5	5	56 (87.50%)	98.44%
Mathieu et al. [41]	5	5	70 (93.33%)	93.33%
	ClusteringSearch	ClusteringPhi		
Gunathilaka et al. [29]	1	2	12 (100.00%)	100.00%
Magare et al. [30]	3	3	4 (80.00%)	80.00%
Romero-Suarez et al. [31]	2	2	16 (100.00%)	100.00%
Sajben-Nagy et al. [32]	2	3	18 (52.94%)	52.94%
Wandro et al. [33]	1	1	14 (93.33%)	93.33%
Vu et al. [34]	3	4	20 (64.52%)	64.52%
Murphy et al. [35]	4	4	20 (100.00%)	100.00%
Brady et al. [36]	3	3	36 (90.00%)	97.50%
Jackel et al. [37]	7	7	50 (44.25%)	46.90%
Petsong et al. [38]	5	5	20 (42.55%)	59.57%
Gencay et al. [39]	5	5	49 (68.06%)	94.44%
Korf et al. [40]	5	5	53 (82.81%)	98.44%
Mathieu et al. [41]	5	5	62 (82.67%)	93.33%

Table 5
Evolution of the bacteria lysed for ExhaustiveSearch and ClusteringSearch when designing phage cocktails of increasing size for the high-dimensional PBINs.

Reference	Cocktail size	Bacteria lysed	
		ExhaustiveSearch	ClusteringSearch
Brady et al. [36]	1	36 (90.00%)	36 (90.00%)
	2	37 (92.50%)	36 (90.00%)
	3	39 (97.50%)	36 (90.00%)
Jackel et al. [37]	1	32 (28.32%)	32 (28.32%)
	2	40 (35.40%)	40 (35.40%)
	3	46 (40.71%)	46 (40.71%)
	4	48 (42.48%)	48 (42.48%)
	5	50 (44.25%)	50 (44.25%)
	6	51 (45.13%)	50 (44.25%)
	7	52 (46.02%)	50 (44.25%)
Petsong et al. [38]	1	15 (31.91%)	15 (31.91%)
	2	19 (40.43%)	18 (38.30%)
	3	22 (46.81%)	19 (40.43%)
	4	24 (51.06%)	20 (42.55%)
	5	26 (55.32%)	20 (42.55%)
Gencay et al. [39]	1	40 (55.56%)	40 (55.56%)
	2	49 (68.06%)	45 (62.50%)
	3	56 (77.78%)	48 (66.67%)
	4	61 (84.72%)	49 (68.06%)
	5	64 (88.89%)	49 (68.06%)
Korf et al. [40]	1	34 (53.13%)	34 (53.13%)
	2	44 (68.75%)	43 (67.19%)
	3	50 (78.13%)	48 (75.00%)
	4	54 (84.38%)	52 (81.25%)
	5	56 (87.50%)	53 (82.81%)
Mathieu et al. [41]	1	50 (66.67%)	50 (66.67%)
	2	61 (81.33%)	57 (76.00%)
	3	67 (89.33%)	60 (80.00%)
	4	69 (92.00%)	61 (81.33%)
	5	70 (93.33%)	62 (82.67%)

For example, scientists could be interested in comparing the number of lysed bacteria among different sizes of phage cocktail and choosing the best one according to their preferences.

Bacteria can grow exponentially with a generation time as short as 20 min. Thus, the addition of a single bacterial strain lysed in a community by increasing the cocktail size with one phage could be relevant for biocontrol in both the medicine and food industries. For example, when the new bacterium is highly pathogenic and resistant to antibiotics, or it is the primary factor causing food spoilage. In both cases, although the percentage of lysed bacteria is nearly identical, the phage cocktail that lyses the new bacterium would be chosen.

Conversely, phage specificity would ensure that the beneficial microbiota remains intact during phage therapy because they are innocuous to eukaryotic and all prokaryotic cells outside their host range. This fact allows highly specific cocktails to be tailor for suit fermented foods such as dairy products that require lactic acid bacteria and probiotic strains. Smaller phage cocktails could entail lower production costs, and thus, their implementation would be more profitable.

In addition, experimental results verify that the same number of bacteria lysed in ClusteringSearch for a certain cocktail size can be lysed with a smaller number of phages in ExhaustiveSearch. Therefore, although ClusteringSearch is faster (generally less than one millisecond), ExhaustiveSearch provides scientists the guarantee of obtaining the maximum economic and biological yield.

Conversely, it is relevant to consider the list of bacteria that are not lysed by any phage to search for phages infecting these bacteria. The list of non-useful phages should be the subject of further research about their inefficacy, or they can be used to investigate the coevolution between bacteria and phages, yielding more virulent phages and more resistant bacteria. Therefore, both lists provide users with additional knowledge.

5. Conclusions

This article presents a new R package called PhageCocktail. The goal of this software is to automatically design phage cocktails that are appropriate for the biocontrol of bacteria. The best phage cock-

tail is the one that lyses the most bacteria with the fewest phages. The package includes four different methods for designing phage cocktails. The four methods have been explained in detail, and each method has its own advantages. To determine the efficiency and runtime of these methods, 13 empirical phage-bacteria infection networks were investigated experimentally.

The experiments show that the best phage cocktail is always found by ExhaustiveSearch. However, its runtime can be high (approximately 1–2 h) when PBIN has a large number of phages (e.g., 167 phages in these experiments for PBIN from Mathieu et al. [41]). For the rest of the PBINs, the runtime is approximately 90 s or lower. Conversely, ClusteringSearch always has a very reduced runtime (generally less than one millisecond) and obtains good phage cocktails but sometimes cannot find the best phage cocktail; this result occurs in 6 out of 13 cases in these experiments.

The other two methods are ExhaustivePhi and ClusteringPhi. In these methods, the size of the phage cocktail is estimated by calculating the ϕ number (Eq. (1)). Therefore, both methods only focus on phage cocktails of size ϕ. The experiments show that ϕ performs a good estimation but is not the most accurate. For example, in some cases, the same number of bacteria can be lysed with a smaller phage cocktail; this result occurs in 3 out of 13 cases in these experiments. However, it is also true that only focusing on size ϕ improves the runtime.

Thus, users can choose among the four methods based on their preferences in speed and accuracy of results. The parameter limit is also interesting because it can avoid a high runtime or phage cocktails that are larger than desired.

Manually designing phage cocktails is tedious and time-consuming, and because high-throughput screening is gaining popularity [42], PhageCocktail will become increasingly useful. This R package provides scientists and other users with the tools to obtain appropriate phage cocktails depending on their requirements and preferences. Also, the package provides the number of phages and bacteria and their corresponding names, and relevant additional knowledge, including a list of non-useful phages, bacteria that cannot be lysed, and comparisons among phage cocktails of different sizes, which are important when users want to interpret and analyze in detail the data in this field.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was partially funded by Junta de Extremadura (Spain) and ERDF (European Regional Development Fund, EU), grant GR18090 provided to the research group TIC015. This work was also partially funded by MCIU (Ministry of Science, Innovation and Universities, Spain), AEI (State Research Agency, Spain), and ERDF (European Regional Development Fund, EU) under the contract PID2019-107299GB-I00/AEI/10.13039/501100011033 (Multi-HPC-Bio project).

References

[1] M.I. Hutchings, A.W. Truman, B. Wilkinson, Antibiotics: past, present and future, Curr. Opin. Microbiol. 51 (2019) 72–80, doi:10.1016/j.mib.2019.10.008.
[2] K. Thiel, Old dogma, new tricks–21st century phage therapy, Nat. Biotechnol. 22 (1) (2004) 31–36, doi:10.1038/nbt0104-31.
[3] S.T. Abedon, P. García, P. Mullany, R. Aminov, Editorial: phage therapy: past, present and future, Front. Microbiol. 8 (2017) 981, doi:10.3389/fmicb.2017.00981.
[4] M.C. Rea, A. Dobson, O. O'Sullivan, F. Crispie, F. Fouhy, P.D. Cotter, F. Shanahan, B. Kiely, C. Hill, R.P. Ross, Effect of broad- and narrow-spectrum antimicrobials on Clostridium difficile and microbial diversity in a model of the distal colon, Proc. Natl. Acad. Sci. 108 (Supplement 1) (2011) 4639–4644, doi:10.1073/pnas.1001224107.
[5] W.C. Summers, The strange history of phage therapy, Bacteriophage 2 (2) (2012) 130–133, doi:10.4161/bact.20757.
[6] C. Lepore, L. Silver, U. Theuretzbacher, J. Thomas, D. Visi, The small-molecule antibiotics pipeline: 2014–2018, Nat. Rev. Drug Discov. 18 (10) (2019) 739–740, doi:10.1038/d41573-019-00130-8.
[7] B.K. Chan, S.T. Abedon, C. Loc-Carrillo, Phage cocktails and the future of phage therapy, Future Microbiol. 8 (6) (2013) 769–783, doi:10.2217/fmb.13.47.
[8] C.A. Guerrero-Bustamante, R.M. Dedrick, R.A. Garlena, D.A. Russell, G.F. Hatfull, Toward a phage cocktail for tuberculosis: susceptibility and tuberculocidal action of mycobacteriophages against diverse *Mycobacterium tuberculosis* strains, MBio 12 (3) (2021), doi:10.1128/mBio.00973-21. E00973–21
[9] M. Li, R.Y.K. Chang, Y. Lin, S. Morales, E. Kutter, H.K. Chan, Phage cocktail powder for *Pseudomonas aeruginosa*respiratory infections, Int. J. Pharm. 596 (2021) 120200, doi:10.1016/j.ijpharm.2021.120200.
[10] H. Park, J. Kim, M. Kim, Y. Park, S. Ryu, Development of new strategy combining heat treatment and phage cocktail for post-contamination prevention, Food Res. Int. 145 (2021) 110415, doi:10.1016/j.foodres.2021.110415.
[11] E.N. Bugaeva, M.V. Voronina, D.M. Vasiliev, A.A. Lukianova, N.N. Landyshev, A.N. Ignatov, K.A. Miroshnikov, Use of a specific phage cocktail for soft rot control on ware potatoes: a case study, Viruses 13 (6) (2021) 1095, doi:10.3390/v13061095.
[12] K. Shahin, L. Zhang, H. Bao, A. Hedayatkhah, A. Soleimani-Delfan, M. Komijani, T. He, M. Barazandeh, M. Mansoorianfar, M. Bouzari, R. Wang, An in-vitro study on a novel six-phage cocktail against multi-drug resistant-ESBL Shigella in aquatic environment, Lett. Appl. Microbiol. 72 (3) (2021) 231–237, doi:10.1111/lam.13418.
[13] F.H. Coutinho, A. Zaragoza-Solas, M. López-Pérez, J. Barylski, A. Zielezinski, B.E. Dutilh, R. Edwards, F. Rodriguez-Valera, RaFAH: host prediction for viruses of bacteria and archaea based on protein content, Patterns 2 (2021) 100274, doi:10.1016/j.patter.2021.100274.
[14] Q. Lamy-Besnier, B. Brancotte, H. Ménager, L. Debarbieux, Viral host range database, an online tool for recording, analyzing and disseminating virus-host interactions, Bioinformatics 37 (17) (2021) 2798–2801, doi:10.1093/bioinformatics/btab070.
[15] F. Molina, A. Simancas, R. Tabla, A. Gómez, I. Roa, J.E. Rebollo, Diversity and local coadaptation of *Escherichia coli* and coliphages from small ruminants, Front. Microbiol. 11 (2020) 2513, doi:10.3389/fmicb.2020.564522.
[16] T.D.S. Sutton, C. Hill, Gut bacteriophage: current understanding and challenges, Front. Endocrinol. 10 (2019) 784, doi:10.3389/fendo.2019.00784.
[17] J.S. Weitz, T. Poisot, J.R. Meyer, C.O. Flores, S. Valverde, M.B. Sullivan, M.E. Hochberg, Phage–bacteria infection networks, Trends Microbiol. 21 (2) (2013) 82–91, doi:10.1016/j.tim.2012.11.003.
[18] M.A. Fortuna, M.A. Barbour, L. Zaman, A.R. Hall, A. Buckling, J. Bascompte, Coevolutionary dynamics shape the structure of bacteria-phage infection networks, Evolution 73 (5) (2019) 1001–1011, doi:10.1111/evo.13731.
[19] F. Molina, A. Simancas, M. Ramírez, R. Tabla, I. Roa, J.E. Rebollo, A new pipeline for designing phage cocktails based on phage-bacteria infection networks, Front. Microbiol. 12 (2021) 149, doi:10.3389/fmicb.2021.564532.
[20] W. Atmar, B.D. Patterson, The nestedness temperature calculator: a visual basic program, including 294 presence-absence matrices, Bull. Ecol. Soc. Am. 78 (1) (1997) 63–65, doi:10.2307/20168111.
[21] M.A. Rodríguez-Gironés, L. Santamaría, A new algorithm to calculate the nestedness temperature of presence–absence matrices, J. Biogeogr. 33 (5) (2006) 924–935, doi:10.1111/j.1365-2699.2006.01444.x.
[22] S.C. Johnson, Hierarchical clustering schemes, Psychometrika 32 (3) (1967) 241–254, doi:10.1007/BF02289588.
[23] F. Murtagh, P. Legendre, Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? J. Classif. 31 (3) (2014) 274–295, doi:10.1007/s00357-014-9161-z.
[24] T. Strauss, M.J. von Maltitz, Generalising Ward's method for use with Manhattan distances, PLoS One 12 (1) (2017) e0168288, doi:10.1371/journal.pone.0168288.
[25] M.J. Barber, Modularity and community detection in bipartite networks, Phys. Rev. E 76 (6) (2007) 066102, doi:10.1103/PhysRevE.76.066102.
[26] A. Kassambara, F. Mundt, Factoextra: extract and visualize the results of multivariate data analyses, R Package 1 (5) (2020) 337–354.
[27] J. French, M. Meysami, Smerc: statistical methods for regional counts, R Package 1 (1) (2021).
[28] T.M. Kodinariya, P.R. Makwana, Review on determining number of cluster in *k*-means clustering, Int. J. Adv. Res. Comput. Sci. Manag. Stud. 1 (6) (2013) 90–95.
[29] G.U. Gunathilaka, V. Tahlan, A.I. Mafiz, M. Polur, Y. Zhang, Phages in urban wastewater have the potential to disseminate antibiotic resistance, Int. J. Antimicrob. Agents 50 (5) (2017) 678–683, doi:10.1016/j.ijantimicag.2017.08.013.
[30] B. Magare, A. Nair, K. Khairnar, Isolation of bacteriophages from air using vacuum filtration technique: an improved and novel method, J. Appl. Microbiol. 123 (4) (2017) 896–902, doi:10.1111/jam.13551.
[31] S. Romero-Suarez, B. Jordan, J.A. Heinemann, Isolation and characterization of bacteriophages infecting *Xanthomonas arboricola* pv. juglandis, the causal agent of walnut blight disease, World J. Microbiol. Biotechnol. 28 (5) (2012) 1917–1927, doi:10.1007/s11274-011-0992-z.
[32] E. Sajben-Nagy, G. Maróti, L. Kredics, B. Horváth, A. Párducz, C. Vágvölgyi, L. Manczinger, Isolation of new *Pseudomonas tolaasii*bacteriophages and ge-

nomic investigation of the lytic phage BF7, FEMS Microbiol. Lett. 332 (2) (2012) 162–169, doi:10.1111/j.1574-6968.2012.02592.x.
[33] S. Wandro, A. Oliver, T. Gallagher, C. Weihe, W. England, J.B.H. Martiny, K. Whiteson, Predictable molecular adaptation of coevolving *Enterococcus faecium* and lytic phage EfV12-phi1, Front. Microbiol. 9 (2019) 3192, doi:10.3389/fmicb.2018.03192.
[34] H.T.K. Vu, S. Benjakul, K. Vongkamjan, Characterization of *Listeria prophages* in lysogenic isolates from foods and food processing environments, PLoS One 14 (4) (2019) e0214641, doi:10.1371/journal.pone.0214641.
[35] J. Murphy, B. Royer, J. Mahony, L. Hoyles, K. Heller, H. Neve, M. Bonestroo, A. Nauta, D. van Sinderen, Biodiversity of lactococcal bacteriophages isolated from 3 Gouda-type cheese-producing plants, J. Dairy Sci. 96 (8) (2013) 4945–4957, doi:10.3168/jds.2013-6748.
[36] T.S. Brady, B.D. Merrill, J.A. Hilton, A.M. Payne, M.B. Stephenson, S. Hope, Bacteriophages as an alternative to conventional antibiotic use for the prevention or treatment of *Paenibacillus larvae* in honeybee hives, J. Invertebr. Pathol. 150 (2017) 94–100, doi:10.1016/j.jip.2017.09.010.
[37] C. Jäckel, S. Hertwig, H.C. Scholz, K. Nöckler, J. Reetz, J.A. Hammerl, Prevalence, host range, and comparative genomic analysis of temperate *Ochrobactrum phages*, Front. Microbiol. 8 (2017) 1207, doi:10.3389/fmicb.2017.01207.
[38] K. Petsong, S. Benjakul, S. Chaturongakul, A.I.M. Switt, K. Vongkamjan, Lysis profiles of Salmonella phages on Salmonella isolates from various sources and efficiency of a phage cocktail against *S. enteritidis* and *S. typhimurium*, Microorganisms 7 (4) (2019) 100, doi:10.3390/microorganisms7040100.
[39] Y.E. Gencay, M. Gambino, T.F. Prüssing, L. Brøndsted, The genera of bacteriophages and their receptors are the major determinants of host range, Environ. Microbiol. 21 (6) (2019) 2095–2111, doi:10.1111/1462-2920.14597.
[40] I.H.E. Korf, J.P. Meier-Kolthoff, E.M. Adriaenssens, A.M. Kropinski, M. Nimtz, M. Rohde, M.J. van Raaij, J. Wittmann, Still something to discover: novel insights into *Escherichia coli* phage diversity and taxonomy, Viruses 11 (5) (2019) 454, doi:10.3390/v11050454.
[41] A. Mathieu, M. Dion, L. Deng, D. Tremblay, E. Moncaut, S.A. Shah, J. Stokholm, K.A. Krogfelt, S. Schjørring, H. Bisgaard, D.S. Nielsen, S. Moineau, M.A. Petit, Virulent coliphages in 1-year-old children fecal samples are fewer, but more infectious than temperate coliphages, Nat. Commun. 11 (1) (2020) 378, doi:10.1038/s41467-019-14042-z.
[42] F.L.G. Altamirano, J.J. Barr, Unlocking the next generation of phage therapy: the key is in the receptors, Curr. Opin. Biotechnol. 68 (2021) 115–123, doi:10.1016/j.copbio.2020.10.002.