bioRxiv preprint first posted online Jun. 15, 2015; doi: http://dx.doi.org/10.1101/020859. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC-ND 4.0 International license.

Automated Contamination Detection in Single-Cell Sequencing

Markus Lux, Barbara Hammer, Alexander Sczyrba
Bielefeld University
{mlux,bhammer,asczyrba}@techfak.uni-bielefeld.de

Abstract

Novel methods for the sequencing of single-cell DNA offer tremendous opportunities. However, many techniques are still in their infancy and a major obstacle is given by sample contamination with foreign DNA. In this contribution, we present a pipeline that allows for fast, automated detection of contaminated samples by the use of modern machine learning methods. First, a vectorial representation of the genomic data is obtained using oligonucleotide signatures. Using non-linear subspace projections, data is transformed to be suitable for automatic clustering. This allows for the detection of one vs. more genomes (clusters) in a sample. As clustering is an ill-posed problem, the pipeline relies on a thorough choice of all involved methods and parameters. We give an overview of the problem and evaluate techniques suitable for this task.

I. Introduction

Todays next-generation sequencing technologies enable the analysis of large amounts of genetic information. A number of exciting data sources are given by single-cell sequencing (SCS). Named *Method of the Year 2013* [1], it will be beneficial in many domains of research, most notably medicine and the analysis of disease pathways. Often, on a single cell level, the pathology of complex diseases is very heterogeneous [8]. In different types of cancer, for example, arises the question why certain neighboring cells are malignant while others are not. SCS is able to identify such differences in a high resolution, enabling the analysis of underlying causes and dynamics in great detail, which in turn can be the groundwork for specific treatment.

However, existing SCS technologies are still in their infancy and in order to gain tools with economic relevance, a number of problems need to be resolved. A major potential for development can be seen in DNA isolation. In SCS, samples are taken using patch pipettes or nanotubes. These methods come with the disadvantage that also foreign DNA such as from within the sample (viruses, bacteriophages), or from the laboratory environment can easily be captured [6]. Much effort has been invested in engineering devices for cell isolation and amplification steps that minimize the contamination caused by the surrounding sequencing setup [6]. Still, such measures only decrease the probability for contamination and remaining foreign DNA is detected by tedious manual screening. Additionally, the fast growing amount of data makes this step consuming a lot of time. Therefore, there is a strong need for data analysis techniques that can aid automatic post-sequencing contamination detection. Some species can be detected using supervised methods (i.e. based on sequence similarity to known taxa from databases) and fast classification tools exist [2, 31, 21].

The majority of species is unknown [22] and thus cannot be detected by such methods. Hence, an, taxonomy-free analysis is required [20]. Here, one particularly promising line of research relies on modern techniques from machine learning, specifically clustering techniques based on k_l-mer frequencies that already found early applications in metagenomic binning [17]. From the perspective of computational intelligence, contamination detection in SCS is very similar to metagenomic binning. Both metagenomic and SCS samples can be represented as a set of high-dimensional data points using oligonucleotide frequencies. Binning and also contamination detection then correspond to the problem to reliably detect clusters in a high dimensional data space.

In this context, quite a few challenges arise: To circumvent negative side effects in such high dimensional spaces and to enable human expert inspection, it is crucial to use appropriate subspace embeddings to transform the data into an easily visualizable representation, i.e. two or three dimensions. Another challenge consists in the automatic determination of the number of clusters and its cluster validity, a deep and crucial question in the context of clustering [28, 14]. In contrast to metagenomics, in SCS one is concerned with much less genomes in a given sample, significantly reducing the complexity of the problem. Also, contamination detection in SCS corresponds to the problem to discriminate between one or more clusters (genomes). This distinction is important since it heavily reduces the set of applicable clustering methods: The majority of methods for estimating the number of clusters rely on cluster-specific measures such as internal validity measures [18]. Since they are not defined for only one cluster, a distinctive null model for unimodal data is required.

In this contribution, we give and overview on the theoretical foundations and methodological considerations of an automated contamination detection pipeline. First, we will discuss the suitability of a particular non-linear dimen-

bioRxiv preprint first posted online Jun. 15, 2015; doi: http://dx.doi.org/10.1101/020859. The copyright holder for this preprint
(which was not peer-reviewed) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity.
It is made available under a CC-BY-NC-ND 4.0 International license.

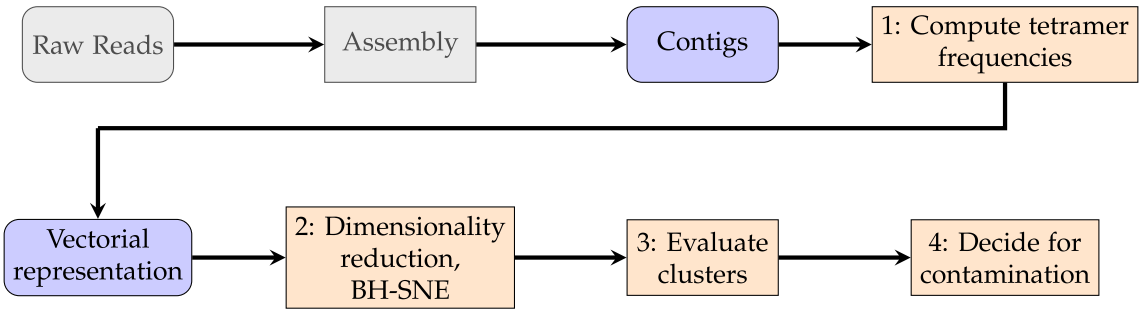
Raw Reads
Assembly
Contigs
1: Compute tetramer frequencies
Vectorial representation
2: Dimensionality reduction, BH-SNE
3: Evaluate clusters
4: Decide for contamination

Figure 1: *Contamination detection pipeline.*

sion reduction method. Then, the main focus will be put on evaluating clustering methods and discussing their suitability with respect to different criteria. The outcome of all involved methods depend on a number of parameters. Here, we will suggest strategies for choosing an optimal parameter set. Finally, we show how the inclusion of other, possibly supervised, methods can improve detection accuracy, resulting in a pipeline that can detect contamination with high rate.

II. Methods

A contamination detection tool will utilize a number of subsequent steps which are outlined in Figure 1. Starting with raw reads from the sequencing process, they are assembled into longer oligonucleotides. Using k_l-mer frequencies, a high-dimensional vectorial representation is obtained. After dimensionality reduction, it is the task of evaluating the number of clusters, specifically determining whether there is one or more cluster, resulting in a final decision for contamination. In the following, we will briefly describe the involved methods for frequency computation, dimensionality reduction, and clustering.

I. Vectorial representation

It is common practice to transfer sequential DNA data into vectors by using signatures of small chunks of DNA [24]. A window of width w is fixed and subsequently shifted over the sequence with step Δw. For each shift, the underlying k_l-mers frequencies are evaluated. This results in one 4^{k_l}-dimensional data point per shift, accounting for the 4 nucleotide bases. Exemplary, taking $k_l = 4$ would result in 256 dimensions, however by accounting for reverse complements, it can be reduced to 136 dimensions.

The choice of window parameters has big influence on the resulting representation. Here, choosing a large window width, capturing genome-specific, rather than gene-specific information will result in less noise [19]. However, a small number of data points is disadvantageous for clustering, such that is has to be taken care to choose w not too large. Using a given number of data points, it is possible to estimate the window width and fixed window step accordingly. The default choice of $k_l = 4$ (tetramer frequencies) usually is robust [19].

II. Dimensionality Reduction

The analysis of high-dimensional data is often problematic due to the curse of dimensionality [12]. Hence, it is crucial to reduce the dimension while keeping desired properties such as cluster structure.

We employ Barnes-Hut SNE (BH-SNE) [26] as a central method in order to reduce dimensionality in a nonlinear way. It is based on t-distributed stochastic neighborhood embedding [27] which aims to minimize the difference between two distributions of pairwise probabilities in the high and lower dimensional space. Considering N data points, high dimensional probabilities are defined as $p_{ij} = (p_{i|j} + p_{j|i}) \ / \ (2N)$ where

$$p_{j|i} = \frac{\exp\left(-||\vec{x}_i - \vec{x}_j||^2 / 2\sigma_i^2\right)}{\sum_{l \neq i} \exp\left(-||\vec{x}_i - \vec{x}_l||^2 / 2\sigma_i^2\right)}$$

can be interpreted as the probability that $\vec{x}_i$ would pick $\vec{x}_j$ as its neighbor under the assumption that it was picked from a Gaussian distribution centered at $\vec{x}_i$. The parameter σ_i for each data point is automatically determined using a hyper-parameter called *perplexity* that is usually insensitive. Probabilities in $\mathbb{R}^d$ are modeled by

$$q_{ij} = \frac{\left(1 + ||\vec{y}_i - \vec{y}_j||^2\right)^{-1}}{\sum_{m \neq l} \left(1 + ||\vec{y}_m - \vec{y}_l||^2\right)^{-1}}$$

Using the long tailed student-t distribution instead of the Gaussian has the advantage that it allows to avoid the *crowding problem* in low dimensional spaces, leaving more space for distant pairs of points. The Kullback-Leibler divergence between p_{ij} and q_{ij} is used to minimize the difference between both probability distributions by numerical optimization.

The original t-SNE method has the major drawback of a quadratic computational runtime and memory complexity, making it unsuitable for larger data sets such as genomes.

bioRxiv preprint first posted online Jun. 15, 2015; doi: http://dx.doi.org/10.1101/020859. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC-ND 4.0 International license.

Barnes-Hut SNE overcomes this deficiency by approximating the similarities between input points, effectively reducing its runtime complexity to linearithmic time while using only linear memory.

BH-SNE has shown to work well for various kinds of data, including genomes [17, 10]. As it puts a particular focus on clusters, in this respect, it is superior to other, possibly linear methods such as PCA. As qualitatively shown in Figure 2, BH-SNE generates clusters which are more compact and separated. A quantitative analysis of the suitability of BH-SNE for clustering DNA sequences can be found in [19]. Clustering algorithms can certainly take advantage of this property.

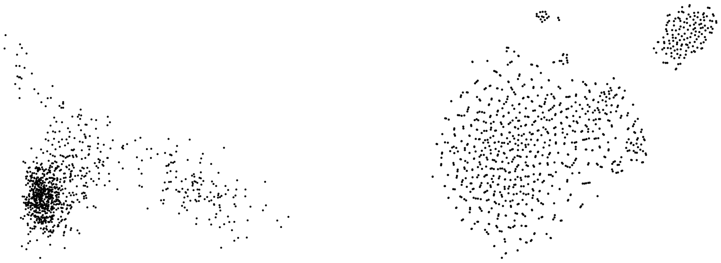

Figure 2: *Comparison of dimension reduction using PCA (left) and BH-SNE (right) of a contaminated single-cell sample.*

III. Clustering

The goal of clustering is to find a grouping of a given set of data points that is optimal with respect to different objectives. As the notion of a cluster is ill-posed, many different clustering algorithms aim for different objectives [9]. Most techniques depend on a parameter, which is the number of clusters k. In some fields, including genome clustering, k is unknown and has to be estimated. A special case is given by SCS contamination detection where the determination of the actual number of clusters is secondary. Here, one is not necessarily interested in a specific grouping, rather in the distinction between $k = 1$ (no structure, clean sample) and $k > 1$ (clusters, contaminated sample). For this task, a large subset of k-estimation procedures falls out of focus since they operate on cluster-specific characteristics, only defined for $k > 1$. The case $k = 1$ requires an appropriate null model to which the data is compared to in order to be able to detect no structure [25]. In the following, we will review techniques suitable for this task.

We propose a set of differentation criteria for clustering algorithms with respect to the suitability in a contamination detection tool:

- Parameter complexity: As it is difficult to optimize parameters of a given method, even more for researchers from external domains, i.e. biology, the number of parameters should be low. All parameters should be either robust to changes, or easily controllable, possibly through custom hyper-parameters.
- Interpretability of results: Having no absolute truth in clustering, it is desirable for a method to deliver results which are interpretable, possibly including measures of confidence, i.e. p-values.
- Existence of an optimal labeling: The method should be able to provide a grouping for the optimal number of clusters, making it possible to tag all contaminant parts in a given sample.
- Computational complexity: Even single-cell genomes can be very large and samples plenty. For interactive data investigation, methods with long runtime are not desirable.

Given these criteria, we will give an overview of relevant methods and discuss pros and contras. In the following, method parameters are given as P_x where the subscript x denotes the actual parameter.

III.1 Gap statistic

A very frequently applied method for estimating the number of clusters is the *Gap Statistic* [25]. It considers the difference of the within-cluster dispersion compared to its expected value under a given reference null distribution which is taken to be uniform, aligned at the principal components of the data. k is taken according to the largest gap between those measures which are estimated using P_B repetitions. It is found by locating either an elbow or the maximum in the gap curve for a given range $k \in \{1, \dots, P_{P_{k_{max}}}\}$. However, the elbow might not be pronounced enough or the maximum is surrounded by a noisy plateau, sometimes resulting in wrong estimations. Even though, the method is statistically well founded, it does not provide any interpretable significance of the result.

III.2 Sub-sampling stability

The *Model Explorer* algorithm (ME) and related methods [4, 30] determine k by looking at the stability of clusterings for $k \in \{2, \dots, P_{k_{max}}\}$ with respect to random sub-sampling of the data. Here, a random subset with fixed ratio P_r is drawn from the data. The number of clusters is chosen as the largest k for which a clustering is stable. Here, *stable* is defined as the average similarity (over P_B repetitions) between sub-samples being above a fixed threshold P_{t_0}. If no given clustering is stable, $k = 1$ is assumed. However, this threshold is arbitrary and often depends on the data. Additionally, such methods tend to find stable solutions even on random data [32].

III.3 Model Order Selection by Randomized Maps

Model Order Selection by Randomized Maps (MOSRAM) [5] can be seen as a variation of the previous Model Explorer algorithm. Instead of sub-sampling, it uses P_B random projections of the data and introduces an additional statistical test for

bioRxiv preprint first posted online Jun. 15, 2015; doi: http://dx.doi.org/10.1101/020859. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC-ND 4.0 International license.

the difference of k-clusterings. A set of significant cluster numbers is selected according to p-values of the test using a significance level P_α. The method still includes the same, fixed similarity threshold P_{t_0} as the basis of the test statistic. The random projections take an additional parameter P_ϵ that determines the target dimension. However, in this context, it does not make sense to apply random projections to data which already has been reduced in dimension, however it is worth to note, that the statistical test can also be applied to the Model Explorer algorithm.

III.4 Ensemble k-means clustering

The *Multi-K* algorithm [16] randomly samples k_i from a given distribution such as uniformly from $k_i \in P_K = \{1, \dots, P_{k_{max}}\}$. It then applies the k-means algorithm P_B times using different k_i in order to build a graph G in which edge weights between points of the same cluster are increased in every iteration. In the following, all edge weights are decreased P_B times, in each iteration counting the number of connected components of G. The optimal number of clusters is taken as the k occurring most often throughout all iterations.

III.5 Prediction-based resampling

In *Clest* [7], using a fixed ratio P_r, the data is split into training set L_b and test set T_b multiple times $b \in \{1, \dots, B\}$. In each iteration, a linear classifier P_C is trained using L_b and tested on T_b. At the same time, T_b is also clustered and both the classification and clustering results are compared using an internal cluster validity index. The same procedure is done for a number B_0 of simulated reference null data sets. For each $k \in \{2, \dots, P_{k_{max}}\}$, original and reference performances are compared, resulting in a set of p-values p_k. The number of clusters is estimated as k for which p_k is significant (using a level P_α) and has the largest statistical power.

III.6 Dip-means

Using the *Dip Statistic* [11], *Dip-means* [15] is based on a significance test for multimodality. It starts by considering the whole data set as one cluster and tests against the null hypothesis that the cluster is unimodal. The test is applied to each data points distance distribution w.r.t. to all other points within the cluster. If a certain percentage P_v of points has significant evidence against unimodality (using a significance level P_α), the cluster is split into two distinct clusters using a clustering algorithm such as k-means. On each resulting cluster, the procedure is performed recursively until no cluster is multimodal anymore. The parameter P_α in combination with P_v can be used to control the number of false positives, detected by the method.

III.7 Number of connected components

Spectral clustering can be used to estimate the number of clusters using the eigengap heuristic [29]. However, this heuristic relies on a pronounced elbow in the distribution of eigenvalues that is not distinct enough in most real world data sets, making its identification difficult. Additionally, the construction of the underlying graph heavily influences the result. As BH-SNE (subsection II) often produces very pronounced clusters, we found that counting the number of connected components (CC) of a P_{k_n}-mutual-nearest-neighbor graph is often sufficient. This way, counting the number of eigenvalues $\lambda = 1$ of the normalized graph Laplacian results in the number of clusters as long as they are compact and separated. Tarjans algorithm can be used to estimate this number much faster than using eigendecomposition.

Most of the presented methods deliver an optimal labeling that corresponds to the optimal number of clusters. Only ME, MOSRAM and Clest do not provide such. It is possible to do a posterior labeling. However, it is in no connection to the estimation method, possibly delivering confusing results.

The runtime complexity of all methods is either quadratic or better. For some, it depends most on the number of reference data sets P_B and the underlying clustering algorithm. It is worth to note, that CC can determine k in linear time and using Dip-means, it is possible to detect contamination early in the algorithm by only testing for multimodality on the one cluster containing all data.

III. Preliminary Results

I. Methods

Early tests on real single-cell data indicate that most methods work reasonably well. However, we found that stability based methods tend to fail to recognize no structure in the data, i.e. non-contaminated samples. This is due to the fact that even in such data, clusterings can appear stable [32], hence we discard these algorithms (Model Explorer, MOSRAM) as suitable candidates. The Gap Statistic also often fails to discover non-structured data. Here, the gap curve shows spurious elbows, even when there is only one cluster in the data. Also, in Clest, the number of parameters seems to lead to unstable results. Although it does work in the majority of cases, its p-values on which basis the decision for k is made, are often very near to the significance level, making it very sensitive. Additionally, included parameters are difficult to tune for end users. In Multi-K, even in the presence of more clusters, $k = 1$ cluster is detected too often, resulting in a number of false negatives. This is due to the fact that, in the underlying graph G, positive edge weights between members of different clusters might persist for a long time, favoring a single connected component. Estimating the number of connected components

bioRxiv preprint first posted online Jun. 15, 2015; doi: http://dx.doi.org/10.1101/020859. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC-ND 4.0 International license.

of a P_{k_n}-mutual-nearest-neighbor graph gives correct results in the majority of cases. Here, P_{k_n} might be interpreted as the minimum number of data points in a cluster, thus is easily interpretable. However, this does hold only for well separated and compact clusters where the largest distance to a nearest neighbor of the same cluster is smaller than the distance to the nearest point in the most nearby cluster. Overlapping or nearby clusters pose a problem for all of the presented methods and are difficult to distinguish from being one. Here, dip-means is standing out as it is also able to discover also overlapping clusters. As long as the structure is significantly multimodal, it is able to detect such.

All of the presented methods work fairly well for estimating the number of clusters for $k > 1$. However, only two methods, counting the number of connected components and dip-means, stand out in their ability to properly differentiate between $k = 1$ and $k > 1$ without too many false detections. Their parameters are few and easily interpretable, making them good candidates for being applied in single-cell contamination detection pipeline. Still, a thorough evaluation of its behavior for this task is required and methods might be modified and combined with other, possibly supervised methods.

II. Results on annotated data

In order to evaluate the performance of our pipeline, we use simulated single-cell data. The main advantages over using real data from the laboratory is given by a fully correct ground truth and ease of controlling the level of phylogenetic relatedness of included genomes. Here, we expect that quantifying contamination in samples containing remotely related genomes results in a higher detection rate than in samples with very closely related genomes, i.e. from the same species. We employ mdasim [23] to simulate multiple displacement amplification from a given reference sample that is either clean or contaminated. To simulate the subsequent sequencing process, we use ART [13] to generate reads. Finally, contigs are assembled by SPAdes [3]. In our observations we found no difference between using the simulated data and real-world samples, both in data quality and detection rates.

Our pipeline reliably detects most contaminated samples with high confidence. Here, easily traceable contamination (i.e. the contaminant is only remotely related) can be detected by counting the number of connected components of a nearest-neighbor graph, as usually, in such cases clusters are very well pronounced. This step can detect most contamination very fast (linear in the number of graph nodes and edges) and further analysis may be skipped.

In uncertain cases (i.e. the contaminant is very closely related), the neighborhood graph might not contain separately connected components anymore. Here, dip-means is employed to test for multimodality of the data. Again, even closely related species can be separated with this approach. Two example are given by Figure 4 and Figure 5. The samples contain two species from the same family and genus respectively. Even though the two graph components are connected by a small bridge, contamination can still be detected by finding a significant multimodality of pairwise distances ($p = 0$). In contrast, Figure 3 depicts a clean sample. Ideally, the neighborhood graph contains only one connected component and the distribution of pairwise distances is unimodal ($p = 0.12$), indicating no contamination.

III. Summary & ongoing work

Preliminary results are very promising and we plan a more thorough evaluation including many samples with different phylogenetic relatedness. Early observations show that our pipeline can discriminate genomes from the same family and even genus. Because all involved method parameters are determined by the system, our pipeline allows for fully automated batch processing. Still, it allows for interactive inspection by experts and provides p-values as confidence. Furthermore, we plan to include various meta data sources such as from the sequencing or assembly process to improve detection accuracy even more.

References

[1] Method of the year 2013. *Nature Methods*, 11(1):1–1, Jan 2014. Editorial.

[2] Christina Ander, Ole B Schulz-Trieglaff, Jens Stoye, and Anthony J Cox. metabeetl: high-throughput analysis of heterogeneous microbial populations from shotgun dna sequences. *BMC bioinformatics*, 14(Suppl 5):S2, 2013.

[3] Anton Bankevich, Sergey Nurk, Dmitry Antipov, Alexey A Gurevich, Mikhail Dvorkin, Alexander S Kulikov, Valery M Lesin, Sergey I Nikolenko, Son Pham, Andrey D Prjibelski, et al. Spades: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology*, 19(5):455–477, 2012.

[4] Asa Ben-Hur, Andre Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. In *Pacific symposium on biocomputing*, volume 7, pages 6–17, 2001.

[5] Alberto Bertoni and Giorgio Valentini. Model order selection for clustered bio-molecular data. 2006.

[6] Paul C Blainey. The future is now: single-cell genomics of bacteria and archaea. *FEMS microbiology reviews*, 37(3):407–427, 2013.

[7] Sandrine Dudoit and Jane Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome biology*, 3(7):research0036, 2002.

[8] James Eberwine, Jai-Yoon Sul, Tamas Bartfai, and Junhyong Kim. The promise of single-cell sequencing. *Nature methods*, 11(1):25–27, 2014.

[9] Vladimir Estivill-Castro. Why so many clustering algorithms: a position paper. *ACM SIGKDD Explorations Newsletter*, 4(1):65–75, 2002.

bioRxiv preprint first posted online Jun. 15, 2015; doi: http://dx.doi.org/10.1101/020859. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC-ND 4.0 International license.

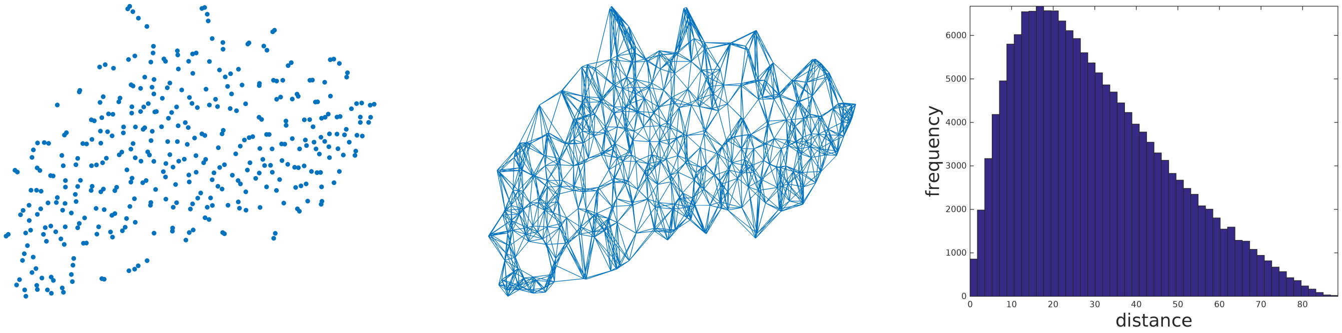

Figure 3: *Cluster analysis of a clean sample. Left: t-SNE representation. Center: 9-nearest-neighbor graph. Right: distribution of pairwise distances.*

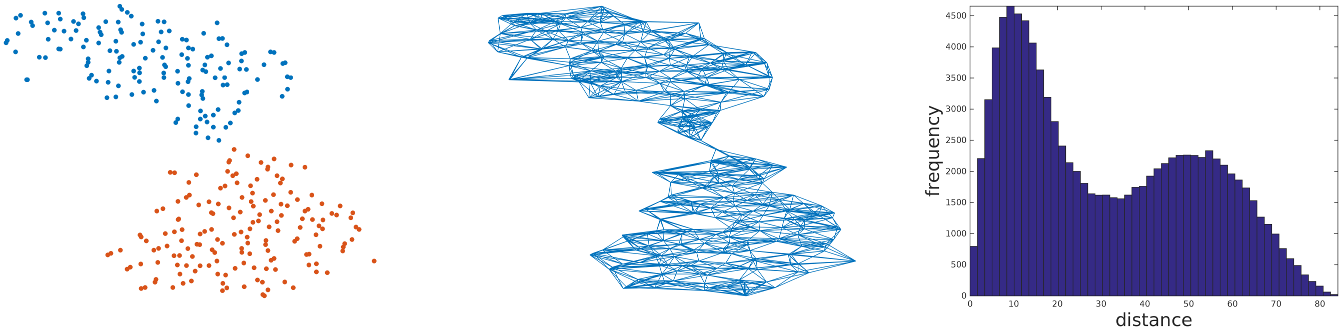

Figure 4: *Cluster analysis of a contaminated sample containing two genomes from the* ***same family*** *(Streptococcaceae). Left: t-SNE representation. Center: 9-nearest-neighbor graph. Right: distribution of pairwise distances.*

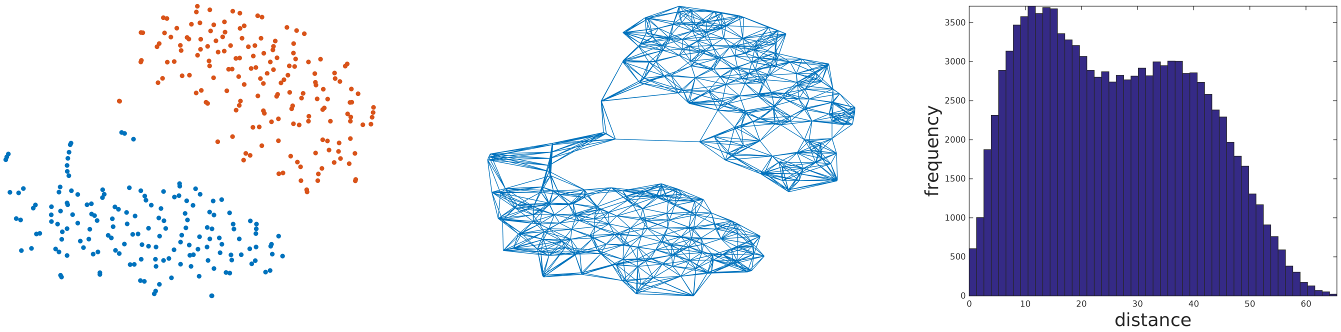

Figure 5: *Cluster analysis of a contaminated sample containing two genomes from the* ***same genus*** *(Streptococcus). Left: t-SNE representation. Center: 9-nearest-neighbor graph. Right: distribution of pairwise distances.*

bioRxiv preprint first posted online Jun. 15, 2015; doi: http://dx.doi.org/10.1101/020859. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC-ND 4.0 International license.

[10] Andrej Gisbrecht, Barbara Hammer, Bassam Mokbel, and Alexander Sczyrba. Nonlinear dimensionality reduction for cluster identification in metagenomic samples. In *Information Visualisation (IV), 2013 17th International Conference*, pages 174–179. IEEE, 2013.

[11] John A Hartigan and PM Hartigan. The dip test of unimodality. *The Annals of Statistics*, pages 70–84, 1985.

[12] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[13] Weichun Huang, Leping Li, Jason R Myers, and Gabor T Marth. Art: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012.

[14] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[15] Argyris Kalogeratos and Aristidis Likas. Dip-means: an incremental clustering method for estimating the number of clusters. In *Advances in neural information processing systems*, pages 2393–2401, 2012.

[16] Eun-Youn Kim, Seon-Young Kim, Daniel Ashlock, and Dougu Nam. Multi-k: accurate classification of microarray subtypes using ensemble k-means clustering. *BMC bioinformatics*, 10(1):260, 2009.

[17] Cedric C Laczny, Nicolás Pinel, Nikos Vlassis, and Paul Wilmes. Alignment-free visualization of metagenomic data by nonlinear dimension reduction. *Scientific reports*, 4, 2014.

[18] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, and Junjie Wu. Understanding of internal clustering validation measures. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 911–916. IEEE, 2010.

[19] Markus Lux, Alexander Sczyrba, and Barbara Hammer. Automatic discovery of metagenomic structure. In *Accepted to the 2015 IEEE International Joint Conferences on Neural Networks*.

[20] Sharmila S. Mande, Monzoorul Haque Mohammed, and Tarini Shankar Ghosh. Classification of metagenomic sequences: methods and challenges. *Briefings in Bioinformatics*, 13(6):669–681, 2012.

[21] Raeece Naeem, Mamoon Rashid, and Arnab Pain. Readscan: a fast and scalable pathogen discovery program with accurate genome relative abundance estimation. *Bioinformatics*, 29(3):391–392, 2013.

[22] Christian Rinke, Patrick Schwientek, Alexander Sczyrba, Natalia N Ivanova, Iain J Anderson, Jan-Fang Cheng, Aaron Darling, Stephanie Malfatti, Brandon K Swan, Esther A Gies, et al. Insights into the phylogeny and coding potential of microbial dark matter. *Nature*, 2013.

[23] Zeinab Tagliavi and Sorin Draghici. Mdasim: A multiple displacement amplification simulator. In *Bioinformatics and Biomedicine (BIBM), 2012 IEEE International Conference on*, pages 1–4. IEEE, 2012.

[24] Hanno Teeling, Anke Meyerdierks, Margarete Bauer, Rudolf Amann, and Frank Oliver Glöckner. Application of tetranucleotide frequencies for the assignment of genomic fragments. *Environmental microbiology*, 6(9):938–947, 2004.

[25] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[26] Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1):3221–3245, 2014.

[27] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[28] Lucas Vendramin, Ricardo J. G. B. Campello, and Eduardo R. Hruschka. Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, 3(4):209–235, 2010.

[29] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[30] Ulrike Von Luxburg. *Clustering Stability*, volume 6. Now Publishers Inc, 2010.

[31] Derrick E Wood and Steven L Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol*, 15(3):R46, 2014.

[32] Yasin ÈŸenbabaoğlu, George Michailidis, and Jun Z Li. Critical limitations of consensus clustering in class discovery. *Scientific reports*, 4, 2014.