Langenkämper *et al. BMC Bioinformatics* (2014) 15:384
DOI 10.1186/s12859-014-0384-0

BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

AKE - the Accelerated *k*-mer Exploration web-tool for rapid taxonomic classification and visualization

Daniel Langenkämper[1*], Alexander Goesmann[2] and Tim Wilhelm Nattkemper[1]

Abstract

Background: With the advent of low cost, fast sequencing technologies metagenomic analyses are made possible. The large data volumes gathered by these techniques and the unpredictable diversity captured in them are still, however, a challenge for computational biology.

Results: In this paper we address the problem of rapid taxonomic assignment with small and adaptive data models (< 5 MB) and present the accelerated *k*-mer explorer (AKE). Acceleration in AKE's taxonomic assignments is achieved by a special machine learning architecture, which is well suited to model data collections that are intrinsically hierarchical. We report classification accuracy reasonably well for ranks down to order, observed on a study on real world data (Acid Mine Drainage, Cow Rumen).

Conclusion: We show that the execution time of this approach is orders of magnitude shorter than competitive approaches and that accuracy is comparable. The tool is presented to the public as a web application (url: https://ani.cebitec.uni-bielefeld.de/ake/, username: bmc, password: bmcbioinfo).

Keywords: Metagenomics, Classification, Acceleration, Web-based, H2SOM, k-mer

Background

Metagenomics is the direct sequencing and analysis of environmental samples. Metagenomic studies are used in a variety of fields including, e.g. bio-medical studies [1] and ecological diversity studies [2]. As a first step after sequencing taxonomic composition is estimated and taxonomic categories are assigned to the data. This is a challenging problem due to sequence length and complexity of the data captured [2]. For analysis of the taxonomic composition the analysis of 16S rRNA sequences is a prominent step, see [3,4]. This imposed some limitations, e.g. the copy number can vary by an order of magnitude [5] and therefore we will focus on whole metagenome analysis. Multiple tools exist that are able to predict the class of a genomic sample sequence, most of them using alignments (e.g. Megan4 [6], (Web)Carma3 [7,8], MG-Rast [9]). As these can be very time consuming, alternative approaches based on profile features have been proposed (Phylopythia(S) [10,11], NBC [12], TACELM [13], TAXSOM [14], PhymmBL [15], Kraken [16], taxy [17]). Sequence data are transformed to profile features, i.e. feature vectors that consist of various measurements describing the nucleic composition of the sequence. Frequently employed characteristics are G/C content [18] and *k*-mer occurrence [19,20]. The speedup of these techniques is traded in for a loss of accuracy, compared to the alignment-based methods. Nevertheless, it has been shown that *k*-mer profiles are distinctive enough for binning in metagenomic studies and for classification up to certain levels in the tree of life [21]. For benchmarking we compared AKE with Phylopythia(S) and NBC. Phylopythia classifies profile features with a SVM-based classifier architecture. The web-based version is called PhylopythiaS. It uses two different models, either a generic model for classification or a sample specific one, which can be generated by the user prior to classification. We benchmarked against the parameterless generic model. NBC implements the naive Bayes classifier for taxonomic assignment as a web application. The *k*-mer length as

*Correspondence: dlangenk@cebitec.uni-bielefeld.de
[1]Biodata Mining, Bielefeld University, Universitätsstraße 15, Bielefeld, Germany
Full list of author information is available at the end of the article

well as the genomes to match against can be chosen. We chose the Bacteria/Archea genomes to match against and a *k*-mer length of 6 for benchmarking.

This paper presents AKE (Accelerated *k*-mer Exploration web-tool) a computational approach to rapid taxonomic assignment for an immediate response to new data. A rapid taxonomic assignment can be of interest, when data sets from lots of samples are to be analyzed immediately or new data sets are generated rapidly by filtering and fusion. A result of AKE is a rapid taxonomic assignment presented as a web-based, interactive and dynamic visualization. AKEs computational speed is achieved by (1) using refined *k*-mer profile features [21], (2) a data-driven, i.e. learned, hierarchical and descriptive model, which provides the basis for classification and visualization, and (3) parallel computing. This work is based on a previous paper by Martin et al. [21] sharing the features and binning method, namely the H^2SOM. However, the classifier architecture is different and Martin et al. do not provide a web interface for visualization of results. Furthermore, the execution speed is increased by using parallelization and a faster implementation. To boost classification accuracy a rejection class is introduced to the model containing non-specific profiles. This results in a web accessible system for low performance computers that features an immediate first visual inspection of new data, i.e. some data might be rejected if it is unspecific. The accuracy is comparable to similar approaches but with a faster execution time. The tool is publicly available as a web application (https://ani.cebitec.uni-bielefeld.de/ake/, username: bmc, pw: bmcbioinfo), which facilitates the ease of use. This releases the users of the burden of resourceful operations on their own systems, e.g. analyses on small-scale computers in laboratories are made possible. Furthermore, no software packages have to be downloaded and installed. The only requirement is an up-to-date web-browser ($\approx$ not older than 2 years).

Recent reports of IMG4 [[22], progress report (http://img.jgi.doe.gov/w/doc/releaseNotes.pdf)] show a rapidly growing amount of available metagenomes. Likewise, the PubMed hits for the term "metagenomics" grew massively in the latter years, showing the importance of the field.

The following Methods section describes the features, methods and data used in this study and how these are used to build a classification system for metagenomic data. In the Results section we present the performance on two real world data sets, compared to similar approaches. Furthermore, the differences in runtime are reported. The Conclusion sums up the results of this study.

Methods

As can be seen in Figure 1 AKE consists of two modules: taxonomic assignment (TA) and modeling (M). In the M-module, a reference set of genome sequences $\Gamma_{\text{ref}} = \{S^{(\zeta)}\}$ is used to learn a model that describes the function for assignment of taxonomic classes to sequence reads $S^{(\zeta)}$ based on a read's profile feature $\mathbf{x}^{(\zeta)}$. For assigning new sequence data $\Gamma_{\text{new}}\{S^{(\xi)}\}$ with the TA-module, these reads are also represented by profile features $\mathbf{x}^{(\xi)}$ and those are assigned to taxonomic classes. The composition of all assignments of $\Gamma_{\text{new}}\{S^{(\xi)}\}$ are visualized in a dynamic and interactive web-tool.

k-mer features

For using the sequences $S^{(\zeta)}\zeta = 0, \ldots, n$ with a mathematical model like the H^2SOM, features $\mathbf{x}^{(\zeta)}\zeta = 0, \ldots, n$ have to be computed for the sequence reads. For this purpose *k*-mer profiles with three different normalizations are used and referred to as $\left[\mathbf{x}^{(\zeta)}_{\text{tf}}, \mathbf{x}^{(\zeta)}_{\text{tfti}}, \mathbf{x}^{(\zeta)}_{\text{oligo}}\right]$. They are listed here with basic explanations, further information can be found in [21].

A *k*-mer $\kappa_j(k, \Sigma)$ is a word of length k on an alphabet Σ. In this case $\Sigma = \{a, c, g, t\}$ is the DNA alphabet and therefore 4^k *k*-mers $\kappa_{j(j=0,\ldots,4^k-1)}(k, \Sigma)$ exist. Let $t_j^{(\zeta)}$ be the number of occurrences of the *k*-mer $\kappa_j(k, \Sigma)$ in sequence $S^{(\zeta)}$, $\mathcal{C}_\kappa(\kappa_j(k, \Sigma)$ a function counting these occurrences and S' a substring of S matching the specified *k*-mer.

$$\begin{aligned} t_j^{(\zeta)} &= \mathcal{C}_\kappa\left(\kappa_j(k, \Sigma), S^{(\zeta)}\right) \text{ with} \\ \mathcal{C}_\kappa\left(\kappa_j(k, \Sigma), S^{(\zeta)}\right) &= \left|\left\{S' \in S^{(\zeta)} | S' = \kappa_j(k, \Sigma)\right\}\right| \end{aligned} \tag{1}$$

A *k*-mer-profile $K^{(\zeta)}(k, \Sigma) \in \mathbb{N}^{4^k}$ is defined as

$$K^{(\zeta)}(k, \Sigma) = \left(t_0^{(\zeta)}, t_1^{(\zeta)}, \ldots, t_{4^k-1}^{(\zeta)}\right) \tag{2}$$

For the sake of compactness we omit the term (k, Σ) for $K^{(\zeta)}(k, \Sigma)$ and $\kappa_j(k, \Sigma)$ in the following text. The term frequency features (tf) are gained by normalizing every *k*-mer profile to unit length.

$$\mathbf{x}^{(\zeta)}_{\text{tf}} = \frac{K^{(\zeta)}}{\left\|K^{(\zeta)}\right\|} \tag{3}$$

By taking into account the abundance of a certain *k*-mer in all *k*-mer-profiles we gain the term frequency term importance (tfti) weighted features. Let $t_j = \sum_\zeta t_j^{(\zeta)}$ denote the sum of frequencies of *k*-mer κ_j in all *k*-mer-profiles in Γ_{ref}. Let $t^{(\zeta)} = \sum_0^{4^k-1} t_j^{(\zeta)}$ be the sum of all frequencies for a sequence $S^{(\zeta)}$. Therefore, we compute the tfti-weighted features for every *k*-mer profile as:

$$\mathbf{x}^{(\zeta)}_{\text{tfti}} = \left(\frac{t_0^{(\zeta)}}{t_0 t^{(\zeta)}}, \frac{t_1^{(\zeta)}}{t_1 t^{(\zeta)}}, \ldots, \frac{t_{4^k-1}^{(\zeta)}}{t_{4^k-1} t^{(\zeta)}}\right) \tag{4}$$

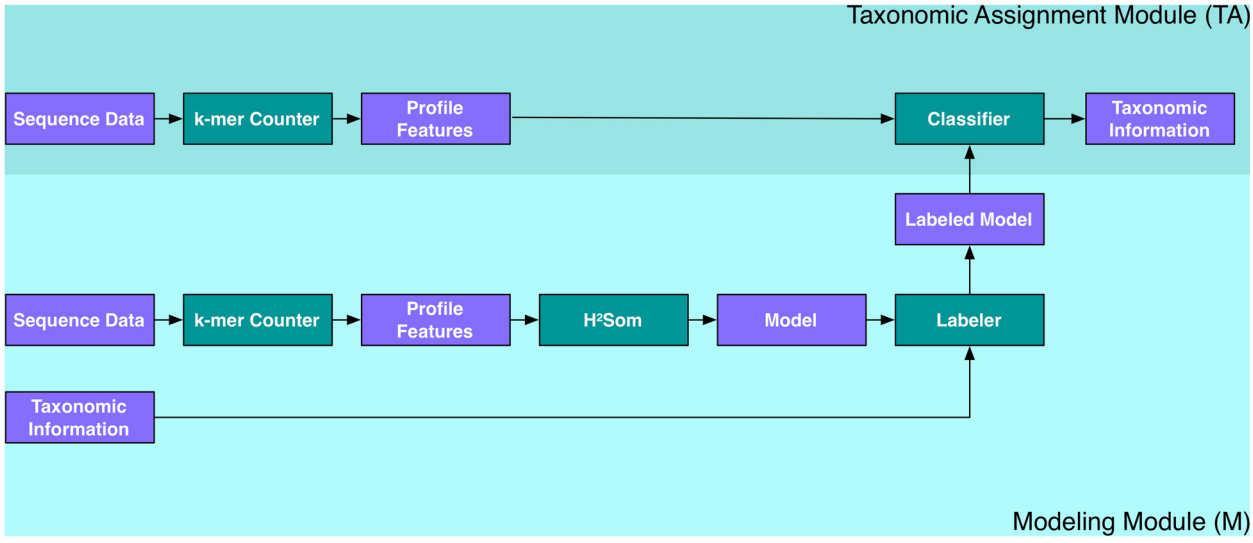

Figure 1 Workflow of AKE. Blue boxes are data, whereas green ones are applications.

To reduce a bias towards frequent k-mers the vectors are normalized to unit length.

Considering the over- and under-representation of k-mers in one sequence compared to the others we compute the oligo features (oligo). Therefore, the occurrence of each k-mer is computed and the expected occurrence of it is estimated. Let

$$p^{(\zeta)}(\eta) = \frac{1}{|S^{(\zeta)}|}\mathcal{C}_\Sigma(\eta) \text{ with } \eta \in \Sigma,$$
$$\mathcal{C}_\Sigma\left(\eta, S^{(\zeta)}\right) = \left|\left\{\eta' \in S^{(\zeta)} | \eta' = \eta\right\}\right|$$

be the probability to observe a certain nucleotide η in a sequence $S^{(\zeta)}$ with a sequence length $|S^{(\zeta)}|$ and let η' be a nucleotide in the sequence $S^{(\zeta)}$ matching a specified nucleotide. Let $E^{(\zeta)}(\kappa_j) \approx |S^{(\zeta)}| \prod_{l=0}^{k-1} p^{(\zeta)}(\kappa_{j,l})$ (with $\kappa_{j,l}$ referring to the l-th symbol in κ_j) be an estimate for the occurrence of a k-mer κ_j in a sequence $S^{(\zeta)}$. The contrast of expectation and observation is

$$g^{(\zeta)}(\kappa_j) = \begin{cases} 0, & \text{if } K_j^{(\zeta)} = 0 \\ \frac{K_j^{(\zeta)}}{E^{(\zeta)}(\kappa_j)}, & \text{if } K_j^{(\zeta)} > E^{(\zeta)}(\kappa_j) \\ -\frac{E^{(\zeta)}(\kappa_j)}{K_j^{(\zeta)}}, & \text{else} \end{cases}$$

The oligo features are computed for each k-mer as

$$\mathbf{x}_{\text{oligo}}^{(\zeta)} = \left(g^{(\zeta)}(\kappa_0), g^{(\zeta)}(\kappa_1), \ldots, g^{(\zeta)}\left(\kappa_{4^k-1}\right)\right) \quad (5)$$

The H2SOM classifier

For creating a descriptive model of the k-mers a Hyperbolic Self Organizing Map is used. The Self Organizing Map is a neural network proposed by Teuvo Kohonen [23]. Many variants have been proposed since, but all share the basic setup that consists of a set of neurons $(\mathbf{u}_i, z_i)_{i=1\ldots I}$ that are arranged in a grid with z_i being the grid coordinate and $\mathbf{u}_i$ being the attached neural unit also called the prototype. The architecture of the grid differs by the type applied.

In the Hyperbolic SOM (HSOM) [24] the algorithm is defined in non-euclidean space. The Hyperbolic Hierarchical SOM (H^2SOM) [25] as used in this paper introduces a hierarchical grid structure to the hyperbolic version.

In metagenomics, the H^2SOM has been applied already for visual exploration and binning [21]. In [26] it was shown, that clustering genome data with a HSOM correlates more to the tree-of-life structure than the standard SOM clustering.

The network is built by placing a central neuron and spawning its $s-1$ children around it using the Möbius transformation. This is done recursively for every neural unit until all have s neighbors and the maximum number of rings r is reached. Hereby $s-3$ neighbors are placed as children, 2 as siblings and 1 already exists as parent. For further information refer to [25] or see Additional file 1. An example of a H^2SOM grid with two rings and seven neighbors ($s = 7, r = 2$) is shown in Figure 2.

The learning of a non-euclidean SOM is done equivalently to an euclidean SOM using a reference set $\Gamma_{\text{ref}} = \{\mathbf{x}^{(\zeta)}\}$, but with a refined neighborhood function (Eq. 6) taking the change from euclidean to hyperbolic space into account.

$$h(i, i') = \exp\left(-\frac{\arctan\left(\left|\frac{z_i - z_{i'}}{1 - \bar{z}_i z_{i'}}\right|\right)}{\sigma^2(t)}\right). \quad (6)$$

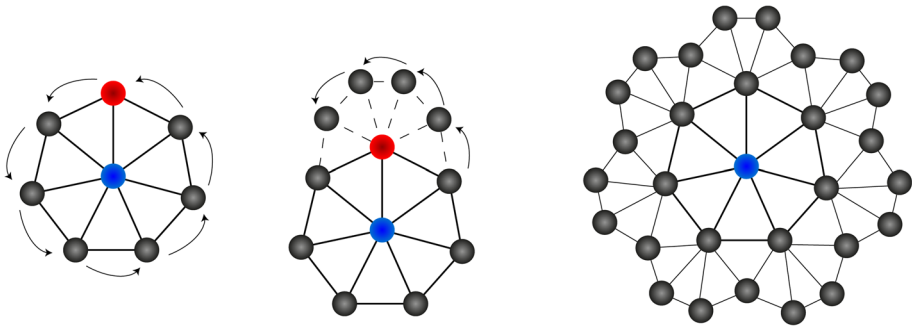

Figure 2 Architecture of H²SOM. The construction of a H²SOM grid with $s = 7$ and $r = 2$. A Möbius transform is used in a recursive way to create a regular shaped hierarchical grid in hyperbolic space.

The number of neural units in the grid of a H²SOM grows exponentially with the number of rings r. This leads to a more trustworthy mapping but dramatically increases the time required for the search for the best matching unit (BMU) during training. Leveraging the hierarchical structure of the grid a beam search is applied to approximate the global BMU in each training step. The search starts with the central neural unit as the initial BMU. For a beam width of $w = 1$ it continues by recursively choosing the BMU among the children of the last winning neural unit until it reaches the current periphery of the grid. The BMU determined for the last ring is an approximation of the global BMU. For values of $w > 1$ searching is done equivalently, but the children of w different neural units are searched for the BMU. It has been shown in [25] that this strategy accelerates the training significantly while staying close to or even surpassing the performance using global search.

The H²SOM depends on parameters that need to be optimized. These are the number of rings r, the spread factor s, the neighborhood adaption modifier n and the learning rate ϵ. The algorithm is very robust against changes in ϵ and n but the parameters determining size and architecture (r, s) are important. By employing cross validation the parameters $r = 5$ and $s = 8$ were determined to create a good descriptive model (see Additional file 2).

Taxonomic labeling of unsupervised neural networks

After training the H²SOM neural units are linked to semantics, i.e. taxonomic categories. To this end, the labeled training data $\Gamma_{\text{ref}}\left\{\left(\mathbf{x}^{(\zeta)}, L^{(\zeta)}\right)\right\}$, where Γ_{ref} is a set of features with their respective labels, are mapped to the H²SOM. This is done with a labeling function $\mathcal{L}(\mathbf{u}_i)$ that is defined on the Voronoi cell $V(\mathbf{u}_i)$ of the training data for each prototype

$$V(\mathbf{u}_i) : V(\mathbf{u}_i) = \left\{\mathbf{x}^{(\zeta)} \in \Gamma_{\text{ref}} | d\left(\mathbf{x}^{(\zeta)}, \mathbf{u}_i\right) < d\left(\mathbf{x}^{(\zeta)}, \mathbf{u}_j\right), \forall i \neq j\right\}$$

using a given metric d (in our case the euclidean metric). We propose two approaches: majority voting $\mathcal{L}^{\text{maj}}$ and purity voting $\mathcal{L}^{\text{pur}}$ defined as

$$\begin{aligned} \mathcal{L}^{\text{maj}}(\mathbf{u}_i) &= \arg\max_l(\Psi(V(\mathbf{u}_i), l)) \text{ with} \\ \Psi(V(\mathbf{u}_i), l) &= \left|\left\{\mathbf{x}^{(\zeta)} \in V(\mathbf{u}_i) | L^{(\zeta)} = l\right\}\right| \end{aligned} \tag{7}$$

and

$$\mathcal{L}^{\text{pur}}(\mathbf{u}_i) = \begin{cases} \mathcal{L}^{\text{maj}}(\mathbf{u}_i), & \text{if } \Psi(V(\mathbf{u}_i), \mathcal{L}^{\text{maj}}(\mathbf{u}_i)) > \alpha \\ \mathcal{R}, & \text{else} \end{cases}, \tag{8}$$

with $\mathcal{R}$ being a special label namely the rejection class and α a threshold value.

Classification rules

A H²SOM labeled in one of the above ways can be used for classification. To assign a sample ξ (a sequence), the profile feature vector $\mathbf{x}^{(\xi)}$ is computed, employing the same k-mer normalization strategy as used for labeling the model. For assignment a particular function $\mathcal{C}\left(\mathbf{x}^{(\xi)}\right)$ is chosen from $\left[\mathcal{C}^{\text{nn}}\left(\mathbf{x}^{(\xi)}\right), \mathcal{C}^{\text{thresh}}\left(\mathbf{x}^{(\xi)}\right), \mathcal{C}^{\text{nbrs}}\left(\mathbf{x}^{(\xi)}\right)\right]$, defined in the following.

The most straightforward function is to assign $\mathbf{x}^{(\xi)}$ to the label $\mathcal{L}(\mathbf{u}_j)$, which is assigned to the nearest neighbor $\mathbf{u}_j$ in the model.

$$\mathcal{C}^{\text{nn}}\left(\mathbf{x}^{\xi}\right) = \mathcal{L}(\mathbf{u}_j) \text{ with } j = \left(\arg\min_i d\left(\mathbf{x}^{(\xi)}, \mathbf{u}_i\right)\right) \tag{9}$$

Furthermore, the distance function $d\left(\mathbf{x}^{(\xi)}, \mathbf{u}_j\right)$ can be seen as a certainty measure that the BMU $\mathbf{u}_j$ is the correct association of $\mathbf{x}^{(\xi)}$. Therefore, we define an

arbitrary threshold β beyond which the association is assumed to be uncertain. The value of β is empirically determined.

$$\mathcal{C}^{\text{thresh}}\left(\mathbf{x}^{(\xi)}\right) = \begin{cases} \mathcal{C}^{\text{nn}}\left(\mathbf{x}^{(\xi)}\right), & \text{if } d\left(\mathbf{x}^{(\xi)}, \mathbf{u}_j\right) < \beta \text{ with } j = \arg\min_i d\left(\mathbf{x}^{(\xi)}, \mathbf{u}_i\right) \\ \mathcal{R}, & \text{else} \end{cases} \tag{10}$$

The previous strategies determine the label in a Winner-Takes-All (WTA) manner. But the H^{2}SOM has the property that neighboring neural units, i.e. grid neighbors, share common properties, usually referred to as "neighborhood preservation". The third version uses this feature to reduce the number of false positive classifications. To this end, the neighborhood of a BMU is evaluated to smooth out unlikely assignments with a large BMU distance and "taxonomic disagreement" to the neighborhood.

$$\mathcal{C}^{\text{nbrs}}\left(\mathbf{x}^{\xi}\right) = \begin{cases} \mathcal{L}(\mathbf{u}_{j+1}), & \text{if } d\left(\mathbf{x}^{(\xi)}, \mathbf{u}_{j+1}\right) + d\left(\mathbf{x}^{(\xi)}, \mathbf{u}_{j-1}\right) < 3 * d\left(\mathbf{x}^{(\xi)}, \mathbf{u}_j\right) \wedge \\ & \mathcal{L}(\mathbf{u}_{j-1}) = \mathcal{L}(\mathbf{u}_{j+1}) \text{ with } j = \arg\min_i d\left(\mathbf{x}^{(\xi)}, \mathbf{u}_i\right) \\ \mathcal{C}^{\text{nn}}\left(\mathbf{x}^{(\xi)}\right), & \text{else} \end{cases} \tag{11}$$

For training sequences exceeding 4 kb from the NCBI full genome database (bacteria/virus, 2014/04/13) were used. A list of GI numbers (http://www.ncbi.nlm.nih.gov/Class/FieldGuide/glossary.html#GI) is provided (see Additional file 3). Out of these sequence data four different data sets were generated for model building. Therefore, the sequences $S^{(\zeta)}$ were cut at different length $\left(15\text{ kb}, 4\text{ kb}, \frac{|S^{(\zeta)}|}{2}, \frac{|S^{(\zeta)}|}{4}\right)$.

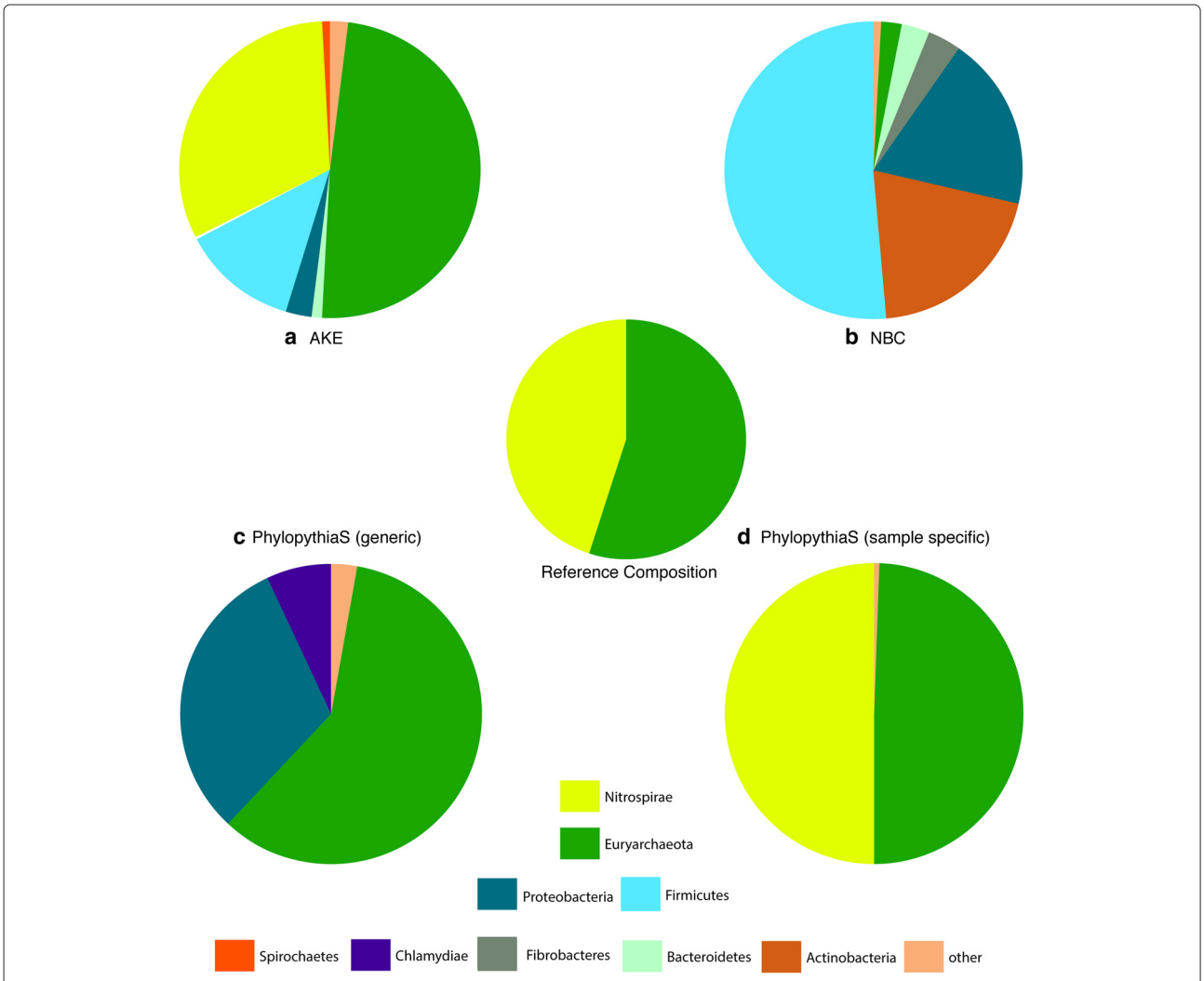

Figure 3 Results of the AMD study. Comparison of AKE/NBC/PhylopythiaS with a generic model/PhylopythiaS with a sample specific model. Data for NBC/PhylopythiaS derived from [11].

Results and discussion

For parameter optimization and model evaluation a cross validation study (see Additional file 2) was done. The most promising k-mer length was determined to be $k = 4$. For larger values of k the classification accuracy increased partly, however a decrease in speed can be observed. The two labeling strategies (Eqs. 7, 8), for building the taxonomic model, combined with the three different classification algorithms (Eqs. 9, 10, 11) were applied. A trade-off between correctness of assignment and number of rejections was observed for all six variants. A good balance between assignment correctness, number of rejections and execution speed was determined using purity voting (Eq. 8) for model construction and nearest neighbor selection (Eq. 9) for taxonomic assignment.

Thus, for the following real world data set examples, purity voting with a threshold of $\alpha = 0.8$ for labeling (Eq. 8) and the nearest neighbor strategy (Eq. 9) for assignment were the most promising settings compared to the other variants. For the H^2SOM algorithm an architecture with $r = 5$ rings and $s = 8$ neighbors was chosen.

Acid mine drainage

The Acid Mine Drainage data set [27] was taken at Iron Mountain in California. The community is comprised of five high abundant species namely Ferroplasma Types I and II, a Thermoplasmatales species, all of phylum Euryarchaeota, and Leptospirillum sp. Group I and II of phylum Nitrospirae. The data has been received from DOE Joint Genome Institute (http://img.jgi.doe.gov (taxon 2001200000)) along with its taxonomic affiliation and is build of 1183 scaffolds of approximately 10 Mb of sequence information.

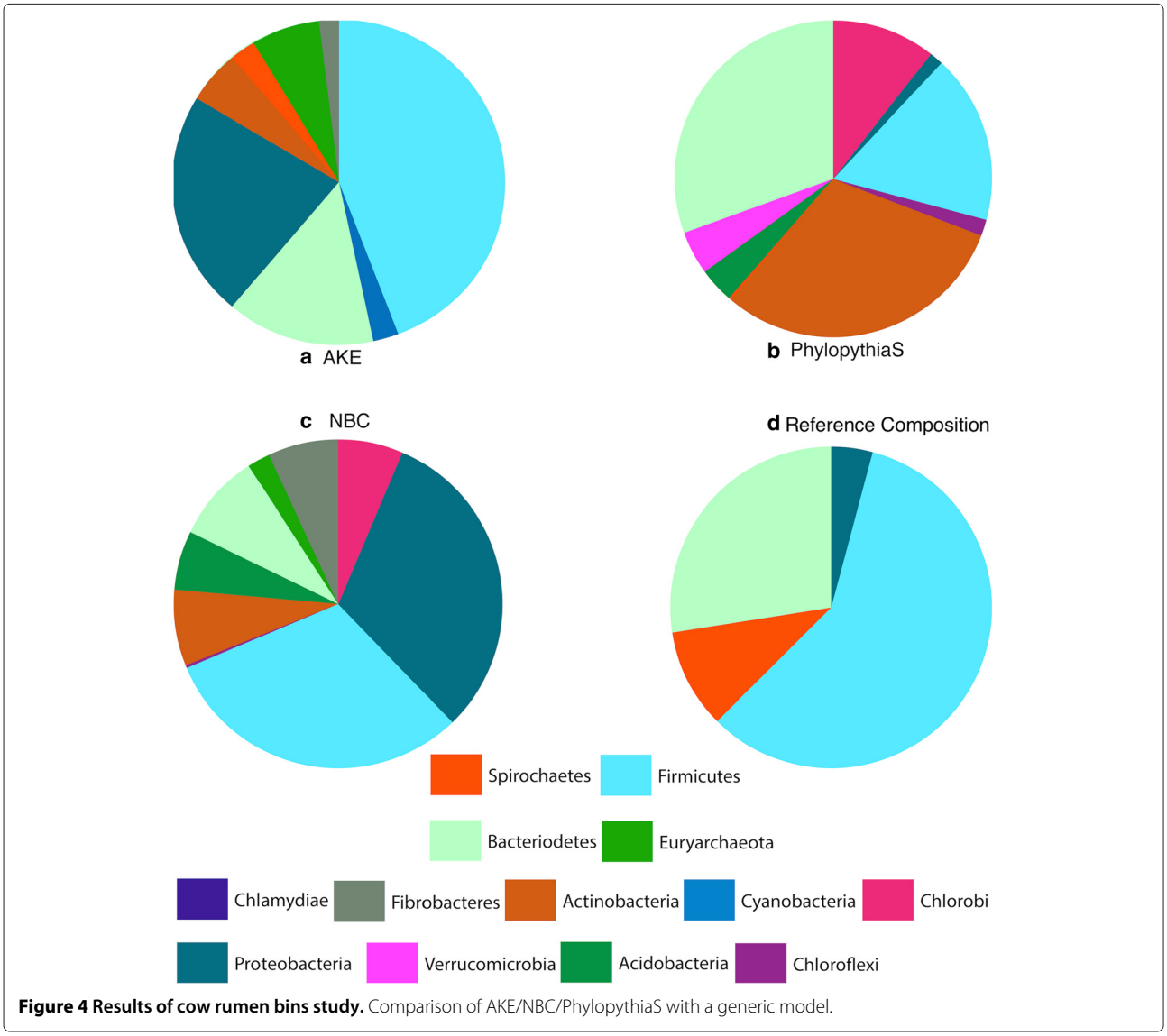

Figure 4 Results of cow rumen bins study. Comparison of AKE/NBC/PhylopythiaS with a generic model.

Table 1 Execution times of AKE

Data set	#Sequences	Megabases	Runtime (*k*-mers) in s	Runtime (Assignment) in s
AMD	1183	10.83	0.63	0.43
Cow rumen (bins)	466	34.14	0.71	0.21
Cow rumen (scaffolds)	26042	568.59	13.68	10.4

Measures for typical metagenome data sets.

We compared AKE with some similar approaches including NBC [12] and PhyloPythiaS [11] with generic and sample specific model. All results were obtained using a model derived from the 15 Kb data set of NCBI genomes mentioned above. We did not explore the possibility to generate a sample specific model as described in [11], but expect it to have a similar positive influence as in the cited study. When using the web service the parameters given above are applied.

The high abundant species are Thermoplasmatales archaeon Gpl (410), Leptospirillum sp. Group II (70), Leptospirillum sp. Group III (474), Ferroplasma acidarmanus Type I (170), Ferroplasma acidarmanus Type II (59). When looking at the results (Figure 3) we see that AKE outperforms NBC and PhylopythiaS (generic model). But it is outperformed by PhylopythiaS employing a sample specific model.

Cow rumen

The Cow Rumen data set consists of a community taken from the deconstruction process of switchgrass in a cow rumen [28]. The cited study could identify 15 draft genomes with completeness between 60% and 93%. On the phylogenetic level of order these samples are comprised of Spirochaetales, Clostridiales, Bacteroidales and Myxococcales. Since a gold standard for all scaffolds does not exist, this reference composition (see Figure 4d)) has to be taken as a rough estimate. The data has been received from NERSC Science Gateways (http://portal.nersc.gov/project/jgimg/CowRumenRawData/submission/). An assignment for the genomic bins (cow_rumen_genome_bins.tar.gz) as well as for the scaffolds (cow_rumen_fragmented_velvet_assembly_scaffolds.fas.gz) is provided. We compared PhylopythiaS (generic model) and NBC with AKE. When looking at the results (Figure 4) we see that AKE outperforms NBC and predicts slightly better than PhylopythiaS.

Online resources

Please note that further classification results are provided online within AKE. These include the results of the AMD and cow rumen data sets with classification down to order as well as a reference composition for these data sets visualized with AKE. Furthermore, the analysis of simulated data sets [29] is provided.

Execution times

The application is written in Python using a C extension for fast computation. The authors implemented the *k*-mer counting as well as the H^2SOM. The execution times are measured using Python's *time()* function. All experiments were repeated ten times and the mean value of this is stated below. The machine used, is the same web server

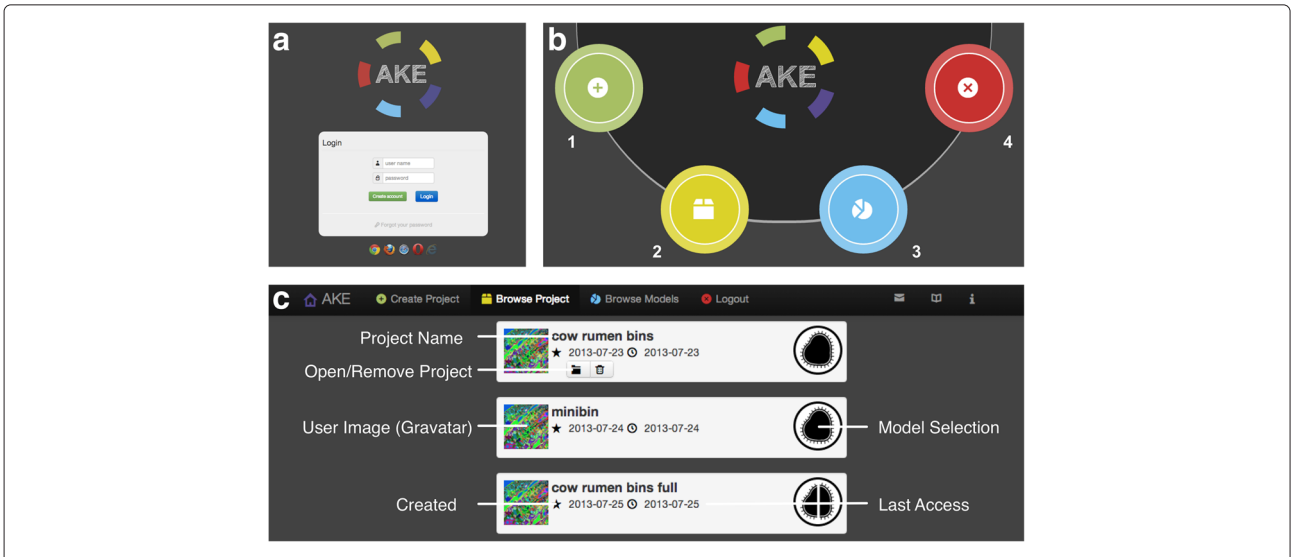

Figure 5 Overview of AKE. a) Login as well as registration and password retrieval can be done here. **b**) Landing Page of AKE: Here all important pages are accessible directly. 1) Create a Project 2) Browse Projects 3) Browse Models 4) Logout **c**) Project View: Basic project management features are provided.

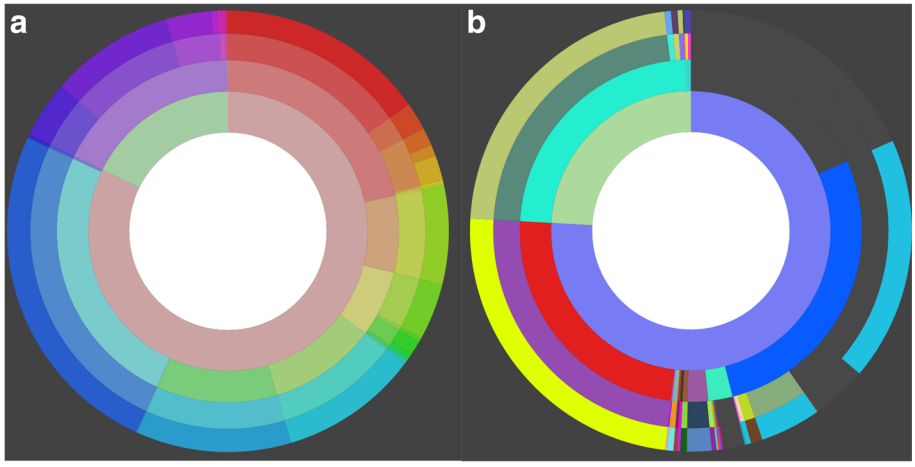

Figure 6 AKE results view with coloration options. a) The entities, i.e. taxonomic categories are colorized according to the position on the disc. **b**) Every entity is colorized in a predefined way.

that serves the results for the web interface. It is a virtual machine running two Intel Xeon E5450 CPUs at 3 GHz with 32 GB main memory operated by Sun Solaris 10. The application is multi-threaded using 4 threads.

The execution times are dominated by the counting of *k*-mers, which is heavily influenced by I/O load on the system (see Table 1). For faster loading all data resides on a tmpfs filesystem (a RAMdisk like filesystem). It is to note that the times were measured with a standalone non-CGI application. A little overhead using CGI can be expected as well as some time for uploading of data.

The web-application

The web-interface is accessible at www.ani.cebitec.uni-bielefeld.de/ake. The website is protected by a login screen (Figure 5a). A login with password can be chosen on this page. The browsers, which are known to work properly with AKE are indicated at the bottom of the page. After login the user is redirected to the landing page (Figure 5b) where every subpage is accessible. A basic project management – creation, removal, storage of basic information (date, last access and model selection) for the creation of the project – is supported (Figure 5c).

During project creation two modes of operation can be chosen. The preview mode is for receiving a fast result for data sets smaller than 100 MB. Here the results are computed immediately. For larger files, which need more computation time, the classification mode can be used. The computation is done on a powerful machine in this

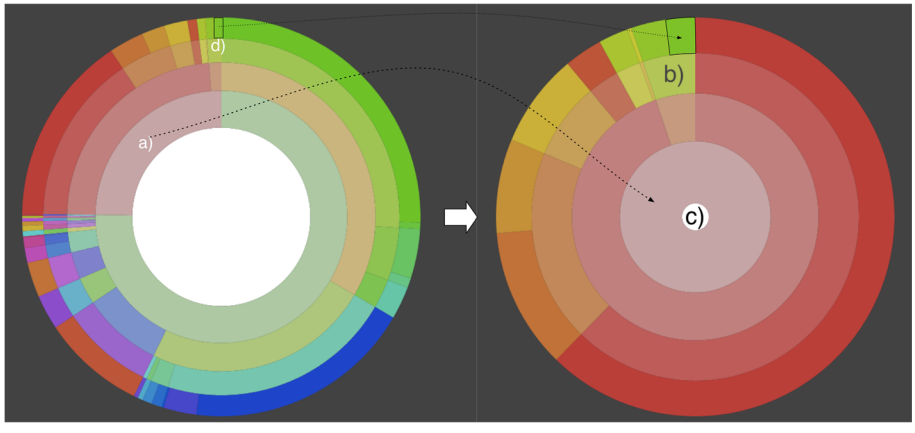

Figure 7 Zooming in of AKE results view. When clicking on one category like a) the disc is transformed. The activated category is the new root and the new whole disk is used to display only its descendant results in the taxonomic subtree. This helps in exploring smaller entities (e.g. see d)) on lower ranks. In addition, the amount of information is reduced so that the remaining one can be accessed more easily. With a click on c) one can go back to the prior view. Furthermore, one can skip taxonomic ranks on zooming in. It is not only possible to zoom in on the next rank but on arbitrary levels of the hierarchy e.g. by clicking on b).

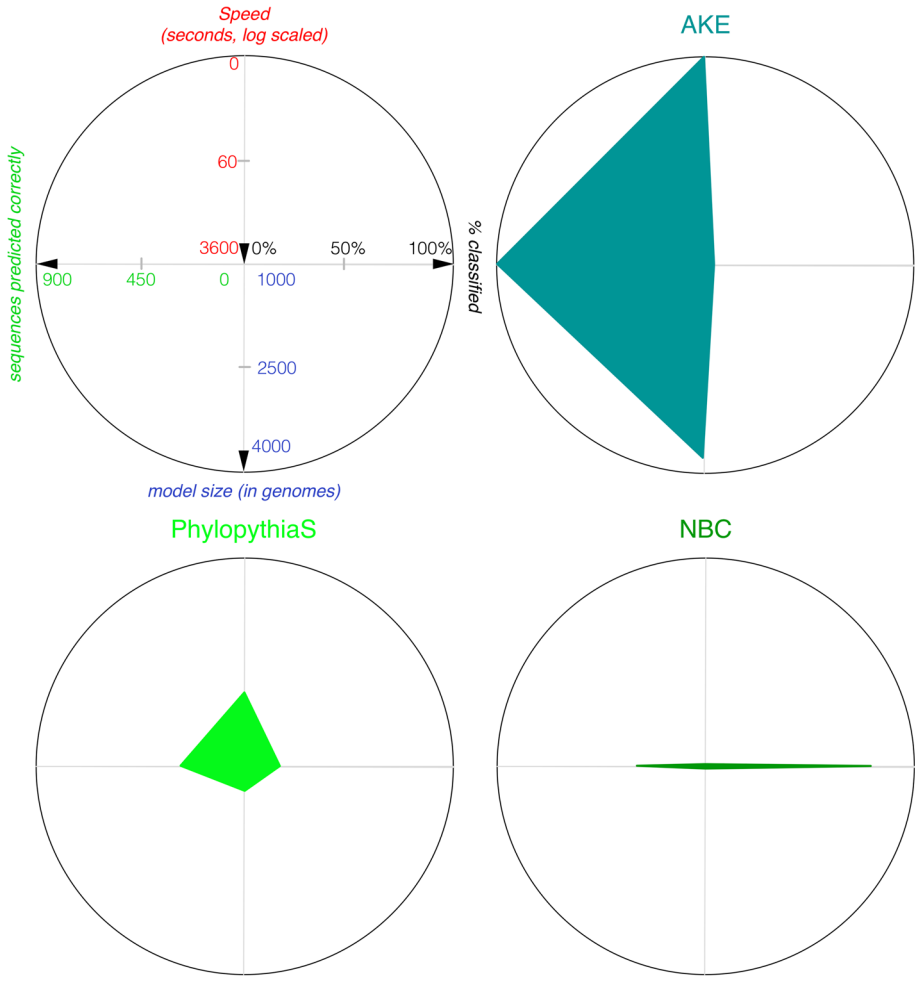

Figure 8 Comparison of web-based taxonomic classifiers for AMD data set. The data is based on [12], the respective web sites and personal measures. Note that Speed (runtime in seconds) is depicted in reverse order. Predicted correctly is the number of predictions which are also present in the reference composition. Model size is the number of genomes which are included in the model building process. The percent classified value expresses the inverse of the percentage of rejected data. (Note that differences between Figure 3 (data is cited) and this Figure (data is measured, July 2014) are most probably due to different application versions).

mode but is not guaranteed to start immediately, so that the user will get notified by email when all results are computed. The Projects' assignment visualizations contain a Krona [30] inspired view. For this view two different colorization options are available (Figure 6). One option colorizes every item in a specific predefined color. This is especially helpful to compare two different results as entities, because taxonomic categories are colorized consistently across results. The other option is helpful when looking at only one result and colorization is inspired by the HSV color wheel. It helps in retaining orientation when zooming in (see Figure 7). The zoom enables the user to interactively browse the classification results. By clicking on a category, it becomes the new root of the visualization. This allows the inspection of small entities and interesting subtrees. For visualization the D^3 framework [31] was used. Here the so-called sunburst tree is generated with the automatic D^3 partition layout. A client-server architecture is used with the back end written in Python with C-extensions. The communication is done via JSON.

Table 2 Performance comparison of PhylopythiaS, NBC, WebCarma and AKE for AMD data set on phylum level

Algorithm	#Correct assignments	#Wrong assignments	#Unknown assignments	Runtime (assignment) in s
AKE	902	213	68	0.43
NBC	218	717	248	3480
PhylopythiaS	105	411	832	207
WebCarma	678	13	492	$6 * 10^5$

Conclusion

A comparison of web-based taxonomic classifiers is shown in Figure 8 based on the analysis of the AMD data set. AKE outperforms PhylopythiaS [11] (generic model) and NBC [12] in all measured categories and the execution time is one (PhylopythiaS) or two (NBC) orders of magnitude faster. A result with WebCarma [8], which is a homology-based classifier, has been obtained within about a week. It outperforms all composition-based methods, with 678 correct assignments, except our system AKE (902 correct assignments) on phylum level. The number of rejects of WebCarma, i.e. the assignment to an "other" unknown class, on phylum level (42%) is comparable to PhylopythiaS but it is much higher than in NBCs or AKEs results. The detailed results are given in Table 2.

The evaluation of different web-based taxonomic classifiers shows that the runtime differs dramatically from a second (AKE), to minutes (PhylopythiaS), to an hour (NBC), to almost a week (WebCarma) due to algorithmic features and implementation details. AKE is faster compared to the other applications because it only needs to compute the euclidean distance between the descriptive model and the data that should be classified, whereas the others need to compute alignments (WebCarma) or apply decision functions (Phylopythia, NBC). Furthermore, optimized C code and multi-threading accelerates the application. The neural network used is especially suited to generate a hierarchical, compact, descriptive model, which allows fast queries using a beam search to limit the number of euclidean distance searches. Although there might be methods reported to be equally fast and more accurate, to the authors knowledge there exists no web-based solution which performs equally well, in terms of execution time *and* accuracy for generic metagenome data. Since accuracy drops down significantly for ranks lower than order we do not report these here, since our focus in development lay on acceleration and a dynamic web-based visualization system.

AKE is a fast taxonomic assignment tool for first visual inspection of whole metagenome data sets. Its web-based dynamic visualization allows fast analyses even on low performance computers without installation of software. Furthermore, the web-based approach enables a cooperative analysis of data with colleagues.

Additional files

Additional file 1: Detailed description of H^2SOM. PDF file giving a detailed description of the H^2SOM algorithm. Open with you favorite pdf reader, e.g. Adobe Reader.

Additional file 2: Table for cross validation study. PDF file presenting results for the cross validation study. Open with you favorite pdf reader, e.g. Adobe Reader.

Additional file 3: List of GI numbers of sequences used for training. Text file listing the gi numbers of the sequences used for training. Unzip and open with your favorite text editor.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

DL, AG and TWN participated in the design of the study. DL implemented the study. DL and TWN analyzed and interpreted the data. DL, AG and TWN prepared the manuscript and revised it. All authors read and approved the final manuscript.

Acknowledgements

Data for AMD comparison study except AKE results kindly provided by Kaustubh Patil and Alice McHardy. This work was supported by the German Federal Ministry of Education and Research [grant 01|H11004 "ENHANCE"] to Daniel Langenkämper. We acknowledge support of the publication fee by Deutsche Forschungsgemeinschaft and the Open Access Publication Funds of Bielefeld University.

Author details

[1]Biodata Mining, Bielefeld University, Universitätsstraße 15, Bielefeld, Germany. [2]Bioinformatik und Systembiologie, Justus Liebig University, Düsternbrooker Weg 20, Gießen, Germany.

Received: 17 July 2014 Accepted: 12 November 2014
Published online: 13 December 2014

References

1. Nakao R, Abe T, Nijhof AM, Yamamoto S, Jongejan F, Ikemura T, Sugimoto C: **A novel approach, based on BLSOMs (batch learning self-organizing maps), to the microbiome analysis of ticks.** *ISME J* 2013, **7**(5):1003–1015. doi:10.1038/ismej.2012.171.
2. Teeling H, Gloeckner FO: **Current opportunities and challenges in microbial metagenome analysis-a bioinformatic perspective.** *Brief Bioinform* 2012, **13**(6):728–742. doi:10.1093/bib/bbs039.
3. Liu Z, DeSantis TZ, Andersen GL, Knight R: **Accurate taxonomy assignments from 16s rrna sequences produced by highly parallel pyrosequencers.** *Nucleic Acids Res* 2008, **36**(18):120–120.
4. Koslicki D, Foucart S, Rosen G: **Quikr: a method for rapid reconstruction of bacterial communities via compressive sensing.** *Bioinformatics* 2013, **29**(17):2096–2102.
5. Kunin V, Copeland A, Lapidus A, Mavromatis K, Hugenholtz P: **A bioinformatician's guide to metagenomics.** *Microbiol Mol Biol Rev* 2008, **72**(4):557–578.
6. Huson DHD, Mitra SS, Ruscheweyh H-JH, Weber NN, Schuster SCS: **Integrative analysis of environmental sequences using MEGAN4.** *Genome Res* 2011, **21**(9):1552–1560. doi:10.1101/gr.120618.111.
7. Gerlach W, Jünemann S, Tille F, Goesmann A, Stoye J: **WebCARMA: a web application for the functional and taxonomic classification of unassembled metagenomic reads.** *BMC Bioinformatics* 2009, **10**(1):430. doi:10.1186/1471-2105-10-430.
8. Gerlach W, Stoye J: **Taxonomic classification of metagenomic shotgun sequences with CARMA3.** *Nucleic Acids Res* 2011, **39**(14):e91. doi:10.1093/nar/gkr225.
9. Meyer F, Paarmann D, D'Souza M, Olson R, Glass EM, Kubal M, Paczian T, Rodriguez A, Stevens R, Wilke A, Wilkening J, Edwards RA: **The metagenomics RAST server - a public resource for the automatic phylogenetic and functional analysis of metagenomes.** *BMC Bioinformatics* 2008, **9**(1):386. doi:10.1186/1471-2105-9-386.
10. McHardy AC, Martín HG, Tsirigos A, Hugenholtz P, Rigoutsos I: **Accurate phylogenetic classification of variable-length DNA fragments.** *Nat Methods* 2006, **4**(1):63–72. doi:10.1038/nmeth976.
11. Patil KR, Roune L, McHardy AC: **The PhyloPythiaS web server for taxonomic assignment of metagenome sequences.** *Plos One* 2011, **7**(6):38581–38581. doi:10.1371/journal.pone.0038581.
12. Rosen GLG, Reichenberger ERE, Rosenfeld AMA: **NBC: the Naive Bayes classification tool webserver for taxonomic classification of**

metagenomic reads. *Trans IRE Professional Group Audio* 2010, **27**(1):127–129. doi:10.1093/bioinformatics/btq619.
13. Rasheed Z, Rangwala H: **Metagenomic taxonomic classification using extreme learning machines.** *J Bioinform Comput Biol* 2012, **10**(5):1250015. doi:10.1142/S0219720012500151.
14. Weber M, Teeling H, Huang S, Waldmann J, Kassabgy M, Fuchs BM, Klindworth A, Klockow C, Wichels A, Gerdts G, Amann R, Glöckner FO: **Practical application of self-organizing maps to interrelate biodiversity and functional data in NGS-based metagenomics.** *ISME J* 2010, **5**(5):918–928. doi:10.1038/ismej.2010.180.
15. Brady A, Salzberg SL: **Phymm and phymmbl: metagenomic phylogenetic classification with interpolated markov models.** *Nat Methods* 2009, **6**(9):673–676.
16. Wood D, Salzberg S: **Kraken: ultrafast metagenomic sequence classification using exact alignments.** *Genome Biol* 2014, **15**(3):46.
17. Meinicke P, Aßhauer K P, Lingner T: **Mixture models for analysis of the taxonomic composition of metagenomes.** *Bioinformatics* 2011, **27**(12):1618–1624.
18. Foerstner KUK, von Mering CC, Hooper SDS, Bork PP: **Environments shape the nucleotide composition of genomes.** *EMBO Rep* 2005, **6**(12):1208–1213. doi:10.1038/sj.embor.7400538.
19. Karlin S, Mrazek J: **Compositional differences within and between eukaryotic genomes.** *Proc Natl Acad Sci U S A* 1997, **94**(19):10227–10232.
20. Deschavanne PJ, Giron A, Vilain J, Fagot G, Fertil B: **Genomic signature: Characterization and classification of species assessed by chaos game representation of sequences.** *Mol Biol Evol* 1999, **16**(10):1391–1399.
21. Martin C, Diaz NN, Ontrup J, Nattkemper TW: **Hyperbolic SOM-based clustering of DNA fragment features for taxonomic visualization and classification.** *Bioinformatics* 2008, **24**(14):1568–1574. doi:10.1093/bioinformatics/btn257.
22. Markowitz VM, Chen I-MA, Palaniappan K, Chu K, Szeto E, Grechkin Y, Ratner A, Jacob B, Huang J, Williams P, Huntemann M, Anderson I, Mavromatis K, Ivanova NN, Kyrpides NC: **IMG: the Integrated Microbial Genomes database and comparative analysis system.** *Nucleic Acids Res* 2011, **40**(Database issue):115–122. doi:10.1093/nar/gkr1044.
23. Kohonen T: **Self-organized formation of topologically correct feature maps.** *Biol Cybern* 1982, **43**(1):59–69. doi:10.1007/BF00337288.
24. Ritter H: **Self-organizing maps on non-euclidean spaces.** *Kohonen Maps* 1999, **73:**97–110.
25. Ontrup J, Ritter H: **A hierarchically growing hyperbolic self-organizing map for rapid structuring of large data sets.** In *Proceedings of the 5th Workshop on Self-Organizing Maps, Marie Cottrell (Paris 1 Panthéon-Sorbonne University)*. Paris (France); 2005.
26. Martin C, Diaz NN, Ontrup J: **Genome feature exploration using hyperbolic self-organising maps.** In *6th international workshop on self-organizing maps WSOM*; 2007.
27. Tyson GW, Chapman J, Hugenholtz P, Allen EE, Ram RJ, Richardson PM, Solovyev VV, Rubin EM, Rokhsar DS, Banfield JF: **Community structure and metabolism through reconstruction of microbial genomes from the environment.** *Nature* 2004, **428**(6978):37–43. doi:10.1038/nature02340.
28. Hess M, Sczyrba A, Egan R, Kim T-W, Chokhawala H, Schroth G, Luo S, Clark DS, Chen F, Zhang T, Mackie RI, Pennacchio LA, Tringe SG, Visel A, Woyke T, Wang Z, Rubin EM: **Metagenomic discovery of biomass-degrading genes and genomes from cow rumen.** *Science* 2011, **331**(6016):463–467. doi:10.1126/science.1200387.
29. Mavromatis K, Ivanova N, Barry K, Shapiro H, Goltsman E, McHardy AC, Rigoutsos I, Salamov A, Korzeniewski F, Land M, Lapidus A, Grigoriev I, Richardson P, Hugenholtz P, Kyrpides NC: **Use of simulated data sets to evaluate the fidelity of metagenomic processing methods.** *Nat Med* 2007, **4**(6):495–500.
30. Ondov BDB, Bergman NHN, Phillippy AMA: **Interactive metagenomic visualization in a Web browser.** *BMC Bioinformatics* 2010, **12:**385–385. doi:10.1186/1471-2105-12-385.
31. Bostock M, Ogievetsky V, Heer J: **D^3 data-driven documents.** *IEEE Trans Vis Comput Graph* 2011, **17**(12):2301–2309.