bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Distributed representations of protein domains and genomes and their compositionality

A. Viehweger [1–3], S. Krautwurst [1], B. König [2], M. Marz [1]

(1) Bioinformatics
Faculty of Mathematics and Computer Science
Friedrich Schiller University Jena
Leutragraben 1
07743 Jena

(2) Institute of Medical Microbiology
University Hospital Leipzig
Liebigstraße 21
04103 Leipzig

(3) Corresponding author: adrian.viehweger@uni-jena.de

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Abstract

Learning algorithms have at their disposal an ever-growing number of metagenomes for biomining and the study of microbial functions. We propose a novel representation of *function* called `nanotext` that scales to very large data sets while capturing precise functional relationships.

These relationships are learned from a corpus of 32 thousand genome assemblies with 145 million protein domains. We treat the protein domains in a genome like words in a document, assuming that protein domains in a similar context have similar “meaning”. This meaning can be distributed by the `Word2Vec` embedding algorithm over a vector of numbers. These vectors not only encode function but can be used to predict even complex genomic features and phenotypes.

We apply `nanotext` to data from the Tara ocean expedition to predict plausible culture media and growth temperatures for microorganisms from their metagenome assembled genomes (MAGs) alone.

`nanotext` is freely released under a BSD licence (https://github.com/phiweger/nanotext).

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Introduction

An organism can be reduced to the functions its genome encodes. However, the definition of function and its representation remain elusive[1,2]. Protein domains in a genome are basic units of function, like words are basic units of "meaning" in a document. Embeddings of protein domains in a vector space are a novel representation that captures even subtle aspects of function. When extended to entire genomes, functional "topics" of these genomes can be inferred, which reflect their current taxonomy. Domain and genome embeddings have many useful properties especially as input to learning algorithms and offer the possibility for use in large scale metagenomic applications such as biomining and genotype-phenotype mapping.

In metagenomics, the bottleneck of discovery has shifted from data generation to analysis. Many current sequencing efforts are extremely data-intensive, regularly reconstructing thousands of unknown genomes in a single study[3–8]. Gene catalogs compiled from metagenomes have millions to billions of records[9,10], many without a documented functional role[11]. This wealth of data holds tremendous potential, from substantially revising the tree of life[12], the discovery of new enzymes and metabolites for biotechnological use[13] to predictive models that distinguish diseases based on microbial composition[14]. To adress these questions, learning algorithms such as *neural nets* are powerful pattern detection tools[15]. Learning is most effective when the signal in the data is "stable", i.e. if given a similar input, the target variable is similar too. Such a stable signal has been found in the functions performed by a microbial community, rather than in its taxonomic makeup[16,17], although this view is debated[18]. To "fit" metagenome-derived functions into learning algorithms, two questions need to be answered: (1) How is "function" defined? (2) How is it represented?

(1) Protein-mediated function can be defined as a sequence of protein domains. Domains are typically identified as highly conserved regions in a multiple alignment of similar protein sequences[19,20]. Most proteins have two or more domains and the nature of their interactions determines the protein's function(s)[21]. Although chemically, the basic building blocks of proteins are amino acids, protein domains are arguably the basic units of "meaning". This is supported by their independent evolution[21–23] and by the fact that the structure of domains is often more conserved than their amino acid sequence[20,24], especially in viruses[25].

(2) Many representations of function exist. *Zhu et al.* used a network-based approach to assign functional similarity to pairs of genomes on the basis of encoded proteins[26,27]. Other approaches use direct counts of protein domains to distinguish organisms[28,29]. Both approaches discard context information, which is very important in bacterial and fungal genomes: Not only are genes

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

frequently co-located in e.g. biosynthetic gene clusters (BGCs)[30] or polysaccharide utilisation loci (PULs)[31], but often they are situated in *polycistronic* open reading frames (ORFs)[32]. Multiple adjacent ORFs are frequently regulated in concert by expression as a single mRNA[33], adding further context dependence. Count-based representations have another disadvantage; they are high-dimensional and sparse. To encode the count of a protein domain out of the 17 thousand domains in the `Pfam` database[19], the resulting *one-hot-encoded* vector would have an equal number of dimensions with all elements zero except one. Such sparse vectors can make learning very inefficient.

A representation that both preserves the context information and results in dense vectors are *word embeddings*[34,35]. They assign words that occur in similar contexts to similar vectors in vector space. The assumption then is that words with similar vectors have similar "meaning". Indeed, word embeddings have been shown to capture precise syntactic and semanic relationships in text such as synonyms. Word embeddings are trained on a large collection of unlabeled texts (*corpus*). Training an embedding results in a vector of numbers for each distinct word in the corpus (*vocabulary*). Different training algorithms exist, the most popular of which is `Word2Vec`[36,37]. Several extensions have been developed: For example, character information can be included in the embedding model[38] or it can be extended to entire documents to create "topic" vectors[39,40]. Similar words or topics can be identified using the *cosine similarity* of the associated vectors. Because word and document vectors capture similarity, they are effective as input for learning algorithms and facilitate training. Without such a "language model", a learning algorithm would have to learn about syntax and semantics in parallel to the actual learning task. However, pretrained embeddings already hold this information.

Embeddings have been trained on biological objects such as genes[41,42], proteins[43,44], chemical structures[45] and nucleotide sequences[46–48]. Most of these approaches focus on the primary sequence. However, as discussed above, structure is oftentimes conserved although the underlying sequence is not. Furthermore, many sequence variations do not affect function, but act as noise during training, for example in the case of synonymous single nucleotide polymorphisms (SNPs). In this article, we asked how an organism's functions might be representable in vector space in such a way as to facilitate downstream learning tasks. To approach this question, we trained a vector representation of protein-mediated function on a large, diverse collection of bacterial genomes and their protein domain annotations. The result is a pre-trained embedding model called `nanotext`. We then investigated which functional aspects are captured by the embedding vectors and finally applied the embedding to several unsolved learning tasks.

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Results

Embeddings of protein domains capture functional relationships

To train a protein domain embedding, we aggregated sequences of `PFAM` domains[19] into a *corpus* of 32 thousand bacterial genomes with 145 million annotated domains. The set of domains in the corpus forms the *vocabulary* and is comprised of about 10 thousand domains. Training resulted in a vector *representation* of size 100 for each unique protein domain and genome in the corpus. We make the resulting pre-trained model available as `nanotext`. Each domain vector is comprised of latent features, which describe the associated domain's functional meaning along multiple dimensions.

Protein domain embeddings can distinguish functional context with near-perfect accuracy. Generally, embedding accuracy can be tested using a variety of tasks[50]. However, no single task captures all aspects of the representation, because embeddings capture meaning, and meaning is multifaceted. Specific assessment tasks usually rely on labelled datasets e.g. of synonyms. No such dataset exists for protein domains. We therefore estimate embedding accuracy using the *semantic odd man out* (SOMO) task[51]: For a set of words, we try to identify the one that does not "fit" into the context. For example, "Cereal" would be *odd* in a set comprising "Zebra", "Lion" and "Flamingo". For each ORF in our corpus with more than one domain, we select a random domain from the vocabulary. The mean of the embedding vectors of this set is then calculated. The "odd" domain is the one with the largest cosine distance to this mean, and in the correct case corresponds to the randomly chosen domain. We achieve a 99.27% accuracy on the SOMO task, which is much higher compared to embeddings generated from natural language texts[51].

Many domain vectors cluster according to known functional classes, which we derived from an existing mapping of protein domains to putative enzyme functions[20]. To visualize clusters, we projected all associated domain vectors into two dimensions using the t-SNE visualization algorithm[52]. We found that many domains cluster according to their enzyme function label (Figure 1), while others do not. This might reflect that many domains have several functions and that those functions can overlap. However, the observed clustering is indicative that the domain embeddings are plausible.

Domain vectors can be used to explore *domains of unknown function* (DUF). We illustrate this with a case study of `DUF1537`: Since its introduction to `Pfam` as a protein family of unknown function, experiments have identified it as ATP-dependent four-carbon acid sugar kinase with now two associated domains – `PF07005` and `PF17042`[53]. *Zhang et al.* used a gene cooccurence network to identify "conserved genome neighborhoods". Querying our embedding model for functionally similar domains to

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

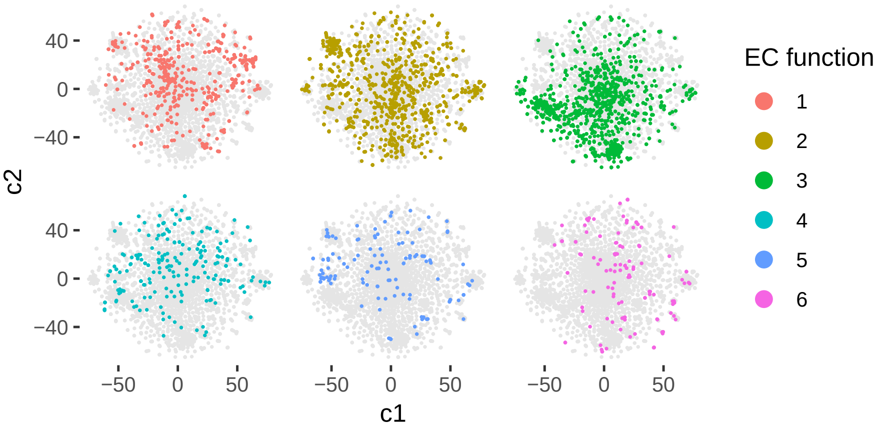

Figure 1: Supplement. Domain vectors cluster according to known functional classes. Projection of the domain vectors from 100 dimensions into two using t-SNE. Some clusters correspond to a single functional class (*enzyme commission* (EC) numbering scheme), which suggests that the learned domain embeddings capture functional relations.

`PF07005` and `PF17042` (because `DUF1537` has since been removed), we find exactly the same "conserved" domains as *Zhang et al.* (Table 1). When we query the embedding model with `PF07005` (`SBD_N`) for its closest vector, we find `PF17042` (`NBD_C`) and vice versa, with a cosine similarity of the associated word vectors of 0.93, respectively.

Composing domain vectors creates new meaning. A surprising result of the original work on word vectors was that they capture linguistic regularities, which can be composed using vector algebra[36]. For example, vector("king") - vector("man") + vector("woman") is close to vector("queen")[36]. These semantic regularities are captured by protein domain embeddings, too. For example, the vector for the enzyme urease (`Urease_beta`, `PF00699`) minus its N-terminal domain (`Urease_alpha`, `PF00449`) plus the catalytic domain of ribulose bisphosphate carboxylase (large chain, `RuBisCO_large`, `PF00016`) results in a vector whose nearest neighbor is the N-terminal domain of the carboxylase (`RuBisCO_large_N`, `PF02788`, cosine similarity 0.93).

Functional similarity captures taxonomic properties

A genome can be abstracted as a sequence of protein domains, or by analogy as a document containing words. Embeddings of genomes result in a type of *topic model*[40] with an associated *topic vector* composed of latent features. The topic of a document might be how much "sports" or "politics" it contains, while the topic of a genome might reflect how anaerobic an organism is or which metabolic constraints it operates under. Note that a topic is merely a cluster of document vectors in embedding

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

space. It is not assigned a label, because it is learned from unlabelled data. We furthermore introduce the term *functional similarity* analogous to nucleotide similarity, to describe the distance between any two genome vectors as measured by their cosine similarity.

Genome embeddings can be used to assign genomes to taxa. Unlike protein domain vectors, genome vectors can be inferred for previously unseen, *out of vocavulary* (OOV) genomes. To illustrate this, we used a collection of 957 metagenome assembled genomes (MAGs) based on data from the *Tara Ocean Expedition*[3,7]. These MAGs did not feature in our embedding training set or in reference databases such as `RefSeq`[54]. Using unknown MAGs imitates the use case of biomining newly sequenced metagenomes. We would expect genome vectors to cluster according to their taxonomy, because organisms with the same taxonomic label frequently share many functions. To visualize this, we projected the genome vectors into two dimensions using t-SNE[52]. We identify clearly delineated clusters that can be assigned to distinct phyla (Figure 2, A). The clustering is hierarchical as to taxonomic rank, in the sense that clusters of e.g. *phyla* are themselves composed of clusters of distinct *classes* (Figure 3, A). Interestingly, many MAGs could not be assigned a taxonomic rank by *Delmont et al.* using marker genes[7], but have their genome vector cluster clearly with known organisms (Figure 3, B). Genome vectors could be a complement if not replacement for marker gene-based approaches, without the need to select these genes based on prior knowledge[55].

Unlike marker-gene based approaches, genome vectors are remarkably stable when MAGs are incomplete. From the *Delmont et al.* high-quality, near-complete MAGs, we successively removed an increasing percentage of contigs *in silico*, inferred genome vectors, and then identified their nearest neighbors in vector space. We found that the functional similarity of "truncated" genome vectors to their "complete" self decreases only slowly with increasing degrees of incompleteness (Figure 2, B). For an illustrative MAG (`TARA_RED_MAG_00040`), we find that up to 90% of contigs can be removed until the corresponding genome vector moves notably in embedding space (Figure 2, C). Thus `nanotext` can assign taxonomy to even highly incomplete genomes.

Functional and nucleotide similarity are complementary measures of how different two genomes are. For some genomes, both measures correlate (Figure 2, D). However, there are pairs of genomes with low nucleotide similarity but high functional similarity (Figure 2, D). In these cases, both measures offer complementary information. Investigating such a cluster, we found three genomes which in the original study could not be assigned to a taxon below the rank of *domain Bacteria*. Based on functional similarity however, these genomes were clearly related, while they would not have been grouped by their nucleotide similarity alone (Table 2). We could confirm that the three genomes were of the same order *Gemmatimonadales* by searching against a large reference collection of `MinHash` signatures

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

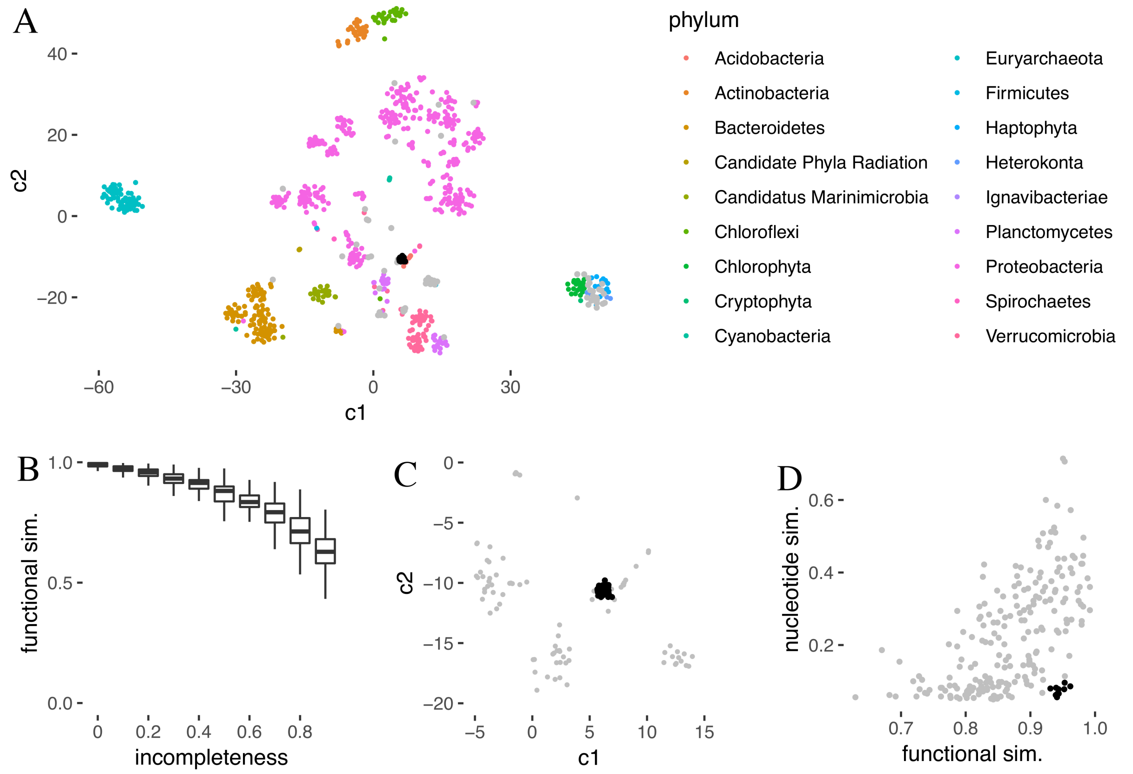

Figure 2: Functional similarity captures taxonomic properties. **(A)** Visualisation of genome vectors using t-SNE projection into two dimensions (components). Clear clusters can be observed which correspond largely to the phylum assigned to the MAGs from which the genome vectors were derived by *Delmont et al.*[7]. Note how archael genomes and algae form separate clusters (left turquoise and bottom right, respectively) although the embeddings were only trained on bacterial genomes. **(B)** Detail from (A): The MAG `TARA_RED_MAG_00040` was truncated by removing an increasingly large, random subset of its contigs. Then, for each truncated genome, the genome vector was inferred and the closest MAG from *Delmont et al.* marked (black points in (A) and (B)). Remarkably, the truncation has little effect on the placement of the genome vector. Up to 90% of contigs can be removed while the associated genome vector remains in the same region in vector space. **(C)** Effect of MAG truncation on functional similarity: For a random subset of 100 MAGs from *Delmont et al.* we removed an increasing percentage of contigs, calculating the cosine similarity between the truncated genome and the original one. It decreases very slowly as genomes are increasingly truncated. This makes cosine similarity an attractive measure of genome similarity in metagenomic contexts where assembled genomes are more often incomplete than not. **(D)** Pairwise comparison of MAGs from *Delmont et al.* between nucleotide (Jaccard) similarity and functional (cosine) similarity. There are several genomes which are very different in terms of average nucleotide identity as approximated from their k-mer composition using `MinHash`[56]. However, some pairs nevertheless exibit high functional similarity (black) which suggests similar taxa. Notably, there are no genomes of high nucleotide but low cosine similarity (upper left triangle), which would be implausible.

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

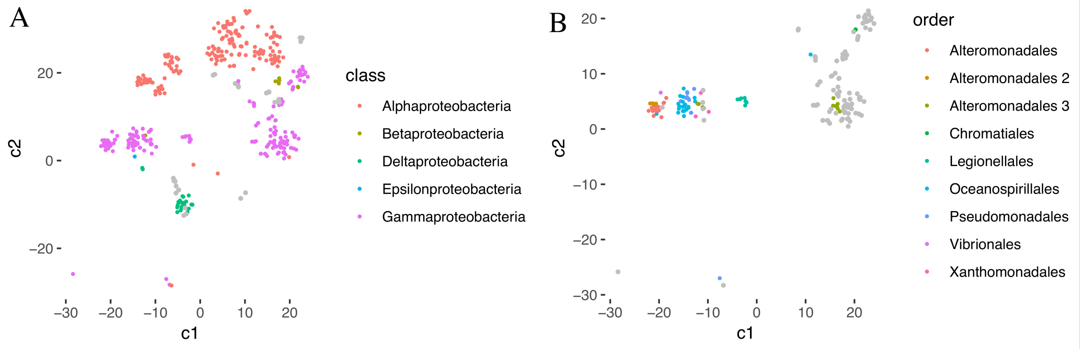

Figure 3: Supplement. Genome vectors cluster hierarchically by taxonomic rank. **(A)** The genome vectors of phylum *Proteobacteria* (pink in Figure 2, A) are labelled according to taxonomic class, and a subset of those vectors (pink) was then labelled by order (according to *Delmont et al.*). **(B)** As was the case for phyla, clusters represent distinct taxonomic entities. At the level of order, many MAGs could not be labelled in the original study, possibly because certain marker genes were missing (grey). However, their proximity to genomes with known taxonomy is clearly informative. Note for example the grey points around the order *Alteromonadales 3* (yellowish green), which could be plausibly grouped with it.

172 (Table 3)[56].

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Table 1: Supplement. Domains found in the neighborhood of the DUF1537 protein family, later to be discovered to confer kinase function (PF07005, PF17042). All contextual domains identified by a previous study[53] can be retrieved from the domain embedding by their high *cosine similarity* to the query vector.

domains	cosine similarity	description
NBD_C	0.93	members of the DUF1537 family
Aldolase_II	0.63	class II aldolase
DctQ, DeoRC	0.53, 0.57	substrate binding proteins of TRAP transporters
KdgT, GntP_permease	0.60, 0.61	permease components of the TRAP transporters
RuBisCO_large	0.56	ribulose 1,5-bisphosphate carboxylase/ oxygenase
PdxA	0.56	4-hydroxy-l-threonine 4-phosphate dehydrogenase

Table 2: Supplement. Pairwise comparison of three MAGs which show low pairwise nucleotide (Jaccard) similarity but high functional (cosine) similarity (see also Figure 2, D). Note how functional similarity is higher than simple protein domain overlap, because it considers the context of individual domains as well.

MAG pair	nucleotide sim.	functional sim.	domain overlap
m05, m40	0.10	0.95	0.83
m05, m42	0.08	0.93	0.70
m40, m42	0.48	0.93	0.71

Table 3: Supplement. Case study MAGs and their closest assembled genomes in NCBI by nucleotide similarity.

MAG	closest assembly	nucleotide sim.	order
TARA_ANE_MAG_00005	UBA2589	0.862	Gemmatimonadales
TARA_RED_MAG_00040	UBA2960	0.744	Gemmatimonadales
TARA_ION_MAG_00042	UBA2960	0.518	Gemmatimonadales

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Genome vectors as inputs for machine learning tasks

Many machine learning algorithms require vectors of numbers as their input. Genome vectors in `nanotext` can be used as direct input to these algorithms without preprocessing or feature engineering. Furthermore, sets of genome vectors can be composed to form new, meaningful topic vectors. A genus or an environment can be described from its constituent genomes, e.g. by simply summing over them. To illustrate this potential, we chose a complex learning tasks which has two components: Given a genome assembly, we want to (1) recommend culture media in which the associated organism is likely to grow, and (2) estimate the growth temperature required for culture from the community composition of the environmental sample. More specifically, task (1) is a genotype-phenotype mapping (classification) and we use a *fully connected neural net* to approach it. Task (2) is a regression for which we use *gradient boosting trees.*

Culture medium prediction

Metagenomics is oftentimes the first window into a microbial environment. However, to study the physiology of individual community members, cultivating a microorganism of interest is very important. While most bacteria are still not culturable, there are recent high-throughput culturing efforts, which are able to culture a surprisingly high number of bacteria[59]. It is likely that many bacteria identified in metagenomics are culturable, but it is difficult (without a deep niche-specific knowledge[60]) to choose among the thousands of medium recipes[61,62]. Furthermore, many of these media are similar, in that they are based upon another or share a significant number of ingredients. It is likely that many similar media can be used to culture a single organism. The notion of "similar media" can be approached using embeddings of medium ingredients[63]. For each of the more than one thousand media in the catalogue of the *German collection of microorganisms and cell cultures* (DSMZ), we trained a 10-dimensional embedding vector. To predict medium vectors from genome vectors, we then had to link two databases, namely the genome assemblies and annotations from the *Genome Taxonomy Database* (GTDB)[12] and matching phenotype records from `BacDive`[62].

Genome vectors can be used to accurately predict appropriate culture media for a given microorganism based on its genome (Figure 4, A). This is perhabs unsurprising, because genome vectors represent a genome's functions which act as a constraint on growth conditions. We used a fully-connected neural net to predict likely media from the catalogue of the DSMZ. Because the result is a medium vector, we can search for similar media using cosine similarity. This provides a good starting point for culture experiments. A common-sense baseline is to always predict the most common label of the data set (medium no. 514), which would result in an accuracy of 0.17, i.e. medium no. 514 represents 17% of

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

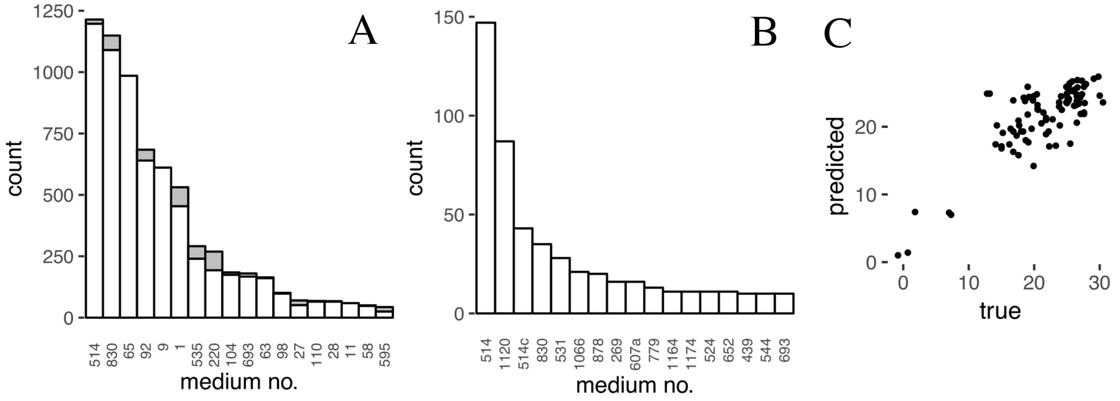

Figure 4: Prediction tasks. **(A)** Prediction of culture media from genomes. Classification task for genotype-phenotype mapping, namely predicting the culture medium for the associated microorganism of a given MAG. Shown is a stacked histogram of the culture media in the `BacDive` database (x-axis) and their count (y-axis). White bars indicate correct predictions, i.e. the target medium is in the top-10 list of closest media compared to the predicted vector. Grey bar fractions indicate false predictions. Only the 20 most common media in the database are displayed on the x-axis. **(B)** Predicted top media for Tara MAGs. The most common media (excluding their variants) in the prediction set are no. 514 ("Bacto Marine Broth"), no. 1120 ("PYSE Medium") – e.g. used to study *Colwellia maris* isolated from seawater[57], no. 830 ("R2A Medium") – developed to study bacteria which inhabit potable water[58], no. 1066 ("Marinobacter Lutaoensis Medium"), no. 878 ("Thermus 162 Medium"), no. 269 ("Acidiphilium Medium") and no. 607 ("M13 Verrucomicrobium Medium") – which includes artificial seawater as an ingredient. All these media are representatives of a "marine topic" and plausible starting media for the organisms associated with the MAGs. **(C)** Inferring the water temperature of the environment for a given set of genomes (regression task). The ten most abundant MAGs from each Tara sampling location (n=93) were used to infer and sum across genome vectors. The resulting aggregate vector was used as input to a *Gradient Boosting Tree* classifier. Water temperature is predicted with an R^2 of 0.66. The dataset is very biased towards moderate temperatures, which likely reduces the predictive accuracy.

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

the media data. A prediction is classified correctly, if the target medium is in the first (1, 10) closest media by cosine similarity, analogous to a common evaluation scheme in multi-class image labelling tasks[15]. On the test set, our model obtains a top-1-accuracy of 63.5% and a top-10-accuracy of 82.5% (Figure 4, A). On the Tara MAGs for which *Delmont et al.* could assign a genus, we obtained a top-1-accuracy of 50% and a top-10-accuracy of 73.2% (Figure 4, B). The lower accuracy on the Tara data is likely due to genomes without a close representative in the training data.

To further assess how well the model generalizes to unseen genome-media pairs, we investigated two cyanobacterial Tara MAGs, which had their genus annotated by *Delmont et al.*, but for which no representative is recorded in `BacDive`: `TARA_ION_MAG_00012` is an MAG that corresponds to the genus *Prochlorococcus*. For this organism, there exist established culture media such as "Artificial based AMP1 Medium"[64]. We were interested in whether our model could predict a similar medium, which could then serve as a starting point for experimentation, were the media in current use unknown. We labelled the AMP1 ingredients according to the protocol established by the `KOMODO` media database[61] and then inferred the target medium vector by summing over the ingredient vectors. Surprisingly, one of the top 10 media predicted for the *Prochlorococcus* MAG – no. 737, "Defined Propionate Minima Medium" (DPM) – has a cosine similarity of 0.979 to the target AMP1 medium. Half of the AMP1 medium ingredients can be found in DPM medium, including vital trace elements. Several non-overlapping ingredients are part of buffers, and can likely be replaced by similar but distinct ingredients. Because our medium embedding can represent such "synonyms", the AMP1 and DPM media are in fact more similar than they appear from shared ingredients alone. A similar generalization of the medium prediction model can be observed for Tara MAG `TARA_ASW_MAG_00003` of the genus *Cyanothece*, which has received considerable attention due to its biotechnological potential[65]. We again encode a common culture medium for this genus – "ASP 2 Medium"[66] – as a medium vector. The predicted medium based on the *Cyanothece* genome is no. 630, "Modified Thermus 162 Medium", with a cosine similarity of 0.98 and again a considerable overlap of ingredients.

Water temperature prediction

Genome vectors can be aggregated into new vectors which represent "topic summaries". Aggregate genome vectors of microbial communities can predict environmental properties. We use the most abundant 10 MAGs in each of the 93 Tara sampling locations to predict the water temperature at each respective sampling site, which is known from the Tara expedition's metadata[3,7]. Besides the fact that the sample size is relatively small and the distribution skewed towards moderate temperatures, we predict the correct water temperature with an R^2 of 0.66 (Figure 4, C).

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Discussion

In this paper, we showed that protein domain and genome embeddings capture many functional aspects of the underlying organism. The main assumption of our approach is that the *function* of a genome can be abstracted as a sequence of protein domains. Much like words determine the topic of a document, protein domains act as atomic units of "meaning" that describe the functions of a genome.

This view of function is very reductive, and much more comprehensive definitions exist[1]. For example, we do not consider functional RNAs[67] or functions that emerge from an interplay of different members of an ecosystem[68,69]. However, our results suggest that this reduced definition of function captures many aspects that are already useful, e.g. for assigning taxa or genotype-phenotype mappings. This success might also be related to our focus on bacteria, where many functions are protein-mediated and the functional mechanisms are much simpler than in eukaryotes. We also completely omit archaea and viruses in this study. However, the embedding model we provide with `nanotext` can be easily extended by including said functional groups in the training corpus.

To expand the corpus with more (non-bacterial) genomes, a major bottleneck is the annotation step. Currently most approaches are based on *Hidden Markov Models* (HMMs)[70;71], which scale poorly to hundreds or even thousands of genomes. Recently, faster homology-based approaches have been proposed[72]. It would be interesting to replace protein domain HMMs with homology-based protein clusters, generated from large collections of metagenomic data such as the *Soil Reference Catalog* (SRC), a catalog assembled from 640 soil metagenomes with two billion protein sequences[10]. With such a large number of sequences, one would need to carefully calibrate the vocabulary size, i.e. the number of protein clusters for the embedding. The `nanotext` embedding was trained with a corpus-to-vocabulary ratio of $10^4 : 1$. To put this into perspective, current corpora in *Natural Language Processing* (NLP) have a ratio above $10^5 : 1$ and well above 100 billion tokens for a vocabulary of about one million words (the English language). Since even billion-scale vector collections can be similarity searched efficiently, scaling to more genomes in the `nanotext` model is not problematic[74]. One further advantage of a vocabulary compiled from protein clusters would be the inclusion of many unknown proteins in the embedding, which – albeit being unknown – could still be used in predictive tasks. Corpora based on metagenomes would further reduce the bacterio-centric bias inherent in our approach, by for example including viral proteins.

For training the embedding models, we used the `Word2Vec` algorithm[36] and its extension to documents[39]. `Word2Vec` is a special case of exponential family embeddings[35], and other embedding methods could be better suited. For the culture medium embeddings for example, a *market basket*

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

embedding might be more appropriate. Domain vectors could be further enriched by "subword information"[38,75,76] i.e. by including nucleotide sequences in the model for inference of out-of-vocabulary words. Embeddings could even be linked across modalities[77]. Note that `Word2Vec` only learns context in a narrow window – of in our case size 10 – and thus cannot learn long-range interactions. However, this is not necessarily a limitation: The embeddings can be used as input for routines that explicitly focus on such long-range interactions. Besides these potential improvements, our embedding model already captures a surprising number of precise and subtle functional properties because it is context-aware, which other metrics like *percentage domain overlap* are not.

We showed, that genome embeddings capture functional and by extension taxonomic properties of the underlying genomes. It would be interesting to extend this work by creating a purely "functional taxonomy", i.e. one based only on genome vectors. Such an approach would assign taxa based on whether certain genes were present or not, also known as *gene exclusivity*[78]. By extension, it should be possible to explore pangenomes using genome vectors. For example, we expect genera with an open pangenome such as *Klebsiella* to present more genome vector variance than genera with closed pangenomes such as *Chlamydia*[79].

Functional similarity-based pangenome studies could further be complemented with nucleotide similarity search. This combination offers orthogonal viewpoints on the relatedness of organisms, with potentially higher resolution than currently possible.

We also illustrated how downstream machine learning tasks benefit from embeddings as input. Not only are embedding vectors convenient mathematical objects. Multiple embedding vectors can be combined to represent e.g. individual genera or bacterial communities, which can then be used to create genotype-phenotype mappings. We illustrated this by predicting likely culture media for assembled genomes. Surprisingly, the notion of embedding similarity allows our predictive model to generalize to genomes and media that were neither part of the training nor test data. Because only very limited data exists where genome assemblies are directly linked to culture media, we had to create a genus-based mapping between the `AnnoTree` genome collection[29] and the `BacDive` database[62]. This compromise likely reduces the predictive power of the learned model. However, as several strain collections started to whole-genome sequence their inventory – such as the DSMZ and the *Japan Collection of Microorganisms* (JCM, http://jcm.brc.riken.jp/en/genomelist_e) – we can expect a much more accurate genotype-phenotype mapping when methods such as `nanotext` are applied.

More generally, learning algorithms can become much more efficient when using embeddings as input, because the algorithms can focus on the actual learning task and need not learn the "semantics" of the problem in parallel. If for example we used raw nucleotide sequences as input to a learning algorithm,

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

it would have to learn concepts such as synonymous SNPs, which embeddings have already encoded. Thus, embeddings reduce the amount of training data required and given a dataset of the same size will oftentimes result in faster, better learning. If needed, pretrained embeddings can be additionally trained on a downstream domain-specific learning task, e.g. as an embedding layer in a neural net. The machine learning models we used are very basic, and could in the future be replaced by more powerful models such as *Siamese neural nets*[80] and/ or optimized using e.g. alternative loss functions[81].

In conclusion, we showed that protein domain and genome embeddings capture significant aspects of a genome's functions, both on the level of domains as well as genomes, enabling a "taxonomy-free taxonomy". They are well suited for subsequent machine learning tasks and solve the "curse of high dimensionality" of previous approaches based on sparse encodings. As representations of function, they have several useful properties, in that they are composable, well-formatted and insensitive in light of incompleteness of the underlying assembly. Especially metagenomic areas such as taxonomic classification, biomining and phenotype prediction can benefit from `nanotext`.

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Methods

Annotation of Tara genomes

We annotated protein domains for a collection of 957 MAGs[7] using `HMMER` (`hmmscan --cut_ga`, v3.2.1)[70] against the `Pfam` database (v32)[19]. We then removed domains with an E-value above 10^{-18} and with a coverage below 35%. A `Snakemake`[82] workflow implementation can be found in the project repository.

Estimation of nucleotide distance using MinHash

To estimate average nucleotide identity between pairs of genomes we used the `MinHash` algorithm[56,83] as implemented in `sourmash` (https://github.com/dib-lab/sourmash)[84]. To generate `MinHash` signatures from genomes, we chose a sketch size of 500 and a k-mer size of 31.

Training of functional embeddings

We combined two large collections of bacterial genome annotations into one corpus. First we included the complete `AnnoTree` collection[29] based on the *Genome Taxonomy Database* (GTDB) ($n = 23936$, release 83)[12]. Second, from the `EnsemblBacteria` database we randomly sampled five genomes for each species ($n = 8667$, release 41)[85]. The sampling balances the dataset; otherwise medically important bacteria would dominate the resulting corpus (https://osf.io/pjf7m/). Each line in the corpus is the sequence of `PFAM` protein domains on a contig. Strand information is not preserved. We did not perform any additional filtering of the protein domains. We trained embeddings on a corpus of 31730 genomes with a total of about 145 million domains.

We obtained word vectors using the `Word2Vec`[36] algorithm for all words in our corpus' vocabulary of 10879 domains, which is about 60% of the total number of domains in the `Pfam` database (v32)[19]. Note that not all domains in `Pfam` are bacterial, and we further excluded protein domains that did not occur in the corpus at least three times. We trained a document topic model using the `Doc2Vec` algorithm[39] with a window size of 10 and a linearly decreasing learning rate (0.025 to 0.0001) over 10 epochs using the *distributed bag of words* (PV-DBOW) training option as implemented in `Gensim`[86]. The result was a 100-dimensional vector. The similarity of any two genome vectors in the collection can be evaluated using cosine similarity, with a range from -1 (no similarity) to 1 (identical). To infer genome vectors for new genomes, we concatenated the protein domain sequences of all contigs and then used 200

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

iterations for inference. This resulted in stable vector estimates with a pairwise cosine distance < 0.01. For the SOMO evaluation task (see results) we withheld 873 randomly selected genomes (3%) from training, to validate the embedding model.

Training of media embeddings

To quantify how similar any pair of culture media was, we created a media embedding. Such a representation has an advantage over using the name or ID of a medium in learning tasks, because many media are very similar, such as when an organism-specific medium is an extension of a base medium. Using an ID, we would create a high-dimensional, one-hot-encoded vector to represent the medium. This vector would be very sparse, with 1 in the index position of a given medium and 0 everywhere else. The current media collection of the DSMZ lists over 1500 media, so any learning algorithm would have problems with the number of dimensions.

To reduce the number of media, we treat a medium recipe as a sequence of ingredients and used `Word2Vec`[36] to create a latent representation in the form of a 10-dimensional vector, similar in idea to embedding cooking recipes (https://bit.ly/2kesqbC) or diets[63]. The DSMZ media are not easily parsable and contain many non-unique ingredient tags such as "beef extract" and the synonymous "meat extract". Therefore we used preprocessed data from the `KOMODO` database of known media[61]. To download all 3637 recipes, we used a custom crawling script (`scrape_komodo.py`). Note that some current additions to the DSMZ media list do not figure in the `KOMODO` database. From each recipe we extracted a list of ingredients[61]. We excluded water (`SEED-cpd00001###`) and agar (`SEED-cpd13334###`) because these ingredients are highly redundant and would act as noise during training. We embedded the ingredients using `Word2Vec` with a window size of 5 and a learning rate as described above over 100 epochs using negative sampling of 15 words per window. To make sure that pairs of media ingredients could occur in the same window, we augmented the data set by shuffling each ingredient list 100 times[87]. The result is a 10-dimensional vector for each media ingredient. To create culture medium embeddings, we summed across the embedding vectors for all ingredients in a medium.

The similarity of any two DSMZ media could then be compared using cosine similarity. For example, the closest media to medium no. 1 are medium no. 306 (0.99) and no. 617 (0.99), one adding *yeast extract* and the other *NaCl* to medium no. 1; an ID-based representation would treat these media as distict, although they are near identical. Indeed, medium no. 617 and 953 have identical ingredients, which is reflected by a cosine similarity of 1.

Embeddings are useful as input to learning algorithms only if they position similar entities in similar

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

vector space, i.e. if similar entities cluster. We therefore visualized the media vector space using t-SNE (Figure 5). Indeed, similar media cluster and thus enable learning algorithms to discriminate media classes. For downstream machine learning tasks, the vector representation has two major advantages: It reduces the dimentionality of the media representation by 2 orders of magnitude, from one-hot-encoding of more than one thousand media to a 10-dimensional vector. Another advantage is that any predicted medium (see results) can suggest n similar media as starting point, instead of just one. While this might seem inexact, we think it offers much more information about culturing previously uncultured organisms, as a wider range of media can be explored and mixed.

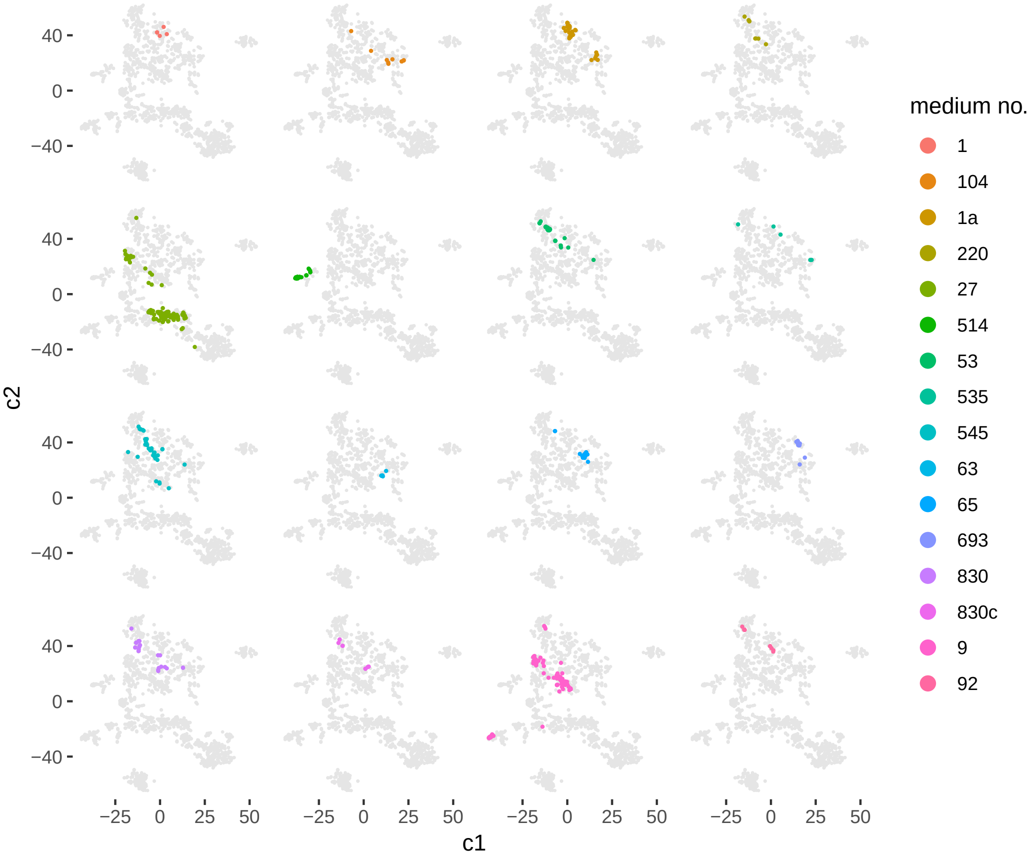

Figure 5: Supplement. Culture medium embedding. We used t-SNE to project the associated 10-dimensional embedding vector into the plane (grey points). We colored all media with more than 0.95 cosine similarity to the top 16 most common DSMZ media in the `BacDive` database. We observe clear clusters of similar media. These clusters can be used by learning algorithms to discriminate media classes. Also note how near-identical media such as no. 830 and no. 830c are embedded in near identical vector space, which acts as a negative control to validate the embedding model.

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Linking `AnnoTree` genome assemblies to `BacDive` culture media

To predict a medium (vector) from a genome (vector) we needed to create a training set that matches the two. The `BacDive` database from the DSMZ holds taxonomic and phenotypic information including culture media for currently over 60 thousand strains[62]. However, these strains do not directly correspond to genomes in the `AnnoTree` collection[12,29]. To link these two, we had to pair records using taxonomy at the rank of genera.

Machine learning

Culture medium prediction

For the medium prediction task, we used a multi-layer fully connected neural network. We selected the training data as follows: For each genus used to link the two databases, we first sampled records from `BacDive` at the genus level. Because this data is highly skewed towards medically relevant genera such as *Mycobacterium*, we randomly selected a maximum of 100 records per genus to balance the training data. As target y, we used the embedding vector of the the most common culture medium in `BacDive` at the genus level. For the same genus we then randomly sampled a genome vector from `nanotext`, which we used as input X. We had to use the most common medium and not sample from these media as we did for the genome input, because many `BacDive` records hold a list of possible culture media with very different recipes and by extension very distant media embeddings. For example, there are two media records for the genus *Rubrimonas*, no. 13 and 514, with a cosine similarity of 0.48 – given that our data set is small, the learning algorithm was not able to learn this complex mapping.

We repeated this process 10 times to augment our dataset. Data augmentation is a common practice when training neural nets. It enables the training of more complex models, which then generalize better. Using data augmentation, we can circumvent the need to collect more data by varying the input slightly. For images this typically means flipping images horizontally or generating new training images by selecting only a subset of pixels. In seminal work on the `ImageNet` challenge for example, the original data was augmented 2048 times[15]. We used a total of 73916 genom-media pairs for training, optimized hyperparameters on a validation set of 3891 (5%) and tested the final model on a holdout set of size 8646 (10%). The neural net architecture consisted of three fully connected layers with (512, 128, 64) nodes. Before applying the non-linear transformation (rectified linear units, ReLU), we normalized the batches of size 128. After each layer we applied Dropout (0.5, 0.3, 0.1). The output layer had 10 nodes to represent a culture medium vector with 10 latent elements, which were activated with a linear transformation. We optimized a cosine similarity loss of the output medium vector with the target

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

medium vector using the `Adam` optimizer with a learning rate of 10^{-2} over the course of 10 epochs. Because we used a cosine similarity loss, we did not rescale (X, y) before training. We implemented the model using the deep learning library `Keras` (https://keras.io).

Water medium prediction

For the water temperature prediction task we used a *Gradient Boosting Trees* (GBT) regressor[88]. For each of the 93 sampling sites in the Tara dataset, we averaged the genome vectors of the 10 most abundant MAGs, where abundance was estimated using the relative number of reads that belonged to any MAG at the given sampling site[7]. Our target variable was the recorded temperature for this site (see supplementary information in *Delmont et al.*). We used grid search to optimize the GBT parameters (learning rate: 0.05, maximum depth: 4, maximum percentage of features used duing iterations: 30%, minimum number of samples per leaf: 3). The final model is an ensemble of 3000 trees. Because the number of samples was small compared to the input dimensions, we used leave-one-out cross-validation (LOOCV) to make predictions. The model was implemented using the machine learning library `Sklearn`[89].

Code availability

All relevant resources to reproduce the major results in this article have been deposited in a dedicated `nanotext` repository (https://github.com/phiweger/nanotext). This includes source code, protein domain and genome embeddings as well as preprocessing workflows. The corpus we trained `nanotext` on is also made available (https://osf.io/pjf7m/).

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Acknowledgments

The authors thank Donovan Parks for providing the `AnnoTree` protein domain annotations.

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

References

1 Stadler, P. F. *et al.* *Theory Biosci.* 128, 165–170 (2009)

2 Doolittle, W. F. *Genome Biol.* 19, 223 (2018)

3 Pesant, S. *et al.* *Sci Data* 2, 150023 (2015)

4 Parks, D. H. *et al.* *Nat Microbiol* 2, 1533–1542 (2017)

5 Tully, B. J. *et al.* *bioRxiv* 162503 (2017)

6 Stewart, R. *et al.* *bioRxiv* 162578 (2017)

7 Delmont, T. O. *et al.* *Nat Microbiol* 3, 804–813 (2018)

8 Stewart, R. D. *et al.* *bioRxiv* 489443 (2018)

9 Qin, J. *et al.* *Nature* 464, 59–65 (2010)

10 Steinegger, M. *et al.* *bioRxiv* 386110 (2018)

11 Tatusov, R. L. *et al.* *Nucleic Acids Res.* 28, 33–36 (2000)

12 Parks, D. H. *et al.* *Nat. Biotechnol.* 36, 996–1004 (2018)

13 Valenzuela, L. *et al.* *Biotechnol. Adv.* 24, 197–211 (2006)

14 Pascal, V. *et al.* *Gut* 66, 813–822 (2017)

15 Krizhevsky, A. *et al.* in *Advances in neural information processing systems 25* (eds. Pereira, F., Burges, C. J. C., Bottou, L. & Weinberger, K. Q.) 1097–1105 (Curran Associates, Inc., 2012)

16 Langille, M. G. I. *mSystems* 3, (2018)

17 Doolittle, W. F. *et al.* *Biol. Philos.* 32, 5–24 (2017)

18 Heintz-Buschart, A. *et al.* *Trends Microbiol.* 26, 563–574 (2018)

19 Finn, R. D. *et al.* *Nucleic Acids Res.* 44, D279–85 (2016)

20 Alborzi, S. Z. *et al.* *BMC Bioinformatics* 18, 107 (2017)

21 Vogel, C. *et al.* *Curr. Opin. Struct. Biol.* 14, 208–216 (2004)

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

22 Tordai, H. *et al. FEBS J.* 272, 5064–5078 (2005)

23 Marsh, J. A. *et al. Genome Biol.* 11, 126 (2010)

24 Illergård, K. *et al. Proteins* 77, 499–508 (2009)

25 Holmes, E. C. (OUP Oxford, 2009)

26 Zhu, C. *et al. PLoS Comput. Biol.* 11, e1004472 (2015)

27 Zhu, C. *et al. Nucleic Acids Res.* 46, D535–D541 (2018)

28 Weimann, A. *et al. mSystems* 1, (2016)

29 Mendler, K. *et al. bioRxiv* 463455 (2018)

30 Blin, K. *et al. Nucleic Acids Res.* 45, W36–W41 (2017)

31 Terrapon, N. *et al. Nucleic Acids Res.* 46, D677–D683 (2018)

32 Gordon, S. P. *et al. PLoS One* 10, e0132628 (2015)

33 Burkhardt, D. H. *et al. Elife* 6, (2017)

34 Hinton, G. E. *et al.* in (eds. Rumelhart, D. E., McClelland, J. L. & PDP Research Group, C.) 77–109 (MIT Press, 1986)

35 Rudolph, M. R. *et al.* (2016)

36 Mikolov, T. *et al.* in *Advances in neural information processing systems 26* (eds. Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z. & Weinberger, K. Q.) 3111–3119 (Curran Associates, Inc., 2013)

37 Pennington, J. *et al.* in *In EMNLP* (2014)

38 Bojanowski, P. *et al.* (2016)

39 Le, Q. V. *et al.* (2014)

40 Blei, D. M. *Commun. ACM* 55, 77–84 (2012)

41 Asgari, E. *et al. PLoS One* 10, e0141287 (2015)

42 Du, J. *et al. bioRxiv* 286096 (2018)

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

43 Yang, K. K. *et al. Bioinformatics* 34, 2642–2648 (2018)

44 Hamid, M.-N. *et al. Bioinformatics* (2018)

45 Jaeger, S. *et al. J. Chem. Inf. Model.* 58, 27–35 (2018)

46 Kimothi, D. *et al.* (2016)

47 Ng, P. (2017)

48 Asgari, E. *et al. Bioinformatics* (2018)

49 Asgari, E. *et al. Bioinformatics* 34, i32–i42 (2018)

50 Schnabel, T. *et al. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* 298–307 (2015)

51 Conneau, A. *et al.* (2018)

52 Maaten, L. van der. *J. Mach. Learn. Res.* 15, 3221–3245 (2014)

53 Zhang, X. *et al. Proc. Natl. Acad. Sci. U. S. A.* 113, E4161–9 (2016)

54 O'Leary, N. A. *et al. Nucleic Acids Res.* 44, D733–45 (2016)

55 Campbell, B. J. *et al. Proc. Natl. Acad. Sci. U. S. A.* 108, 12776–12781 (2011)

56 Ondov, B. D. *et al. Genome Biol.* 17, 132 (2016)

57 Wannicke, N. *et al. FEMS Microbiol. Ecol.* 91, (2015)

58 Sandle, T. *PDA J. Pharm. Sci. Technol.* 58, 231–237 (2004)

59 Browne, H. P. *et al. Nature* 533, 543–546 (2016)

60 Ark, K. C. H. van der *et al. Microb. Biotechnol.* 11, 476–485 (2018)

61 Oberhardt, M. A. *et al. Nat. Commun.* 6, 8493 (2015)

62 Reimer, L. C. *et al. Nucleic Acids Res.* (2018)

63 Tansey, W. *et al.* (2016)

64 Moore, L. R. *et al. Limnol. Oceanogr. Methods* 5, 353–362 (2007)

65 Bandyopadhyay, A. *et al. MBio* 2, (2011)

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

66 Welsh, E. A. *et al. Proc. Natl. Acad. Sci. U. S. A.* 105, 15094–15099 (2008)

67 Waters, L. S. *et al. Cell* 136, 615–628 (2009)

68 Sunagawa, S. *et al. Science* 348, 1261359 (2015)

69 Roux, S. *et al. Nature* 537, 689–693 (2016)

70 Eddy, S. R. *PLoS Comput. Biol.* 7, e1002195 (2011)

71 Hauser, M. *et al. Bioinformatics* 32, 1323–1330 (2016)

72 Mahlich, Y. *et al. Bioinformatics* 34, i304–i312 (2018)

73 Steinegger, M. *et al. Nat. Commun.* 9, 2542 (2018)

74 Johnson, J. *et al.* (2017)

75 Joulin, A. *et al.* (2016)

76 Wu, L. *et al.* (2017)

77 Salvador, A. *et al.* in *2017 IEEE conference on computer vision and pattern recognition (CVPR)* 3068–3076 (2017)

78 Wright, E. S. *et al. BMC Genomics* 19, 724 (2018)

79 McInerney, J. O. *et al. Nat Microbiol* 2, 17040 (2017)

80 Koch, G. *et al.* in *ICML deep learning workshop* 2, (2015)

81 Wojke, N. *et al.* in *2018 IEEE winter conference on applications of computer vision (WACV)* 748–756 (2018)

82 Köster, J. *et al. Bioinformatics* 28, 2520–2522 (2012)

83 Broder, A. Z. in *Compression and complexity of sequences 1997. Proceedings* 21–29 (IEEE, 1997)

84 Brown, C. T. *et al. The Journal of Open Source Software* (2016)

85 Zerbino, D. R. *et al. Nucleic Acids Res.* 46, D754–D761 (2018)

86 Řehůřek, R. *et al.* (University of Malta, 2010)

87 Barkan, O. *et al.* (2016)

bioRxiv preprint doi: https://doi.org/10.1101/524280; this version posted January 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

88 Friedman, J. H. *Ann. Stat.* 29, 1189–1232 (2001)

89 Pedregosa, F. *et al. J. Mach. Learn. Res.* 12, 2825–2830 (2011)