Sparse non-negative matrix factorization for retrieving genomes across metagenomes

Vincent Prost, Stéphane Gazut, Thomas Brüls

Sparse non-negative matrix factorization for retrieving genomes across metagenomes

Vincent Prost[1,2], Stéphane Gazut[1], and Thomas Brüls[2]

[1] CEA, LIST, Laboratoire Sciences des Données et de la Décision, 91191 Gif-sur-Yvette, France
{vincent.prost, stephane.gazut}@cea.fr
[2] CEA, DRF, Institut Jacob, Genoscope, 91057 Evry, France
{vincent.prost, bruls}@genoscope.cns.fr

Abstract. The recent development of next generation sequencing allows to sequence DNA at high rate and low cost, therefore facilitating metagenomics, the study of complex microbial communities sequenced in their natural environment. A metagenomic dataset consists of billions of unordered small fragments of genomes (reads), coming from hundreds or thousands of different genomes. It is very challenging to recover individual genomes from metagenomes, because of the complexity of the task and the large amount of data. The clustering of reads into operational taxonomic units (OTUs), known as binning, is a key step but most of computational tools are performing read assembly as pre-processing which is computationally intensive, requiring terabytes of RAM and has a lot of drawbacks : it produces errors and loses a lot of information.
We use abundance signals, meaning the counts of long k-mers (subsequences of size k) appearing in samples. We show how we can, using online learning methods for sparse non-negative matrix factorization, recover relative abundances of genomes across multiple metagenomes and perform assembly free binning. The combinatorial explosion of the k-mers is solved with local sensitive hashing. The underlying k-mer abundances are estimated with sparse coding and dictionary learning techniques.

Keywords: Non negative matrix factorization · Sparse coding · Metagenomics · Clustering

1 Introduction

High-throughput sequencing can extract genomic information about microbial communities in their natural environment, enabling a relatively new field of research, metagenomics. Metagenomic studies already have expanded knowledge in various domains, like ecology or human medicine with the study of human gut microbiota [13]. The recent development of technology has improved the sequencing rate and lowered the cost. Therefore, more and more data is produced. Metagenomic datasets are particularly big and complex. It consists of billions of unordered small genome fragments, which are strings of letters (A, C, G or T). Those are relatively small, from 100 up to 400bp (based pair), compared

to the size of one genome, which is about 10^6bp for a bacteria. Plus, they are done in a random way, without knowing the position in the genome nor which genome it is. That's why recovering individual genomes, by assembling the DNA fragments, is a very complex and challenging task.

Many computational tools aim to solve an intermediate task called binning. Binning means clustering the reads coming from the same species or strain together. We can distinguish two different strategies for binning. A first strategy uses de novo assembly [14] as a pre-processing step and perform binning on contigs, larger fragments of DNA (10^3 - 10^4 bp). It has advantages, it reduces the number of objects and larger sequence improve the robustness of signals. But doing de-novo assembly on metagenomic dataset is computationally intensive and requires a lot of memory. Plus, it is a source of errors and loss of information.

A second strategy is to directly tackle raw short reads. It aims to solve a more complex problem, due to the larger number of objects it deals with, but avoids the drawbacks of de novo assembly. We identify our contribution against the context of this second strategy.

2 Related work

2.1 Clustering long *k*-mers

In order to perform binning, many methods ([3], [15]) use the number of occurrences of long k-mers (substring of size k). With sufficiently long k-mers ($k > 20$), we can assume that they will be species specific. Solving the binning problem, is therefore equivalent to clustering k-mers. The Lander-Waterman model [8] states that random sequencing will lead to Poisson distributed nucleotide position. Abundancebin [15] uses the assumption that occurrences of long k-mer are Poisson distributed with parameter proportional to the abundance of the species it comes from. If we note λ_i this parameter, the count $n(w_j)$ of a k-mer w_j is Poisson distributed :

$$P(n(w_j) = c) = Poisson(\lambda_i; c) \tag{1}$$

Where $Poisson(\lambda_i; c)$ is the probability of a Poisson random variable taking the value c. [15] estimates the parameters λ_j, by maximizing the likelihood using EM algorithm. Reads are then partitioned into bins according to the k-mers they contain and the estimated parameters. [3] uses long k-mers ($k = 36$) for grouping reads together after a filtering step.

2.2 Handling the size of data : local sensitive hashing

A problem arises concerning the memory needed for storing the counts of all observed k-mer. The number of possible k-mer is very large (eg. $k = 30$ give $4^{30} \approx 10^{18}$ possible k-mer). The use of inverted indexes as in [15] will not be tractable in memory for large datasets, especially for mutli-samples datasets. [10] propose to use a Local Sensitive Hashing (LSH) technique, initially used to

facilitate nearest-neighbour query in high dimension context [7], for the purpose of reducing the size of stored data.

Each k-mer w_i is represented in the complex vector space $\mathbb{C}^k$ by assigning to every letter a complex number : $A := 1, C := -i, G := -1, T := i$. Those numbers can be weighted with a quality number which states about the confidence of the measure. d random hyperpanes in $\mathbb{C}^k$ are drawn, let's say with normal vector $v_j \in \mathbb{C}^l$. Each hyperplane separates the space into two half spaces , therefore defining 2^d subspaces called buckets. We define the following hashing function :

$$h_j(w_i) = sign(w_i^T v_j) \in \{-1, 1\} \tag{2}$$

In conclusion, each k-mer w_i, initially living in a space of cardinal 4^k, can be associated to a binary code $(h_1(w_i), h_1(w_i), .., h_d(w_i)) \in \{0, 1\}^d$ of size d and be represented in a space of cardinal 2^d. This way we can control the size of the "dictionary" with the number of hyperplans we choose.

2.3 LSA : analogy with LSI (Latent Semantic Indexing)

Once the size of the "dictionary" is fixed to a reasonable size, we can count and store k-mers occurring in each sample in an abundance matrix, $X \in \mathbb{R}^{n \times 2^d}$ where n is the number of samples and 2^d is the number of buckets. [10] carries out an analogy with LSI (Latent Semantic Indexing [6]) a classic method for document classification. Projecting each sample into the singular vector space with SVD (Singular vector decomposition) : $X = U\Sigma V^T$ where U and V are orthogonal and Σ is diagonal and then performs fixed radius k-means on the lines of V.

3 Material and methods

3.1 Sparse non negative matrix factorization

In our model, data is a sparse composition of positive components, an addition of underlying parameters that we want to recover. Non negative matrix factorization (NMF) stands out as a natural way of proceeding. It has been proposed originally by Lee and Seung [9]. In NMF, we want to approximate the data $X \in \mathbb{R}^{n \times 2^d}$ by the product of two non-negative matrices $U \in \mathbb{R}^{n \times K}$ and $V \in \mathbb{R}^{2^d \times K}$, usually by finding a solution to the following constrained minimization problem :

$$U, V = \underset{U,V \geq 0}{\operatorname{argmin}} \mathcal{D}(X||UV^T) + J(U, V) \tag{3}$$

Where $\mathcal{D}$ is an error function and J a penalty term ensuring sparsity or regularity of U and V. In our model, the values of matrix X are a sparse composition of k-mer counts. We note S_{isj}, the count of a k-mer specific to species s appearing in bucket j and sample i. Then :

$$X_{ij} = \sum_{s=1}^{K} S_{isj} \tag{4}$$

S_{isj} follows a Poisson distribution of parameter $U_{is}V_{js}$, where U_{is} is proportional to the abundance of species s in sample i and V_{js} is a sparse coefficient. Due to the additivity of the Poisson distribution, X_{ij} is also Poisson distributed :

$$P(X_{ij} = c) = Poisson\left(\sum_{s=1}^{K} U_{is}V_{js}; c\right) \tag{5}$$

As stated in the original paper [9] and developped in [2], maximizing the likelihood of this model is equivalent to solving (3) when $\mathcal{D}$ is the Kullback-Leibler divergence :

$$KL(X||UV^T) = \sum_i \sum_j \log \frac{X_{ij}}{[UV^T]_{ij}} - X_{ij} + [UV^T]_{ij} \tag{6}$$

Lee and al. proposed an iterative algorithm for solving (3) when $J := 0$. This algorithm is tested in the experiments as "L&S-KL".

We can expect V to be very sparse. The sparsity of V depends on the expected number of k-mer sharing a same bucket. Therefore depends on the number of buckets 2^d compared with the number a different observed k-mers. That's why we need to set constraints in the optimization problem to ensure sparsity of V, by penalizing either l_0 or l_1 norm of the lines of V : $J(U, V) = \beta \sum_{i=0}^{2^d} \|v_i\|_\gamma, \gamma \in \{0, 1\}$, where v_i is the ith line of V. Sparsity constraints are also inevitable in our case, where we'll often have $n < K$, in order to seek the solution with the fewest number of non zeros among the infinite number of solutions [1].

3.2 Online dictionary learning

Iterative methods, like in [9], require to keep in memory the whole dataset, which is not suitable knowing the potentially large dimensions of the matrices. We can, as Mairal and al. [11], use an online dictionary learning method that aims to solve (3) when $\mathcal{D}(X||UV^T) = \|X - UV^T\|_2^2$, where $\|.\|_2$ is the Froebinus norm. It proceeds in an online fashion by alternating for each new coming input x_t a sparse coding step (updating v_t) :

$$v_t = \underset{v \geq 0}{\operatorname{argmin}} \|x_t - Uv\|_2^2 + \lambda \|v\|_1 \tag{7}$$

And a dictionary update step (updating matrix U) :

$$U^{t+1} = \underset{U \in \mathcal{C}}{\operatorname{argmin}} \sum_{i=1}^{t} \|x_i - Uv_i\|_2^2 \tag{8}$$

where x_i denotes the ith column of X and v_i the ith line of V. $\mathcal{C}$ is a convex set. We'll test Mairal and al. methods with different sparse coding steps. We note lasso-DL when sparse coding step is a lasso regression as in eq. (7). and omp-DL when it's Orthogonal Matching Pursuit, aiming to solve :

$$v_t = \underset{v \geq 0}{\operatorname{argmin}} \|v\|_0 \text{ subject to } \|x_t - Uv\|_2 < \epsilon \tag{9}$$

[12] proposes an algorithm inspired by [11] but aiming to solve problem (3) when $D(.||.) := KL(.||.)$ and with sparsity constraints. It will be noted KL-DL in the experiements.

3.3 Data

We consider three types of datasets.

1. Synthetic datasets that simulate k-mer counts in a sparse Poisson factor model, cf. eq. (4).
2. Semi-synthetic datasets that simulate the sequencing of biological samples by randomly sampling sequences of referenced genomes.
3. Real metagenomic datasets.

With the first type of datasets, we will evaluate the performance of methods in the ideal case of the Lander-Waterman model. We have a control on the variables of the model and we can evaluate the ability of methods to retrieve the underlying abundances.

For the second type, real genomes are used to simulate a shotgun sequencing. It removes some limitations of the Lander-Waterman model, typically sequencing errors. In this case, we'll evaluate the final binning results by quantifying the ability of algorithms to cluster reads into bins of same species with precision and recall metrics (P/R) [4].

The performance on the third type of dataset is more complex to evaluate because we usually do not know the "ground truth". We have to use other bioinformatic tools in order to evaluate the performance of the binning.

3.4 Experiments

Synthetic data As in [1], we first try the algorithms on synthetic signals, given a random underlying abundance parameters $A \in \mathbb{R}^{n \times K}$. At each iteration, T samples $(x_1, x_2, .., x_T)$ are independently drawn following (10).

$$\begin{aligned} x_i^{(j)} &= \sum_{k=1}^{K} \pi_k s_k^{(j)} \\ \pi_k &\sim Binomial(p, 3) \\ s_k^{(j)} &\sim Poisson(A_{j,k}) \end{aligned} \tag{10}$$

Where $x^{(j)}$ denote the jth coordinate of vector x. The computed left matrix U was compared against the known abundance parameter A. The error is the sum of quadratic error between columns of A and the closest columns of U : $error = \sum_{k=0}^{K} \min_i \frac{\|A_{.,k} - U_{.,i}\|}{\|A_{.,k}\| \|U_{.,i}\|}$. The tests were done with parameters $p = 5 \times 10^{-2}, K = 140, n = 20, T = 1000$. Figure 3.4 shows the comparison of different online learning methods. The curve "k-means" represents the online version of k-means (sequencial kmeans [5]). "kl-means" is the same algorithm but with

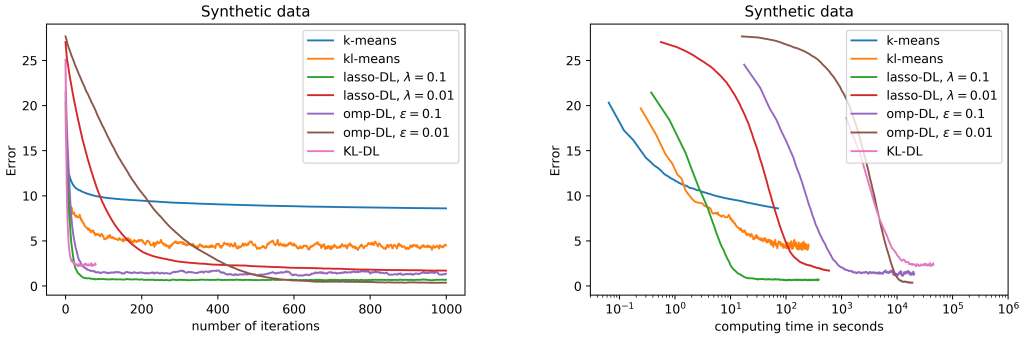

Fig. 1. Left : error as a function of iteration number. Right error : as a function of computation time on a logarithmic scale.

euclidean distance replaced by KL-divergence. All methods with relaxed sparse constraints manage to better recover underlying abundances than k-means. Using OMP sparse coding step tends to slightly improve the estimation but with higher computing cost. Lasso-DL achieve the best convergence speed both in iteration and computation time. KL-DL is much slower than other methods but surprisingly fails to improve the final estimation.

Semi-synthetic data We tested methods on synthetic metagenomic datasets with short sequence length (400 bp), simulating a cohort of $n = 50$ samples. Each sample contains a random subset of a given number of bacterial genomes (Genome nb). The number of sequences varies from 200,000 (for Genome nb = 20) to 7,5 millions (for Genome nb = 700) per sample.

For the partitioning of sequences, first k-mers are clustered. Values of V are computed, then, as in [16], we assign k-mer i to cluster k if : $k = \text{argmax}_j V_{i,j}$. In the end, sequences are partitioned to bins regarding their k-mers content like in [10]. Data has been prepossessed with hashing (cf. seq. 2.2) with $d = 27$ (Table 1) and $d = 30$ (Table 2). The number of clusters K is the same for all methods and is set to 1.5 × Genome nb. Parameters λ and ϵ have been set to achieve a good compromise between the sparsity of V and the reconstruction error.

We can see that online dictionary learning methods outperform others in overdetermined cases ($K > n$). Their efficiency decreases as the sparsity decreases (cf. Table 1) but do scale well to large dimensions (cf. Table 2). L&S-KL and KL-DL could not have been computed on datasets with $d = 30$ due to too big computing times.

Real data . The method has been applied to a real metagenomic dataset (10 billions of reads, 1135 samples) extracted from human gut microbiota. Results have been submitted to a journal publication : `https://doi.org/10.1101/599332`.

Table 1. Comparison of binning results on synthetic metagenomic datasets ($d = 27$).

Genome nb	k-means		L&S-KL		omp-DL		lasso-DL		KL-DL	
	P/R		P/R		P/R		P/R		P/R	
20	74.1	70.8	83.6	90.0	72.6	71.1	76.9	80.1	78.2	90.7
100	57.5	56.2	65.8	66.6	80.5	77.2	80.1	77.0	79.2	80.3
200	61.1	53.0	51.7	50.7	68.1	71.1	70.0	68.3	69.2	70.0
700	46.2	43.9	44.2	45.7	49.3	47.7	49.1	47.5	-	-

Table 2. Comparison of binning results on synthetic metagenomic datasets ($d = 30$).

Genome nb	k-means		omp-DL		lasso-DL	
	P/R		P/R		P/R	
700	52.4	62.9	75.2	80.7	73.6	81.6

It shows success for detecting low abundance species that clasic methods can't identify.

4 Conclusion

We have shown that sparse non negative matrix factorization can be used for analysing metagenomic datasets. We explored and compared different methods and validated them through experiments on synthetic and semi-synthetic data. We have demonstrated through experiments that online dictionary learning methods coupled with sparse coding are able to recover underlying parameters in a sparse Poisson factor model and in an overdetermined setting which we think represents the specificity of our data. Finally, we showed that NMF can be applied for clustering k-mers and perform binning of short reads without prior assembly and in fixed memory with satisfying results.

Among the many directions further work can take, we'll note two points. Solving limitations due to the existence of pangenomes with soft-clustering approaches and acceleration of Kullback-Leibler divergences based dictionary learning algorithms.

5 Acknowledgments

This work has been mainly funded by the office of the High Commissioner of CEA. The authors would like to thank Olexiy Kyrgyzov for having written and computed codes for the real dataset.

References

1. M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, Nov 2006.

2. Ali Taylan Cemgil. Bayesian inference for nonnegative matrix factorisation models. *Intell. Neuroscience*, 2009:4:1–4:17, January 2009.
3. Francis Y.L. Chin, Henry C.M. Leung, S.M. Yiu, and Yi Wang. MetaCluster 5.0: a two-round binning approach for metagenomic data for low-abundance species in a noisy sample. *Bioinformatics*, 28(18):i356–i362, 09 2012.
4. Wikipedia contributors. Precision and recall, 2019. [Online; accessed 29-May-2019].
5. Richard O. Duda. Pattern recognition for hci. `www.cs.princeton.edu/courses/archive/fall08/cos436/Duda/PR_home.htm`, June 1997. Accessed: 2019-05-27.
6. Susan T. Dumais. Latent semantic analysis. *Annual Review of Information Science and Technology*, 38(1):188–230, 2004.
7. Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, pages 518–529, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.
8. Eric S. Lander and Michael S. Waterman. Genomic mapping by fingerprinting random clones: A mathematical analysis. *Genomics*, 2(3):231 – 239, 1988.
9. Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, pages 535–541, Cambridge, MA, USA, 2000. MIT Press.
10. Brian Lowman Cleary, Ilana Lauren Brito, Katherine Huang, Dirk Gevers, Terrance Shea, Sarah Young, and Eric J Alm. Detection of low-abundance bacterial strains in metagenomic datasets by eigengenome partitioning. *Nature biotechnology*, 33, 09 2015.
11. Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, March 2010.
12. Duy Nguyen and Tu Ho. Fast parallel randomized algorithm for nonnegative matrix factorization with kl divergence for large sparse datasets. *International Journal of Machine Learning and Computing*, 6, 04 2016.
13. Junjie Qin, Yingrui Li, Zhiming Cai, Shenghui Li, Jianfeng Zhu, Fan Zhang, Suisha Liang, Wenwei Zhang, Yuanlin Guan, Dongqian Shen, Yangqing Peng, Dongya Zhang, Zhuye Jie, Wenxian Wu, Youwen Qin, Wenbin Xue, Junhua Li, Lingchuan Han, Donghui Lu, Peixian Wu, Yali Dai, Xiaojuan Sun, Zesong Li, Aifa Tang, Shilong Zhong, Xiaoping Li, Weineng Chen, Ran Xu, Mingbang Wang, Qiang Feng, Meihua Gong, Jing Yu, Yanyan Zhang, Ming Zhang, Torben Hansen, Gaston Sanchez, Jeroen Raes, Gwen Falony, Shujiro Okuda, Mathieu Almeida, Emmanuelle Le Chatelier, Pierre Renault, Nicolas Pons, Jean-Michel Batto, Zhaoxi Zhang, Hua Chen, Ruifu Yang, Weimou Zheng, Songgang Li, Huanming Yang, Jian Wang, S. Dusko Ehrlich, Rasmus Nielsen, Oluf Pedersen, Karsten Kristiansen, and Jun Wang. A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature*, 490(7418):55 – 60, 2012.
14. Daniel R Zerbino and Ewan Birney. Velvet: Algorithms for de novo short read assembly using de bruijn graphs. *Genome research*, 18:821–9, 06 2008.
15. Ye Y. Wu YW. A novel abundance-based algorithm for binning metagenomic sequences using l-tuples. *J Comput Biol*, 18(3):523–34, 2011.
16. Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 267–273, New York, NY, USA, 2003. ACM.