# MNHN-Tree-Tools: A toolbox for tree inference using multi-scale clustering of a set of sequences

Thomas Haschka, Loic Ponger, Christophe Escude, Julien Mozziconacci

*Bioinformatics*
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Application Note


Genome analysis

# `MNHN-Tree-Tools`: A toolbox for tree inference using multi-scale clustering of a set of sequences

**Thomas Haschka [1,*], Loic Ponger [1], Christophe Escudé [1] and Julien Mozziconacci [1,2,*]**

[1] Muséum National d'Histoire Naturelle, Structure et Instabilité des Génomes, UMR7196, Paris 75231, France
[2] Institut Universitaire de France

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

## Abstract

**Summary:** Genomic sequences are widely used to infer the evolutionary history of a given group of individuals. Many methods have been developed for sequence clustering and tree building. In the early days of genome sequencing, these were often limited to hundreds of sequences, but due to the surge of high throughput sequencing, it is now common to have millions of sampled sequences at hand. We introduce MNHN-Tree-Tools, a high performance set of algorithms that builds multi-scale, nested clusters of sequences found in a FASTA file. MNHN-Tree-Tools does not rely on sequence alignment and can thus be used on large datasets to infer a sequence tree. Herein we outline two applications: A human alpha-satellite repeats classification and a tree of life derivation from 16S/18S rDNA sequences.
**Availability:** Freely available with a Zlib License from our website:
`http://treetools.haschka.net/`
**Supplementary information:** An in depth discussion about the algorithm with numerical simulations:
`http://treetools.haschka.net/mnhn-as.pdf`
**Manual:** A detailed users guide and tutorial:
`https://gitlab.in2p3.fr/mnhn-tools/mnhn-tree-tools-manual/-/raw/master/manual.pdf`
**Contact:** julien.mozziconacci@mnhn.fr and thomas.haschka@mnhn.fr

## 1 Introduction

Sequences are slowly diverging in the course of evolution. The similarity between genomic loci, as for instance specific gene sequences, can in principle be used to infer the evolutionary relationship between individuals. Clustering methods are often used to group sequences together into species, genus, families, orders, class, phylum's, kingdoms and domains. Different experimental methods, such as DNA barcoding (Hajibabaei *et al.* (2007); DeSalle and Goldstein (2019)), are used to determine the set of sequences to be clustered. Sequences are then often curated and gathered into large databases (Munoz *et al.* (2011); McDonald *et al.* (2012)). With the recent advances in DNA high-throughput sequencing (Goodwin *et al.* (2016)), specimen collections, and storage capacities, it is now common to deal with datasets with millions of entries. Several computational approaches have been developed to keep up with the size of these datasets (Rognes *et al.* (2016); Mahé *et al.* (2015)) but they all provide clusters rather than trees. We propose here a new and fast method that performs a multiple alignment free, multi-scale clustering of a set of sequences found in a FASTA (Lipman and Pearson (1985)) file, leveraging the density-based algorithm for discovering clusters in large spatial databases with noise (DBSCAN) (Ester *et al.* (1996)). Nested clusters are then identified to build a tree.

Briefly sketched, the DBSCAN algorithm is a two parameter algorithm requiring a radius $\epsilon$ and a minimum number of objects *minpts*, in our case sequences, to be found withing this radius. As such, this algorithm finds density $\rho = \frac{n_{\mathrm{minpts}}}{V(\epsilon)}$ connected regions, i.e. clusters with a density $> \rho(\mathrm{minpts}, \epsilon)$ (Ester *et al.* (1996)). The use of DBSCAN has been proposed by others as a guide to phylogenetic inference (Ruzgar and Erciyes (2012); Mahapatro *et al.* (2012)). The novelty introduced by

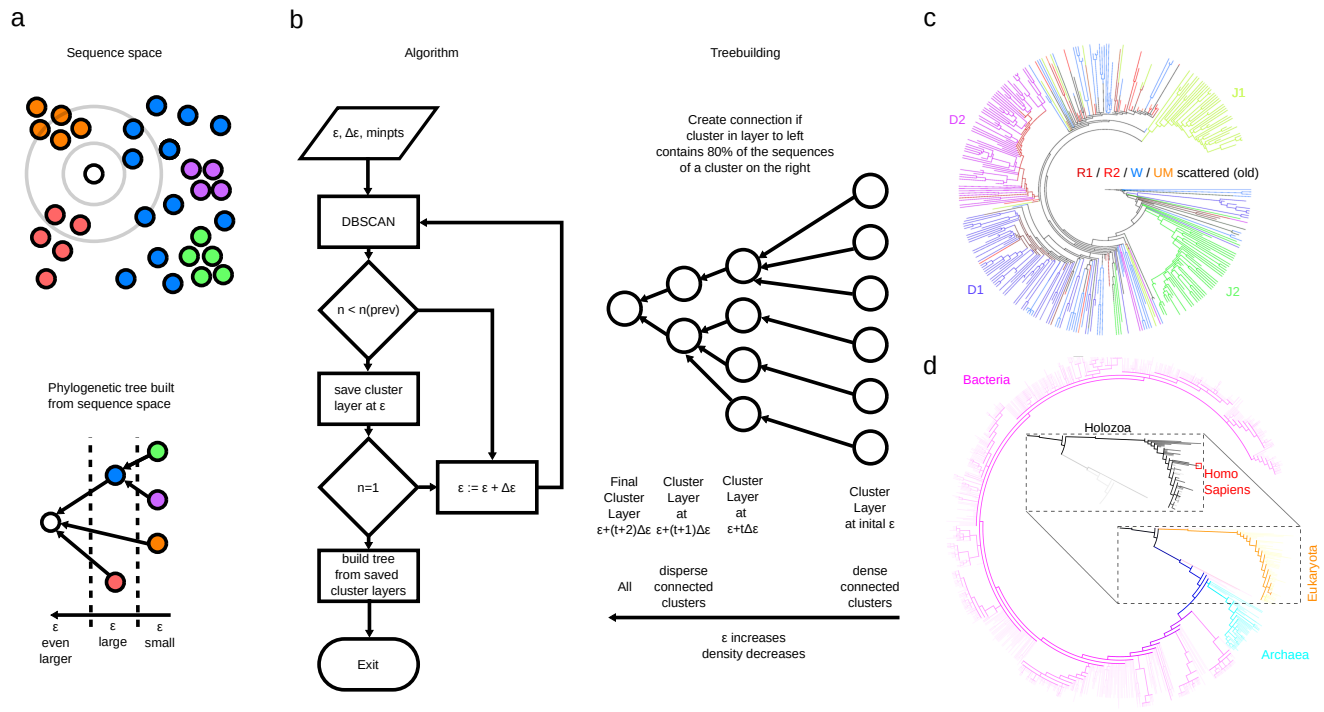


**Fig. 1. Overview of `MNHN-Tree-Tools`**(a) Closely related sequences form dense clusters (in purple and green). These are embedded into a less dense cluster (in blue). The DBSCAN algorithm applied at various radius values ($\epsilon$), can identify these nested clusters. A tree of the identified clusters can then be build (b) Detailed computational workflow (c) Tree build with human Alpha-satellites sequences. Colors correspond to the family annotations in the original dataset (Uralsky et al. (2019)). (d) The tree of life built from 16S/18S RNA sequences. Bacteria, Archaea and Eukaryota are highlighted, with the color intensity corresponding to a logarithmic gradient of the number of sequences in the tree branches. A zoomed representation of Holozoa, clearly outlined as a subclass of Eukaryota, shows the Homo Sapiens branch.

our multi-scale approach is that we perform the DBSCAN algorithm at various densities, and use these layered results to infer a "phylogenetic" tree. Clustering for different $\epsilon$ values allows us to find dense sequence clusters embedded into diffuse clusters (Fig. 1a). We can then build a tree of density connected clusters by successive DBSCAN runs with increased $\epsilon$ parameters and cluster comparison as outlined in Fig. 1b and in the supplementary document. The DBSCAN algorithm was chosen over newer density based methods as, contrary to OPTICS (Ankerst *et al.* (1999)), DBSCAN allows us to control the density of the clusters found and thus allows us to precisely build trees from layers of specific densities. Further DBSCAN features a reduced algorithmic complexity and has as such a runtime advantage over algorithms such as SUBCLU (Kailing *et al.* (2004)). MNHN-Tree-Tools contains the utilities to cluster sequences using two different distance measures:

- The $L_2$-norm operating on a principal component analysis (PCA) based subspace projection of the k-mer sequences representations (Chatterji *et al.* (2008))
- The Smith-Waterman distance (Smith and Waterman (1981)), which features parametric penalties for both substitutions and insertions.

In comparing the k-mer/PCA based approach to the use of the Smith-Waterman distance we show that a k-mer/PCA based distance can yield better clusters and trees due to the inherent feature selection of the PCA while the Smith-Waterman distance can provide more accessible results. In depth details of this comparison are discussed in the supplement to this article. The Smith-Waterman distance computation was implemented in OpenCL (Stone *et al.* (2010)) allowing for execution on Graphics Processing Units (GPUs). We also used the Message Passing Interface (MPI) library (Forum (1994)) to distribute the workloads across different high performance computing cluster nodes.

## 2 Description of `MNHN-Tree-Tools`

MNHN-Tree-Tools is modular suite of command line tools written in the C language. In this section we outline the core utilities, which lead to a multilayered clustering with clusters organised into a tree. We further refer the reader to our manual and supplemental document for a complete documentation.

**Input data:** `MNHN-Tree-Tools` uses as input a FASTA file format that gathers sequences which do not need to be aligned. Typical lengths can vary from 100 to 10000 bp, with length variations up to 10% within samples, but are ultimately only limited by k-mer length or PCA information retention.

**fasta2kmer** A utility to transform FASTA files into a k-mer representation, with a choosable variable subsequence length k.

**kmer2pca** Computes projections of a k-mer representation onto its first principal components. The number of principal components to be used is here defined using an input parameter. The right number has to be chosen in a tradeoff between information capture and curse of dimensionality.

**adaptive_clustering_PCA** Performs clustering at different densities, with the following variable input parameters: initial $\epsilon$, $\Delta\epsilon$ and *minpts*. c.f. (fig 1b)

**split_sets_to_newick** Generates a Newick tree from the clusters obtained.

## 3 Performance and accuracy

We evaluated the accuracy of the algorithm presented herein in three different ways: At first we measured the difference of obtained results to trees that were annotated by experts and as such provided us with valuable ground truth as highlighted in the case studies section below. Further we

compared the algorithm to partitions found by the SWARM2 tool (Mahé *et al.* (2015)). Complementing these experiences we used MNHN-Tree-Tools to evaluate simulated datasets. The comparison to ground truth trees shows that we are capable to find known partitions and highlight accuracy values in the tables provided by the supplemental document. The knowledge about this accuracy was further refined by the application of MNHN-Tree-Tools on simulated datasets that contain sequence clusters of monitored sequence density and inter cluster genetic distance. A comparison to SWARM2 (Mahé *et al.* (2015)) clearly shows that our tree based approach yields, as we search for clusters at different densities, a sweet spot where the found partitions are in close correspondence to those annotated by experts. Classical partitioning tools such as SWARM2 (Mahé *et al.* (2015)) on the other hand yield a single set of partitions that does not correspond, for the application presented herein, to expected results. We refer the reader to our supplement for further details on the accuracy and performance of MNHN-Tree-Tools where the outlined experiences are discussed in detail.

## 4 Case studies

**Human alpha-satellites classification:** Sequences were retrieved from (Uralsky *et al.* (2019)). Our algorithm reconstructed a tree ((Fig. 1c) from these 426 106 sequences which was coloured according to their family annotation.

**The tree of life - The SILVA dataset:** Ribosomal RNA sequences from diverse species (2 225 272 ) were downloaded from (Munoz *et al.* (2011)). Our algorithm was used to reconstruct a tree of life based on these sequences ((Fig. 1d).

For these two applications, the run time for one clustering step ranges from 5 minutes (426 106 seq.) to 173 minutes (2 225 272 seq.) on a single Intel(R)i7-4771 3.50GHz core. The clustering for different epsilon values can easily be run in parallel on several cores.

## Acknowledgements

We thank the “Maison de la Simulation de Champagne Ardenne" for allowing us to develop our algorithm on the ROMEO supercomputer. The project was funded by the Muséum National d’Histoire Naturelle (MNHN) and the Institute Universtiare de France (IUF).

## References

Ankerst, M., Breunig, M. M., peter Kriegel, H., and Sander, J. (1999). Optics: Ordering points to identify the clustering structure. pages 49–60. ACM Press.

Chatterji, S., Yamazaki, I., Bai, Z., and Eisen, J. A. (2008). Compostbin: A dna composition-based algorithm for binning environmental shotgun reads. In *Annual International Conference on Research in Computational Molecular Biology*, pages 17–28. Springer.

DeSalle, R. and Goldstein, P. (2019). Review and interpretation of trends in dna barcoding. *Frontiers in Ecology and Evolution*, **7**, 302.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD’96, page 226–231. AAAI Press.

Forum, M. P. (1994). Mpi: A message-passing interface standard. Technical report, USA.

Goodwin, S., McPherson, J. D., and McCombie, W. R. (2016). Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, **17**(6), 333.

Hajibabaei, M., Singer, G. A., Hebert, P. D., and Hickey, D. A. (2007). Dna barcoding: how it complements taxonomy, molecular phylogenetics and population genetics. *Trends in Genetics*, **23**(4), 167 – 172.

Kailing, K., Kriegel, H.-P., and Kröger, P. (2004). Density-connected subspace clustering for high-dimensional data. In *IN: PROC. SDM. (2004*, pages 246–257.

Lipman, D. and Pearson, W. (1985). Rapid and sensitive protein similarity searches. *Science*, **227**(4693), 1435–1441.

Mahapatro, G., Mishra, D., Shaw, K., Mishra, S., and Jena, T. (2012). Phylogenetic tree construction for dna sequences using clustering methods. *Procedia Engineering*, **38**, 1362–1366. INTERNATIONAL CONFERENCE ON MODELLING OPTIMIZATION AND COMPUTING.

Mahé, F., Rognes, T., Quince, C., de Vargas, C., and Dunthorn, M. (2015). Swarm v2: highly-scalable and high-resolution amplicon clustering. *PeerJ*, **3**, e1420.

McDonald, D., Price, M. N., Goodrich, J., Nawrocki, E. P., DeSantis, T. Z., Probst, A., Andersen, G. L., Knight, R., and Hugenholtz, P. (2012). An improved greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. *The ISME Journal*, **6**(3), 610–618.

Munoz, R., Yarza, P., Ludwig, W., Euzéby, J., Amann, R., Schleifer, K.-H., Oliver Glöckner, F., and Rosselló-Móra, R. (2011). Release ltps104 of the all-species living tree. *Systematic and Applied Microbiology*, **34**(3), 169 – 170.

Rognes, T., Flouri, T., Nichols, B., Quince, C., and Mahé, F. (2016). Vsearch: a versatile open source tool for metagenomics. *PeerJ*, **4**, e2584.

Ruzgar, E. and Erciyes, K. (2012). Clustering based distributed phylogenetic tree construction. *Expert Systems with Applications*, **39**(1), 89–98.

Smith, T. and Waterman, M. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, **147**(1), 195 – 197.

Stone, J. E., Gohara, D., and Shi, G. (2010). Opencl: A parallel programming standard for heterogeneous computing systems. *Computing in Science Engineering*, **12**(3), 66–73.

Uralsky, L., Shepelev, V., Alexandrov, A., Yurov, Y., Rogaev, E., and Alexandrov, I. (2019). Classification and monomer-by-monomer annotation dataset of suprachromosomal family 1 alpha satellite higher-order repeats in hg38 human genome assembly. *Data in Brief*, **24**, 103708.