MISSION-T2D

Multiscale Immune System SImulator for the Onset of Type 2 Diabetes integrating genetic, metabolic and nutritional data

Work Package 2

Deliverable 2.3

Report on database pre-processing

European Commission

SEVENTH FRAMEWORK PROGRAMME

Document Information

Grant Agreement	**Nº**	600803	**Acronym**	MISSION-T2D
Full Title	Multiscale Immune System SImulator for the Onset of Type 2 Diabetes integrating genetic, metabolic and nutritional data			
Project URL	http://www.mission-t2d.eu			
EU Project Officer	**Name**	Dr. Adina Ratoi		

Deliverable	**No**	2.3	**Title**	Report on database preprocessing
Work package	**No**	2	**Title**	Clinical data provision (genetics and aging) and gut microbiota modeling

Date of delivery	**Contractual**		**M15**		**Actual**		**M15**	
Status	**Version 1.4**				**Final**		**1.4**	
Nature	**Prototype**		**Report**	X	**Dissemination**		**Other**	

Dissemination level	Consortium+EU	X
	Public	

Target Group (If Public)		Society (in general)	
Specialized research communities	X	Health care enterprises	
Health care professionals		Citizens and Public Authorities	

Responsible Author	**Name**	Stefano Salvioli	**Partner**	UniBO
	Email	stefano.salvioli@unibo.it		

Version Log			
Issue Date	**Version**	**Author (Name)**	**Partner**
24.04.2014	1.1	Gastone Castellani	UniBO
10.05.2014	1.2	Filippo Castiglione	CNR
18.05.2014	1.3	Gastone Castellani	UniBO

28.05.2014	1.4	Stefano Salvioli	UniBO

Executive Summary	I In this document are reported the results of the Task 2.3 "Integration to the overall workflow" summarised by the Deliverable 2.3 "Report on database preprocessing", regarding the pre-processing of the gut microbiota (GM) data. In this report we show all the steps we performed to pre-process the gut microbiota data in order to use them as an input for the deterministic and stochastic modelling. The crucial step is the clustering of the meta-genomic data that can be performed with variable threshold. The variable threshold is giving different cluster sizes with a different population for each Operational Taxonomic Unit (OTU).
Keywords	Meta-genomic data, Clustering procedure, Operational Taxonomic Unit (OTU) definition, Preston Plot, Relative Species Abundance (RSA).

Contents

1 Introduction

In this document are described the results of the Task 2.3. We will be focused mainly on the preprocessing of Gut Microbiota data, because the procedures for the preprocessing of the genetic and genomic data have been described in a previous report. The main goal is to pre-process the sequencing data for the Gut Microbiota in order to use them as input of deterministic and stochastic models. We choose to develop the stochastic model of GM because its stationary solution is a probability distribution of bacterial species abundance that can be compared more easily to the experimental data. In the previous report we shown as difficult is to fit the experimental data with a deterministic Lotka-Volterra model, and that the reason of this difficulty is the presence of noise and high individual variability in the experimental data. Fortunately, with a stationarity assumption and with the simplifying hypothesis of neutral interaction between bacterial species we were able to obtain a probability distribution of the bacterial species by using the Chemical Master Equation, and to show that this distribution is in good agreement with experimental data and that it can provide testable prediction on the distribution of abundance of bacterial species in different condition (diet and health status).

For this reason the crucial step in the pre-processing procedure is the definition and the identification of the Operational Taxonomical Unit (OTU). An OTU is corresponding to a bacterial species, according to the 16S RNA sequenced from faecal samples. This mapping between bacterial species and sets of similar sequences is performed by a clustering procedure, and the number of sequences in each cluster is controlled by a threshold that is computed by a similarity distance between sequences. As the similarity increase as the cluster contains a lower number of sequences, conversely, if the similarity decreases, the cluster become populated by more sequences. All these procedures are extremely time consuming and we used a dedicated server with general purpose software developed in Python, C and Mathematica. The server is a Linux 36 core cluster with about 200 Gb of RAM hosted in the Physics and Astronomy Department of the Bologna University (Partner 3) and is appropriate to perform large scale data preprocessing.

2 Deliverable Results

2.1 Gut microbiota data selection

To investigate links between diet, environment, health, age and Gut Microbiota, we analysed the data described in the paper of Claesson et al [1]. These include 178 subjects, non-antibiotic-treated, for whom we also have dietary information, and who are stratified by community residence setting:

(i) community-dwelling (n=83);

(ii) attending an out-patient day hospital (n=20);

(iii) in short-term (<6 weeks) rehabilitation hospital care (n=15);

(iv) in long-term residential care (long-stay) (n=60).

The mean subject age was 78 (±8 s.d.) years, with a range of 64 to 102 years, and all were of Caucasian (Irish) ethnicity. In the study there were included also 13 young adults with a mean age of 36 (±6 s.d.) years.

Dietary data (for 168 of the 178 subjects) were collected through a semi-quantitative, 147-item, food frequency questionnaire (FFQ), weighted by 10 consumption frequencies. The authors identified four main dietary groups:

- diet 1 (‘low fat/high fibre’),
- diet 2 (’moderate fat/high fibre’),
- diet 3 (‘moderate fat/low fibre’)
- diet 4 (‘high fat/low fibre’)

To study the individual microbiota composition, the authors sequenced amplicons of the 16S rRNA gene V4 region on a 454 Genome Sequencer FLX Titanium platform and generated 5.4 million sequence reads, with an average of 28,099 (610,891 s.d.) reads per subject.

Data were downloaded from MG-RAST server (Project ID 154) in the fasta format.

2.2 16S rRNA and OTUs determination

The comparison of 16S rRNA gene sequences is a powerful tool used for taxonomy and phylogenetic analysis. The reasons for that are several and rely on some particular features of the 16S rRNA strand, among which that [2]:

(i) it is present in almost all bacteria;

(ii) it has exactly the same function in all cells;

(iii) horizontal transfer of rRNA genes is absent or rare;

(iv) it is conserved enough in sequence and structure of be readily and accurately aligned;

(v) it contains both rapidly and slowly evolving regions (the fast regions are useful for determining closely related species, whereas the slow regions are useful for determining distant relationships).

To determine a bacteria taxonomy, one usually compares its 16S rRNA sequence with some database. For this purpose, we need to fix a similarity percentage threshold (usually 97%), such that if the query sequence overlaps a certain database sequence of at least that percentage, then we can assert that the two sequences are ‘equal’ and we can classify our bacteria through the database.

For this purpose there are many database available, such as Greengenes, SILVA or RDP.

If we want to study the bacterial ecosystem, without relying on a database, we can otherwise cluster the 16S rRNA sequences into OTUs (Operational Taxonomic Units). An OTU is a cluster of sequences which overlap of at least some percentage and which thus correspond to phylogenetically similar bacteria. In other words, OTU is an operational definition of a species or group of species. The main advantage of this approach is that, as mention above, we do not need to use a database and we can thus for example detect unclassified bacteria.

What we did was to create OTUs through UCLUST, choosing different similarity thresholds, to study the system at different taxonomic scales.

2.3 MG-RAST and preprocessing tools of sequencing data

The metagenomics RAST server (MG-RAST) [3] is an open source system which provides a large set of bioinformatics tools for the analysis of metagenomes sequencing data. MG-RAST supports shotgun metagenomic data, amplicon (16S, 18S and ITS) sequence datasets and metatranscriptome (RNA-Seq) sequence datasets.

The data uploaded in the server was preprocessed by using SolexaQA [4] to trim low-quality regions from FASTQ data. This filter trims each read from the 3' end until the quality scores exceed the threshold, which was: the lowest phred score must be 15 (base call accuracy of 96.8%) and the sequences must have at most 5 bases below this quality score.

Also, as the data was 454 sequencing data, the sequences in FASTA format were submitted to the following approach: reads which had more than two standard deviations away from the mean read length were discarded. [5]

2.4 Clustering Methods

Clustering is a generic name for the non-supervised procedures of partitioning the data in several classes in respect to certain properties.

There are various ways of defining the proper way of performing this partition. The usual approaches follow an ontological philosophy, dividing all the data in a certain number of pre-determined classes. These classes can be determined *a priori* from a biological point of view, for example dividing all the possible OTUs in the known families of bacteria. This approach avoids over-fitting the experimental noise, but limits the possibility of discovering new subclasses or entirely new species. A different approach would be to let the algorithm infer the classes while running. This method would be able to discover new variants, but it is very prone to over-fitting noise or splitting in an arbitrary way a species just to obtain the desired number of clusters.

2.5 UCLUST algorithm

UCLUST [6] is a sequences clustering method, which has the main advantages of being faster, using less memory, having an higher sensitivity and being able to classify bigger datasets, compare to other methods (such as Mothur, which compares all versus all sequences).

The core step in the UCLUST algorithm is searching a database stored in memory. UCLUST performs de novo clustering by starting with an empty database in memory.

The algorithm reads the sequences in input order, so if we have some reason to prefer some sequences to others, we should pre-order them putting at the beginning of the file those that we prefer. For example when data are available in the fastq format, we should extract the quality informations and order them by quality score.

At the beginning of the process, the first sequence in the input file will be used as first seed of the database.

Then, query sequences are processed one after the other. If UCLUST finds a match between the query sequence and a database sequence (i.e. a seed), then the query is assigned to its cluster (figure 1 upper panel), otherwise the query becomes the seed of a new cluster (figure 1 lower panel).

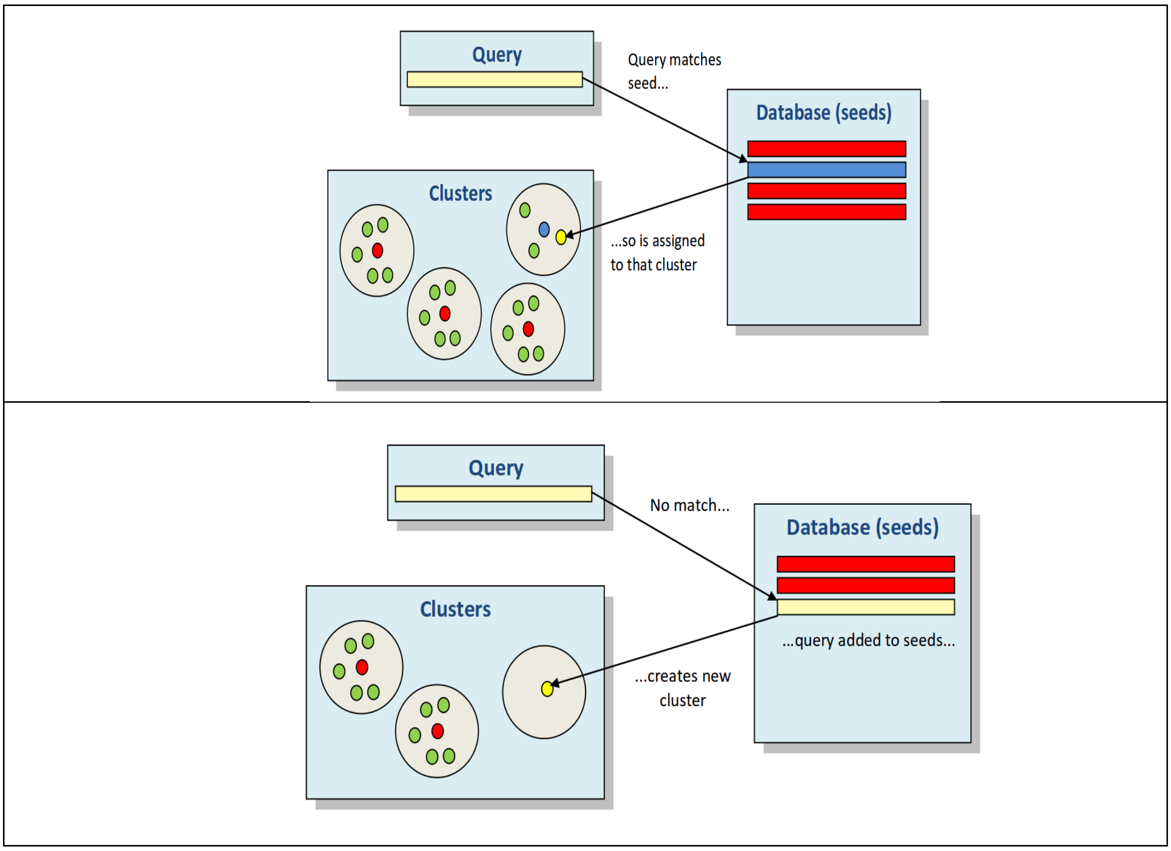


Figure 1 Schematic representation of the working of UCLUST if the query sequence does not match any seed. Figure form [6].

In this procedure, we say that a query sequence matches a database sequence if their similarity is high enough, where their similarity is calculated from a global alignment, i.e. an alignment that includes all letters from both sequences.

More precisely, similarity is computed as the number of matching (identical) letters divided by the length of the shortest sequence.

Let us observe that only seeds need to be stored in memory (because other cluster members do not affect how new query sequences are processed). This is an advantage for large datasets because the amount of memory needed and the number of sequences to search against are reduced. However, this design may not be ideal in some scenarios because it allows non-seed sequences in the same cluster to fall below the identity threshold.

2.6 UCLUST validation

In order to validate UCLUST and to check whether its way to cluster sequences was enough accurate, although it does not create OTUs with an exact maximum distance, we analyzed the distance distribution within cluster for some particular OTUs.

Thus, we clustered the sequences of one particular sample (about 27000 sequences) with UCLUST at two different similarity thresholds (97% and 90%), we computed the OTU abundances and we selected the five most abundant clusters. Then, for each of these clusters we selected the included sequences and computed the distance matrix (alignment done with PyNAST and distances computed with mothur, see below). Finally we represented the distance distributions in the form of histograms, as shown in the following figures.

At 97% of similarity, the five most abundant OTUs of the considered sample are:

OTU name:	number of sequences:
denovo468	4586
denovo512	4420
denovo209	2070
denovo197	1970
denovo475	1676

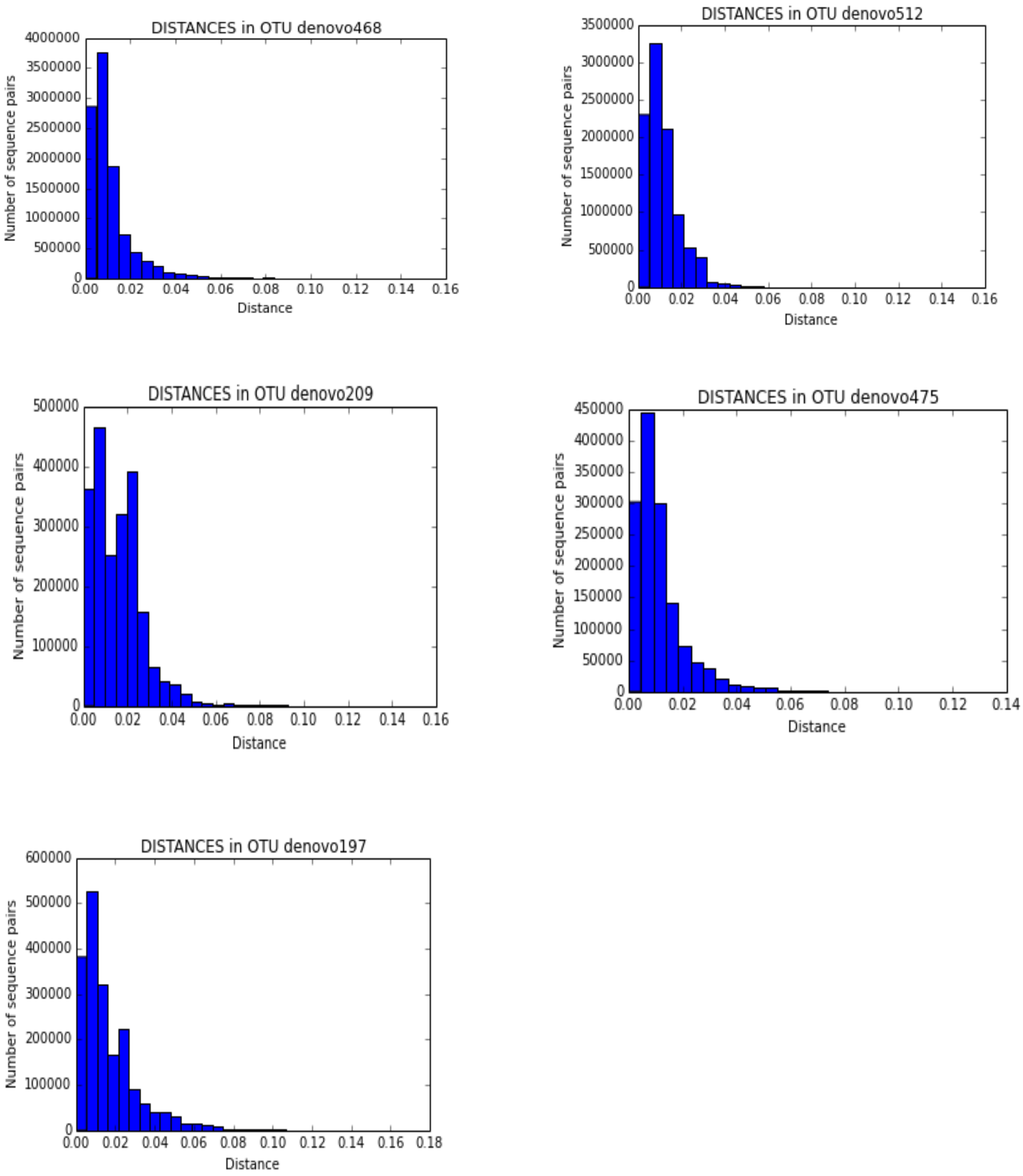


Figure 2: distance distribution for the five most abundant OTUs of one sample UCLUST clustered with a similarity threshold of 97%.

At 90% of similarity, the five most abundant OTUs of the considered sample are:

OTU name:	number of sequences:
denovo39	4810
denovo97	4671
denovo33	2420

denovo40	2399
denovo8	1942

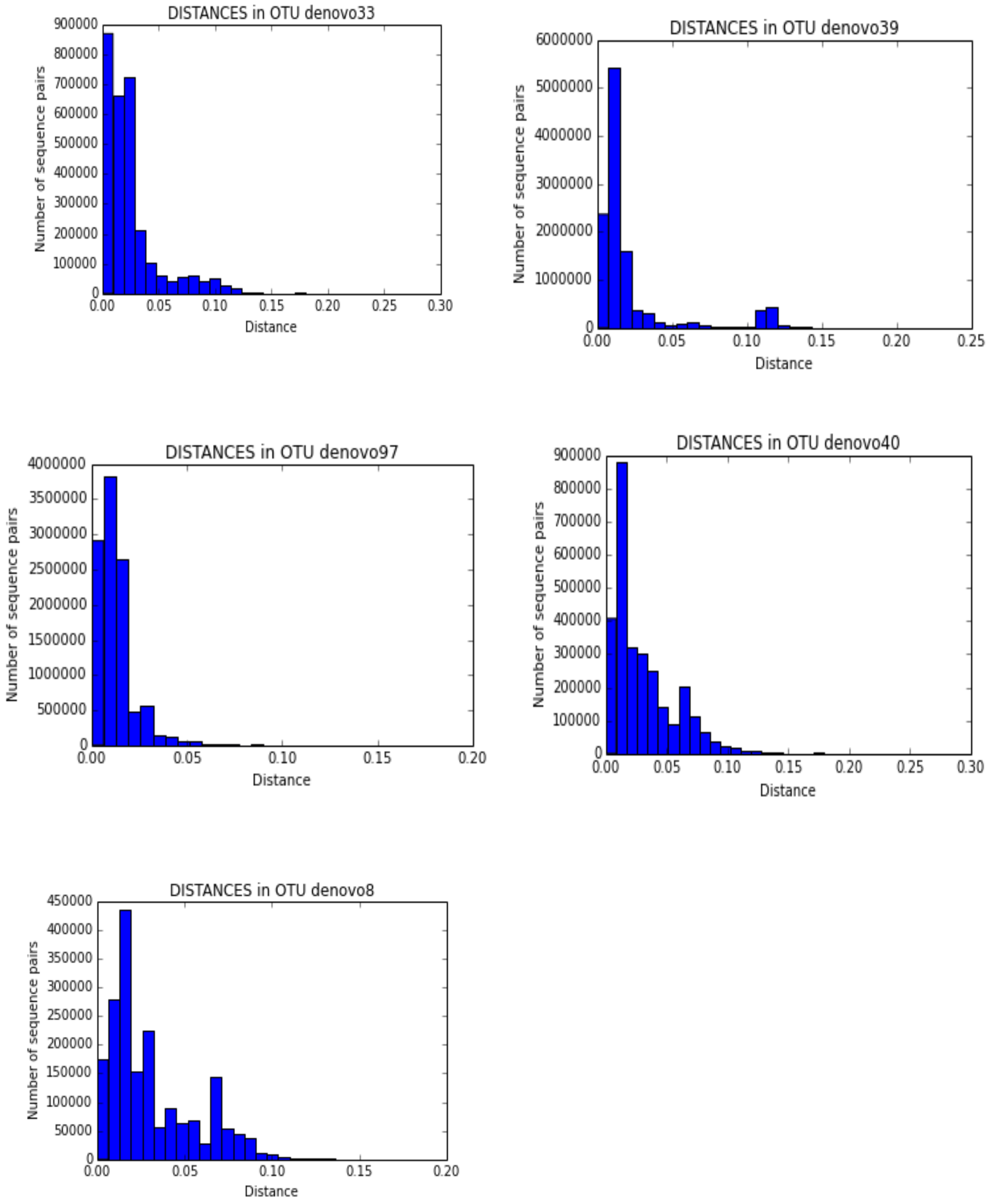


Figure 3: distance distribution for the five most abundant OTUs of one sample UCLUST clustered with a similarity threshold of 90%

From the previous figures we can see how well UCLUST builds its clusters even if it does not compare all versus all sequences and we can already underline the differences between clustering at high similarity thresholds (in this case 97%) or at low similarity thresholds (90%).

At 97% of similarity, in fact, we can see how, even in these very plentiful clusters, the maximum distance inside each cluster is approximately always under 0.1, in a scale from 0 to 1. Distance histograms show long tails on the right, as expected in a distance distribution.

Furthermore, histograms do not contain secondary peaks, which could have been a sign of multiple clusters inside the considered one.

At 90%, instead, histograms are more spread and maximum distances exceed 0.1 remaining however always approximately under 0.15. Also here, the shape of the distances distribution has, as expected, a tail towards high distances. In some OTU we begin to observe some little secondary peaks; this does not mean that UCLUST works bad, but it tells us that considering bigger OTUs, we are going to cluster together more than one species, that is what we expect when we observe the system at a higher taxonomic level.

2.7 PyNAST (NAST)

In NAST [7], an unaligned sequence is termed the ‘candidate’ and is matched to templates by comparison of 7-mers in common.

At first, a BLAST [8] pairwise alignment is performed between the candidate and the template. As a result of the pairwise alignment performed by BLAST, new alignment gaps (hyphens) are introduced between the bases of the template whenever the candidate contains additional internal bases (insertions) compared with the template (figure 4 - A, B). Any pairwise alignment algorithm must do this to compensate for nucleotides not shared by both sequences. This expansion, when intercalated with the original template spacing, results in candidates occupying more columns (characters) than the original template format (figure 4 - C). Since a consistent column count may be an option chosen by the user, the candidate template alignment is compressed back to the initial number of characters with NAST. After insertion bases are identified (figure 4 - C), a bidirectional search for the nearest alignment space (hyphen) relative to the insertion results in character deletion of the proximal place holders. Ultimately, local misalignments, spanning from the insertion base to the deleted alignment space, are permitted to preserve the global multiple sequence alignment format.

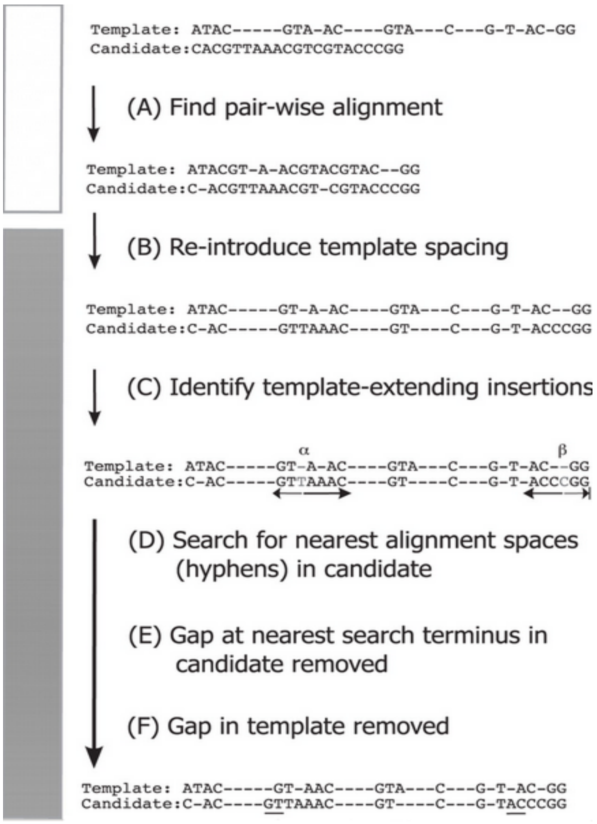

Template: ATAC-----GTA-AC----GTA---C---G-T-AC-GG
Candidate:CACGTTAAACGTCGTACCCGG
(A) Find pair-wise alignment
Template: ATACGT-A-ACGTACGTAC--GG
Candidate:C-ACGTTAAACGT-CGTACCCGG
(B) Re-introduce template spacing
Template: ATAC-----GT-A-AC----GTA---C---G-T-AC--GG
Candidate:C-AC-----GTTAAAC----GT----C---G-T-ACCCGG
(C) Identify template-extending insertions
α
β
Template: ATAC-----GT-A-AC----GTA---C---G-T-AC--GG
Candidate:C-AC-----GTTAAAC----GT----C---G-T-ACCCGG
(D) Search for nearest alignment spaces (hyphens) in candidate
(E) Gap at nearest search terminus in candidate removed
(F) Gap in template removed
Template: ATAC-----GT-AAC----GTA---C---G-T-AC-GG
Candidate:C-AC----GTTAAAC----GT----C---G-TACCCGG

Figure 4: Example of NAST compression of a BLAST pairwise alignment using a 38 character aligned template. Figure from [7].

2.8 Mothur

For distance matrix computation, we used the mothur's method dist.seqs [9]. This algorithm is well optimized, since the distances are not stored in RAM, but they are printed directly to a file. Furthermore, it is possible to ignore large distances that one might not be interested in.

To run dist.seqs an alignment file must be provided in fasta format, so sequences should be aligned before computing their distances, and this was done exactly with PyNAST.

By default an internal gap is only penalized once, a string of gaps is counted as a single gap, terminal gaps are penalized (there is some discussion over whether to penalize them or not), all distances are calculated, and only one processor is used.

Distances are computed as in the following example.

SequenceA: ATGCATGCATGC

SequenceB: ACGC - - - CATCC

Here, there would be two mismatches and one gap. The length of the shorter sequence is 10 nt, since the gap is considered as a single position. Therefore the distance would be 3/10 or 0.30.

2.9 RDP classifier

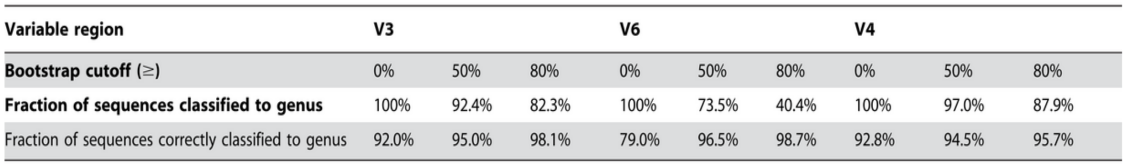
In order to compute the 16S rRNA taxonomic assignment, we chose to exploit one of the most common algorithm: the RDP classifier [10].

The RDP Classifier is distributed with a pre-built database of assigned sequences, which is used by default. Each rRNA query sequence is assigned to a set of hierarchical taxa using a naive Bayesian rRNA classifier. The classifier is trained on the known type strain 16S sequences (and a small number of other sequences representing regions of bacterial diversity with few named organisms). The frequencies of all sixty-four thousand possible 8-base subsequences (words) are calculated for

the training set sequences in each of the approximately 880 genera. When a query sequence is submitted, the joint probability of observing all the words in the query can be calculated separately for each genus from the training set probability values. Using the naive Bayesian assumption, the query is most likely a member of the genera with the highest probability. In the actual analysis, the algorithm randomly selects only a subset of the words to include in the joint probability calculation, and the random selection and probability calculation is repeated for 100 trials. The number of times a genus is most likely out of the 100 bootstrap trials gives an estimate of the confidence in the assignment to that genus. For higher-order assignments, the algorithm sums the results for all genera under each taxon.

For each rank assignment, the Classifier automatically estimates the classification reliability using bootstrapping. Ranks where sequences could not be assigned with a bootstrap confidence estimate above the threshold are displayed under an artificial 'unclassified' taxon. The default threshold is 80%.

For partial sequences of length shorter than 250 bps (longer than 50 bps), a bootstrap cutoff of 50% was shown to be sufficient to accurately classify sequences at the genus level, and to provide genus level assignments for higher percentage of sequences (figure 5) [11].



Variable region	V3			V6			V4		
Bootstrap cutoff (≥)	0%	50%	80%	0%	50%	80%	0%	50%	80%
Fraction of sequences classified to genus	100%	92.4%	82.3%	100%	73.5%	40.4%	100%	97.0%	87.9%
Fraction of sequences correctly classified to genus	92.0%	95.0%	98.1%	79.0%	96.5%	98.7%	92.8%	94.5%	95.7%

Figure 5: Of 7208 full-length 16S reference sequences from the human gut 6054 were classified at genus-level with 80% bootstrap support. With these full-length assignments as references the V3, V4 and V6 regions were extracted and re-classified at three different bootstrap thresholds, and compared with the full- length classification (last row). Figure form [11].

We can choose to use 50% as bootstrap cut-off since the accuracy is closest to the one with 80% cut-off, and the total number of sequences that could be assigned to genus level is closest to that obtained without any cut-off threshold imposed.

Other algorithms for taxonomic assignment are for example Greengenes, BLAST, mothur or RTAX.

In any case, the RDP-classifier has been found to produce the most accurate and stable results, especially for gut communities [11]. Furthermore, the RDP-classifier resulted more than 30 times faster than the Greengenes classifier.

Thus, we chose to use the RDP classifier because of its documented accuracy and stability, straightforward usage, independence of sequence alignments, high speed, and suitability for very large datasets generated by next-generation sequencing technologies.

2.10 QIIME

All sequence analysis described in the previous sections (except for the distance matrix computation), were actually executed through QIIME (Quantitative Insights Into Microbial Ecology) [12], which is a pipeline application that uses several third-party applications (UCLUST, PyNAST and RDP Classifier for our purpose).

More precisely, QIIME is an open source software package for comparison and analysis of microbial communities, primarily based on high-throughput amplicon sequencing data (such as 16S rRNA) generated on a variety of platforms, but also supporting analysis of other types of data (such as shotgun metagenomic data). QIIME takes users from their raw sequencing output through initial analyses such as indeed sequence alignment, OTU picking and taxonomic assignment. QIIME also provides tools for construction of phylogenetic trees from representative sequences of OTUs, visualization and statistical analysis.

QIIME consists of a series of scripts written with in python[14] that call the external toolboxes and perform the data conversion to allow a straightforward pipelining from one to the other. On top of this, other scripts collate these pipelines together to form a proper analysis program.

This approach allows modularity and power: most of the required programs are shipped inside QIIME, but can also be installed separately to keep up with the newest versions, and the pipelines can choose the best tool for each step. This makes confronting the results of several ways of analyzing data straightforward.

2.11 RSA distribution and Preston Plot

The Relative Species Abundance (RSA) is the distribution that tells how many species have a certain number of individuals and is very important to assert the biodiversity of an ecosystem, since it describes how common or rare a species is relative to other species. To obtain the RSA distribution for the microbiota ecosystem, after clustering the 16S rRNA sequences as previously explained, we computed the abundance of each OTU.

The results were graphed as Preston plots, that is we set the x-axes (number of individuals) in $\log_2$. This approach is commonly used to represent RSA distribution, which otherwise would have very long tails; this is a way to give more importance to the left part of the distribution, that is to its shape near the rarest species, that are of grate concern for the ecosystem biodiversity.

The following figures show the resulting Preston plots for a young microbiota; we can already assert that the distribution changes its shape as we change the similarity threshold with which we build the OTUs, that is as we change the taxonomic level at which we observe the ecosystem.

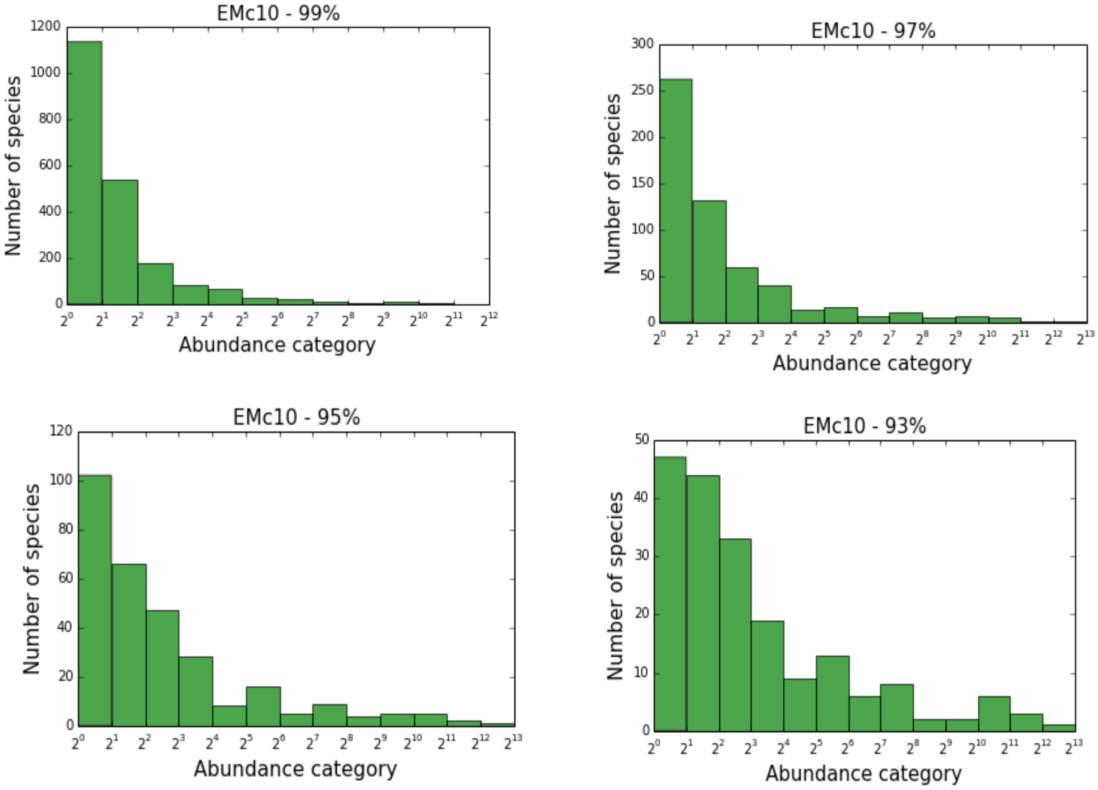
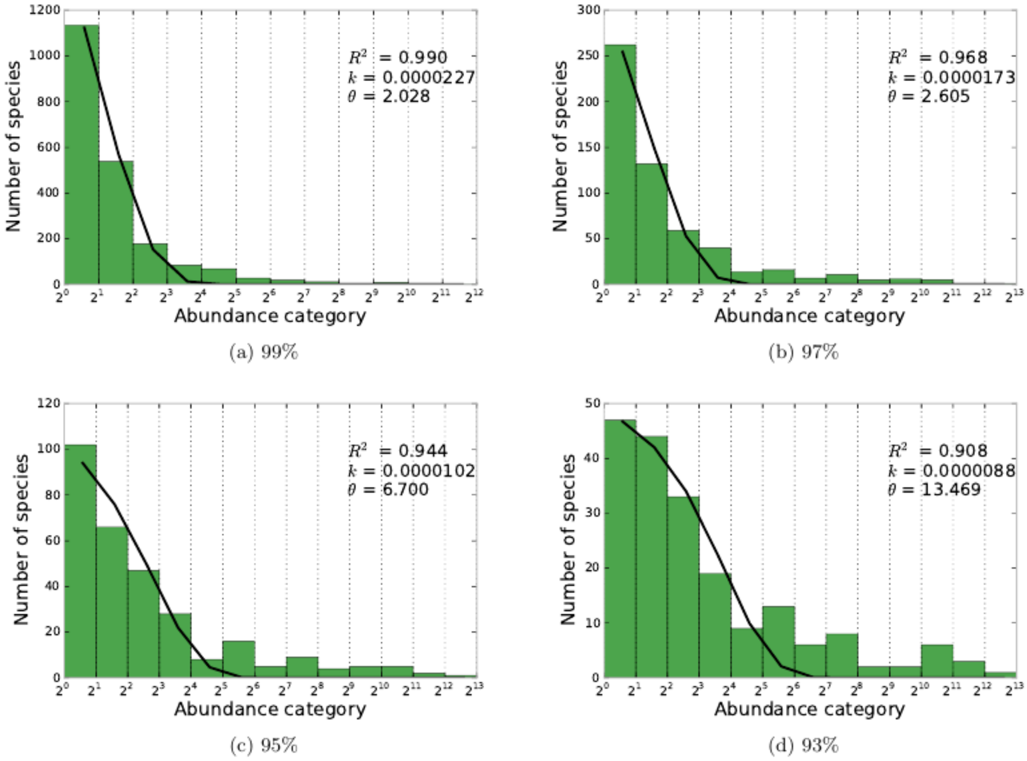


Figure 6: Preston plot of the species abundances for one young individual; figure titles specifies the similarity thresholds used to define the OTUs.

2.12 Fitting the data with the stationary distribution obtained from the stochastic model



(a) 99%

(b) 97%

(c) 95%

(d) 93%

Figure 7: Relative species abundance of gut microbiota community. Fit with gamma distribution (two parameters).

Figure 7 shows the results for one of the 13 young samples. We analyzed the RSA at different similarity thresholds, to study the ecosystem biodiversity at different taxonomic scales. We can see from this example how well our model fits our data that is how well it describes the gut microbiota ecology. Interestingly, these four figures remind us a well-known effect in ecology: the meta-community effect. It is possible to observe, starting from panel a) to d), by decreasing the similarity threshold, that the distribution become more LogNormal.

These goodness of fit can be further improved if we consider the sum of two Gamma distributions.

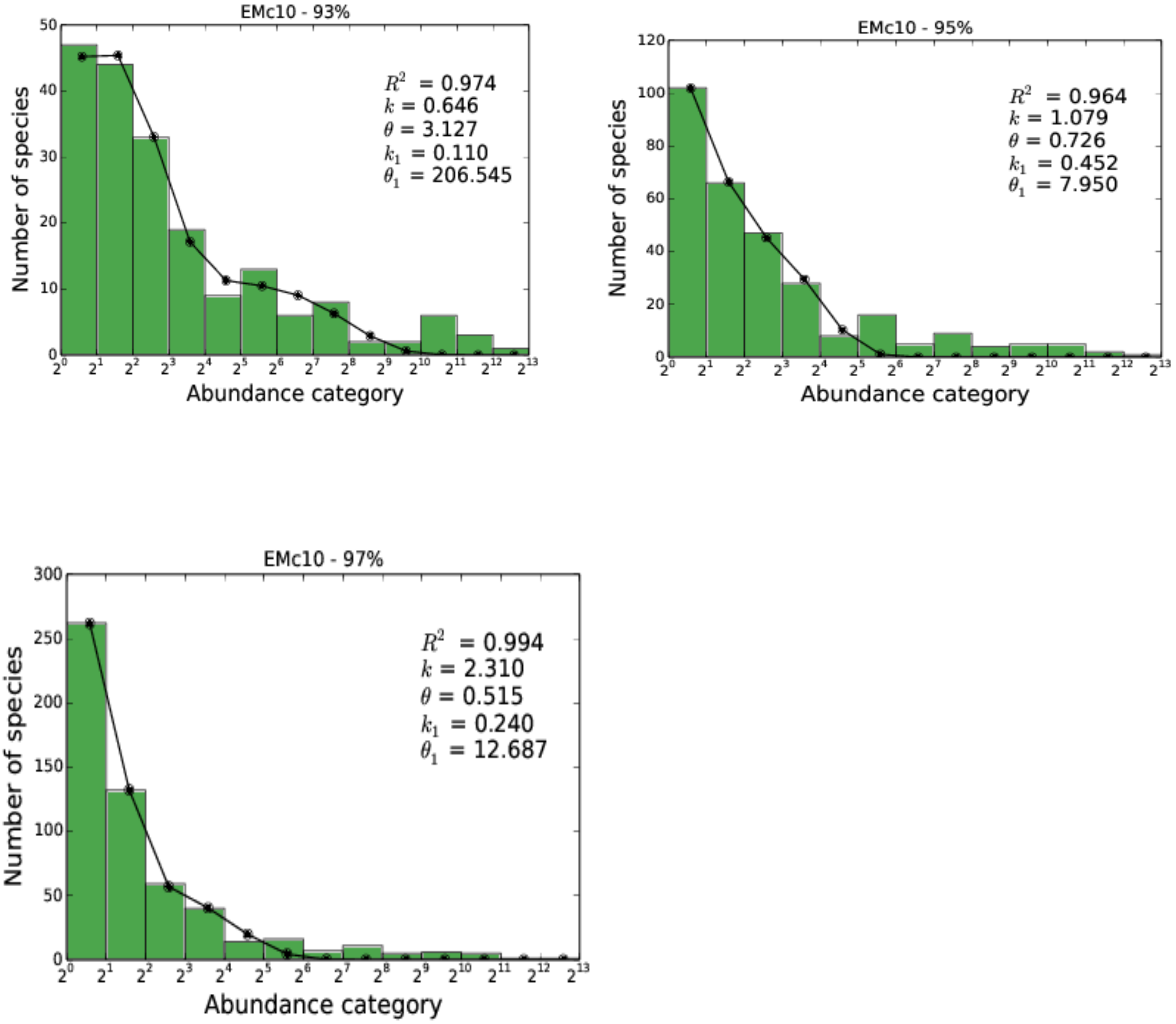


Figure 8: Relative species abundance of gut microbiota community of a young sample fitted with a sum of gamma distribution.

3 Conclusions

The proposed pre-processing method is effective for the OTUs identification and is has been used as the basis for the assessment of the RSA (Relative Species Abundance). The resulting histograms are used as input for the stochastic model and the agreement we found was very good.

The fitting with the obtained gamma distribution is very good, and the parameters of the distribution can discriminate between different diets and conditions.

The preprocessing is also giving the taxonomy of the bacterial species, so we can associate to each profiles the name of the bacteria for each OTU.

Another interesting aspect is the functional analysis; given the taxonomy of each sample, we can associate to it a metabolic profile, that it the biomolecules that this bacteria are producing.

This information will be used as input for the immune system simulator together with the bacteria distribution.

4 Bibliography

[1] M. J. Claesson et al., Gut microbiota composition correlates with diet and health in the elderly, Nature, 7410(488), 2012.

[2] J. Janda et al., 16S rRNA gene sequencing for bacterial identification in the diagnostic laboratory: pluses, perils, and pitfalls, Journal of clinical microbiology, 9(45), 2007.

[3] F. Meyer et al., The Metagenomics RAST server - A public resource for the automatic phylogenetic and functional analysis of metagenomes, BMC Bioinformatics 9(386), 2008

[4] M.P. Cox et al., SolexaQA: At-a-glance quality assessment of Illumina second-generation sequencing data, BMC Bioinformatics, 11(485), 2010.

[5] S.M. Huse et al., Accuracy and quality of massively parallel DNA pyrosequencing, Genome Biol, 8(7), 2007

[6] Uclust, extreme high-speed sequence clustering, alignment and database search, 2010. URL http://www.drive5.com/uclust.

[7] T.Z. DeSantis Jr, et al., NAST: a multiple sequence alignment server for comparative analysis of 16S rRNA genes. Nucleic Acids Research, 34(2):W394–W399, 2006.

[8] Blast, 2013. URL http://en.wikipedia.org/wiki/BLAST.

[9] Dist.seqs, 2011. URL http://www.mothur.org/wiki/Dist.seqs.

[10] Rdp, 2013. URL http://rdp.cme.msu.edu.

[11] M. J. Claesson, et al., Comparative analysis of pyrosequencing and a phylogenetic microarray for exploring microbial community structures in the human distal intestine. PloS one, 4(8):e6669, 2009.

[12] Qiime, 2013. URL http://qiime.org/.