Large Scale Explorative Oligonucleotide Probe Selection for Thousands of Genetic Groups on a Computing Grid: Application to Phylogenetic Probe Design Using a Curated Small Subunit Ribosomal RNA Gene Database

Faouzi Jaziri, Eric Peyretaillade, Mohieddine Missaoui, Nicolas Parisot, Sébastien Cipière, Jérémie Denonfoux, Antoine Mahul, Pierre Peyret, David R.C. Hill

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 350487, 9 pages
http://dx.doi.org/10.1155/2014/350487

Research Article

Large Scale Explorative Oligonucleotide Probe Selection for Thousands of Genetic Groups on a Computing Grid: Application to Phylogenetic Probe Design Using a Curated Small Subunit Ribosomal RNA Gene Database

Faouzi Jaziri,[1,2] Eric Peyretaillade,[2,3] Mohieddine Missaoui,[1,2] Nicolas Parisot,[2,4] Sébastien Cipière,[1] Jérémie Denonfoux,[2,4] Antoine Mahul,[5] Pierre Peyret,[2,3] and David R. C. Hill[1]

[1] *UMR CNRS 6158, ISIMA/LIMOS, Clermont Université et Université Blaise Pascal, F63173 Aubière, France*
[2] *Clermont Université et Université d'Auvergne, EA 4678 CIDAM, BP 10448, F63001 Clermont-Ferrand Cedex 1, France*
[3] *Clermont Université et Université d'Auvergne, UFR Pharmacie, F63001 Clermont-Ferrand Cedex 1, France*
[4] *CNRS, UMR 6023, LMGE, F63171 Aubière, France*
[5] *Clermont Université, CRRI, F63177 Aubière, France*

Correspondence should be addressed to Pierre Peyret; pierre.peyret@udamail.fr and David R. C. Hill; drch@isima.fr

Received 25 September 2013; Accepted 5 December 2013; Published 6 January 2014

Academic Editors: Y. Lai and S. Ma

Phylogenetic Oligonucleotide Arrays (POAs) were recently adapted for studying the huge microbial communities in a flexible and easy-to-use way. POA coupled with the use of explorative probes to detect the unknown part is now one of the most powerful approaches for a better understanding of microbial community functioning. However, the selection of probes remains a very difficult task. The rapid growth of environmental databases has led to an exponential increase of data to be managed for an efficient design. Consequently, the use of high performance computing facilities is mandatory. In this paper, we present an efficient parallelization method to select known and explorative oligonucleotide probes at large scale using computing grids. We implemented a software that generates and monitors thousands of jobs over the European Computing Grid Infrastructure (EGI). We also developed a new algorithm for the construction of a high-quality curated phylogenetic database to avoid erroneous design due to bad sequence affiliation. We present here the performance and statistics of our method on real biological datasets based on a phylogenetic prokaryotic database at the genus level and a complete design of about 20,000 probes for 2,069 genera of prokaryotes.

1. Introduction

The total number of species on our planet is of about 9 million, according to the latest biodiversity estimate. However, the vast majority of these species are not yet discovered and only over 1.2 million species have been already catalogued in a central database [1]. Most nondescribed species are microorganisms. Microbial communities represent the most important and diverse group of organisms living on earth. They play an important role in the functioning of ecosystems [2]. The comprehension of the role of microorganisms is then a major challenge of microbial ecology. Because of the huge microbial biocomplexity, high-throughput molecular tools allowing simultaneous analyses of existing populations are well adapted to survey microorganisms in complex environments [3].

Phylogenetic Oligonucleotide Arrays (POAs) are currently widely used and are one of the most promising approaches for studying microbial communities. They generally use oligonucleotide probes to target small subunit ribosomal RNA (SSU rRNA) genes and discriminate organisms. SSU rRNA gene is a phylogenetic biomarker largely used in the

majority of studies. However, the sequences could be highly conserved leading to some difficulties for species discrimination. Consequently, specific oligonucleotide probes selection for POAs could be a very difficult task to obtain a high resolution level [4].

Efficient oligonucleotide probes must have the following two properties: sensitive and specific. The sensitivity of a probe means its capacity to detect low levels of its complementary target in complex samples. A sensitive probe is one that is able to access its complementary sequence in the target and returns a strong signal when the target is present in the hybridized sample. The sensitivity generally increases with probe length as the binding energy for longer probe/target hybrid complexes is typically higher and hybridization kinetics are irreversible.

The specificity of a probe means its capacity to hybridize only with its complementary counterpart target. A specific probe is one that does not cross-hybridize with a nontarget sequence and returns a weak signal when the target is absent from the hybridized sample. The specificity generally decreases with the increase of probe length: short oligonucleotide probes are more specific, allowing discrimination of single nucleotide polymorphisms under optimal conditions, but at the cost of reduced sensitivity. The specificity is the most important criterion of the probes quality measure in probe design algorithms [5]. Probe design algorithms usually use specific algorithms such as suffix array method or BLAST [6] to check the specificity of probes by searching possible cross-hybridizations against datasets. However, the exponential increase of the number of sequences deposited in public databases induced an important increase in the computational capacity requirements of oligonucleotide probe design algorithms [7] and also a fundamental change in the way these algorithms are designed.

It is true that we can find fast probe design software running on regular PCs because they allow selecting probes for few DNA sequences or/and do not check the specificity of the obtained probes. The probe specificity tests against the large and ever growing biological datasets require a particular attention to develop a new generation of probe design software able to deal with high performance computing. In this context, parallel and distributed architectures such as computing clusters or computing grids [8] can provide interesting performances. Computing grids provide a promising approach to use distributed resources to meet the continuously evolving computational needs of bioinformatics tools [9]. They are particularly suited when the parallelism can be based on data splitting providing true independent computing [10]. They allow a transparent use of geographically dispersed resources for largescale distributed applications. They are adapted for time consuming algorithms that can be split into several independent jobs.

In addition to the use of known probes in POAs that allow us to simultaneously study several thousand known organisms, it is also important to design explorative probes that can detect unknown sequences not yet available in public databases and explore the vast majority of microorganisms that are still nondiscovered [3].

Here, we present a new parallelization method of a probe design algorithm to select known and explorative oligonucleotide probes using a computing grid. This software runs on the European Grid Infrastructure (EGI). EGI is a multidisciplinary grid infrastructure providing more than 250.000 CPU cores and more than 100 petabytes over 51 countries (http://www.egi.eu/). We introduced an efficient parallelization method to take advantage of the computing power available in the EGI grid to perform largescale oligonucleotides selection. We present also a new algorithm for the construction of a personal high-quality phylogenetic database that can be used to select specific, sensitive, and explorative probes targeting any prokaryotic or fungal taxonomic group, for phylogenetic oligonucleotide microarrays.

2. Related Works and Limitations

Phylogenetic Oligonucleotide Arrays (POAs), targeting the SSU rRNA genes, are known as one of the most interesting approaches to study the microbial diversity in complex environments [11]. In the last ten years, several works were done to study the biodiversity of different environments using such POAs. A microarray composed of 132 probes of length 18 mers was proposed to monitor prokaryotic microorganisms involved in sulphate reduction [12]. Another microarray considered as the most evolved POA called "PhyloChip" was developed by Brodie et al. [13] based on the Affymetrix GeneChip platform. The PhyloChip is composed of nearly 500 000 oligonucleotide probes targeting almost 9000 operational taxonomic units. This tool has been used to characterize prokaryotic communities from various ecosystems [13–17].

Additionally, several tools were proposed to select probes for phylogenetic arrays; they are discussed hereafter and in Dugat-Bony et al. [3].

The PRIMROSE program [18] was proposed to select both oligonucleotide probes and PCR primers. The probe design mechanism of PRIMROSE consists in first producing a multiple alignment for a given group of sequences. Probes are then selected and subsequently tested against an input database, to search for potential cross-hybridizations and to verify the coverage of the targeted group of sequences. PRIMROSE has been mainly used in PCR-based and FISH (fluorescent in situ hybridization) approaches [19, 20], but only a few applications of POAs using PRIMROSE have been reported [21]. The PRIMROSE tool does not allow selecting explorative probes. The ARB software package [22] proposed a probe design tool that allows selecting oligonucleotide probes with a length of 10 to 100 mers. This tool consists in searching all possible signature sequences of a targeted group of organisms specified by the user. Probes are then selected and matched against a database using the ARB Probe Match software. The ARB probe design tool has been used to design low-density custom-made POAs, composed of only a few hundreds of probes [23–25]. However, this probe design software is not well suited for large scale oligonucleotide probe design. Furthermore, it allows selecting only probes targeting known organisms and does not allow selecting explorative probes.

ARB and PRIMROSE tools allow selecting promising probes or primers for a single organism or a group of related organisms. However, emerging experimental approaches seek to simultaneously detect numerous organisms of interest thereby requiring the identification of large numbers of compatible probes [7, 26].

Oligonucleotide retrieving for molecular applications (ORMA) [27] is one of the most recent software proposed to select oligonucleotide probes. ORMA is composed of a set of scripts developed under Matlab and uses the BLAST program to check the specificity of the oligonucleotide probes selected. It allows designing probes for molecular application experiments on sets of highly similar sequences. ORMA was first applied to the design of probes targeting 16S rRNA genes, but it can also be used on any set of highly correlated sequences. This probe design tool has been used to design the HTF-Microbi-Array allowing high taxonomic level fingerprinting of the human intestinal microbial community [28].

All of these programs allow selecting probes targeting only known microbial communities with available sequences in public environmental databases. A few tools such as PhylArray [29] were designed with the possibility of selecting explorative probes for phylogenetic microarrays. PhylArray was developed with the Perl language. It allows selecting probes for a group of SSU rRNA sequences to globally survey known and unknown bacterial communities. Probe selection using PhylArray can take several days for only one large group of sequences.

In this work, we present a new parallel approach to select both known and explorative oligonucleotide probes on computing grids. The probe design strategy is based on the original algorithm PhylArray described in Militon et al. [29].

3. Material and Methods

3.1. Implementation. Our method was implemented in a program called PhylGrid 2.0. It was developed under Linux CentOS 5.4 with C++ and Perl. It uses three other programs: BLAST [6], Clustalw-MPI [30], and Opal [31].

Our approach hides the EGI grid to the user who just uses a regular computer which acts as a grid UI (User Interface: a grid component for user access to the grid). The first step was to implement the software on the User Interface (UI). This allows a direct connection to the EGI grid using a proxy authentication for the submission of multiple jobs. The main resources used by our grid application are the Workload Management System (WMS), a Berkeley Database Information Index (BDII), Computing Elements (CEs), and Storage Elements (SEs). We used the gLite middleware API commands. Submission, jobs management, and file transfer were implemented.

3.2. SSU rRNA Database Building. Probe design requires building a SSU rRNA database used as input and also as a reference database to check the specificity of all possible probes. This database must be of high quality in order to obtain the right design and to avoid wrong cross-hybridization results caused by poor sequences quality and erroneous affiliation in public environmental databases. Here, we developed a new algorithm to revisit, for more precision, the initial database described in Militon et al. [29].

All SSU sequences of the taxonomic divisions Prokaryotes (PRO), fungi (FUN), and environmental samples (ENV) downloaded from the European Molecular Biology Laboratory (EMBL) nucleotide sequence database were used as a reference to build our database carefully crafted for our probe design software. Several steps were needed. First, small subunit rRNA gene sequences (16S for prokaryotes and 18S for fungi) were extracted and filtered according to their quality and size. We kept only the sequences that met the following criteria.

(i) The sequence length is greater than 1,200 bases.

(ii) The sequence length is smaller than 1,600 bases for prokaryotic sequences and 1,800 bases for fungal sequences.

(iii) The sequence is assigned to the genus level in EMBL database (taxonomic information is extracted from the (OC) organism classification EMBL field).

(iv) The percentage of unknown nucleotides (not {A, C, G, T}) in the sequence is less than 1%.

(v) The maximum number of consecutive unknown bases must not exceed 5. The last two criteria allow removing low quality sequences.

These stringent parameters were chosen in order to allow an efficient probe design. Then, extracted sequences were grouped at the genus taxonomic rank and each group was included in its specific kingdom (prokaryote or fungi) according to the NCBI taxonomy database.

The next step consists in checking the orientation of the obtained sequences. We used BLASTN program and a reference sequence to check and correct the orientation of sequences that had been incorrectly oriented in the EMBL database.

Subsequently, a BlastClust was made on each group to eliminate redundant sequences, using the following parameters allowing a single-linkage clustering at 100% identity cutoff:

(i) -p F (nucleotide sequences);

(ii) -S 100 (similarity threshold);

(iii) -L 1 (minimum length coverage);

(iv) -b F (required coverage as specified by -L and -S on only one sequence of a pair).

Finally, for each group, we checked the homogeneity of its sequences. The aim was to eliminate sequences badly annotated and to create a homogeneous group of sequences to allow selecting specific probes for this group. This step was done using a modified version of Clustalw [32] to compute distances between sequences and the K-means method [33] to clustering sequences.

We used this algorithm to build a 16S rRNA database at the genus level. We obtained 2,069 prokaryotic genera; each is composed of a set of homogeneous sequences representing

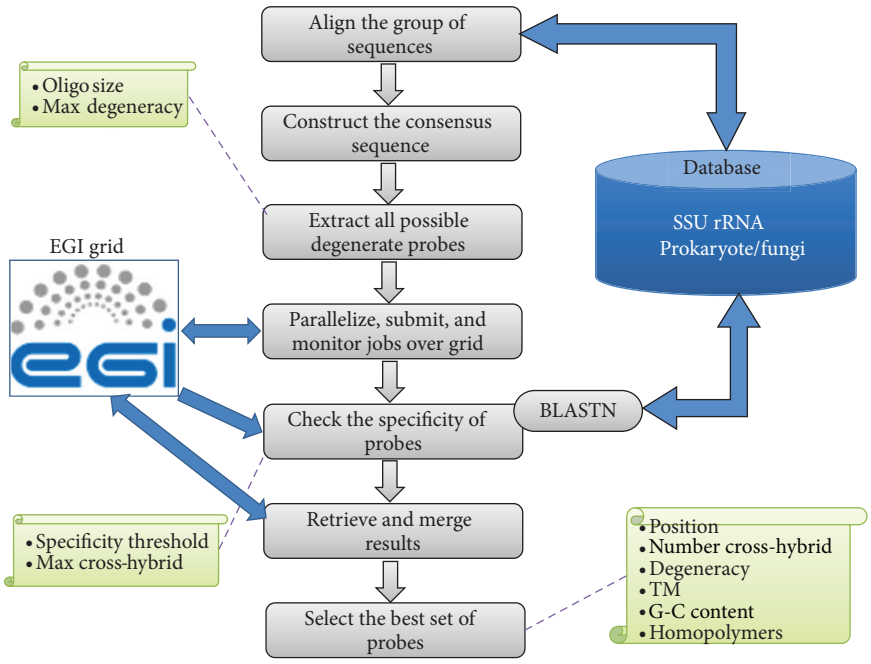

Figure 1: Summary of algorithm steps.

the whole diversity. Our algorithm can be easily adapted and used to build high-quality SSU rRNA databases for different taxonomic ranks (family, order, class, etc.).

3.3. The Probe Design Algorithm. Our algorithm uses 4 main input parameters: probe length, maximum degeneracy of a consensus probe, specificity threshold (the minimum value used to determine if the probe may hybridize with a nontarget sequence), and maximum number of cross-hybridizations. Figure 1 shows the different steps of our algorithm linked to the EGI grid.

To design probes for an input group of sequences selected by the user, a multiple sequence alignment is first made. For small groups of sequences, Clustalw-MPI [30] is used to align the sequences of the given group. However, for large groups of sequences, the multiple alignment is made in three steps to improve its quality and speed. First, BlastClust is made on each large group (using the parameters -L .98, -S 98, -p F, and -b F) to construct main subgroups of highly similar sequences. Then, sequences of each subgroup are aligned using Clustalw-MPI. Finally, Opal [31] is used to merge the obtained alignments and to reconstitute a complete alignment for the whole group.

The alignment file created is then used to construct a consensus sequence using the IUPAC degenerate nucleotide codes [34]. The aim is not only to obtain a common sequence that entirely represents the whole group of sequences targeted but also to improve alignment and correct possible sequencing errors. For example, in each column of the alignment representing a molecular site, if the number of unknown nucleotides ("N" or gap "-") is less than half the number of sequences aligned, all the unknown bases of the aligned sequences, at this position, are replaced by the specific or degenerate base calculated from all the specific nucleotides of this position. Else a gap "-" is inserted in the consensus sequence at this position.

The next step of the probe design strategy consists in incrementing a window of length "l" (l is the length of probes set by the user) along the consensus sequence to find all possible degenerate probes that do not contain gaps ("-") and whose degeneracy does not exceed the threshold value of maximum degeneracy allowed.

Subsequently, a parallelization is made to distribute all the extracted degenerate probes into "N" jobs (N is the number of jobs set by the user). For each job, all the degenerate probes are processed. Otherwise, all possible specific and explorative oligonucleotide probes are generated from each degenerate probe, using IUPAC codes [34]. These oligonucleotides are checked for cross-hybridizations against the reference SSU rRNA database initially built, using BLASTN program with the following parameters: "-W 7 -F F -S 1 -e 100 -b 20000".

Finally, all the obtained regular and explorative oligonucleotide probes are regrouped and saved in a final result file. For each degenerate or specific probe, all the associated information is provided, such as position, degeneracy, number and list of cross-hybridizations, and mismatch positions.

3.4. Parallelization Method. Selecting probes for a group of nucleic acid sequences and checking the specificity of each possible probe against a complete SSU rRNA database require a very important computation time. Our software allows running this kind of design on a computing grid. First, the user must choose the number of jobs to use. The consensus sequence, constructed from the alignment file

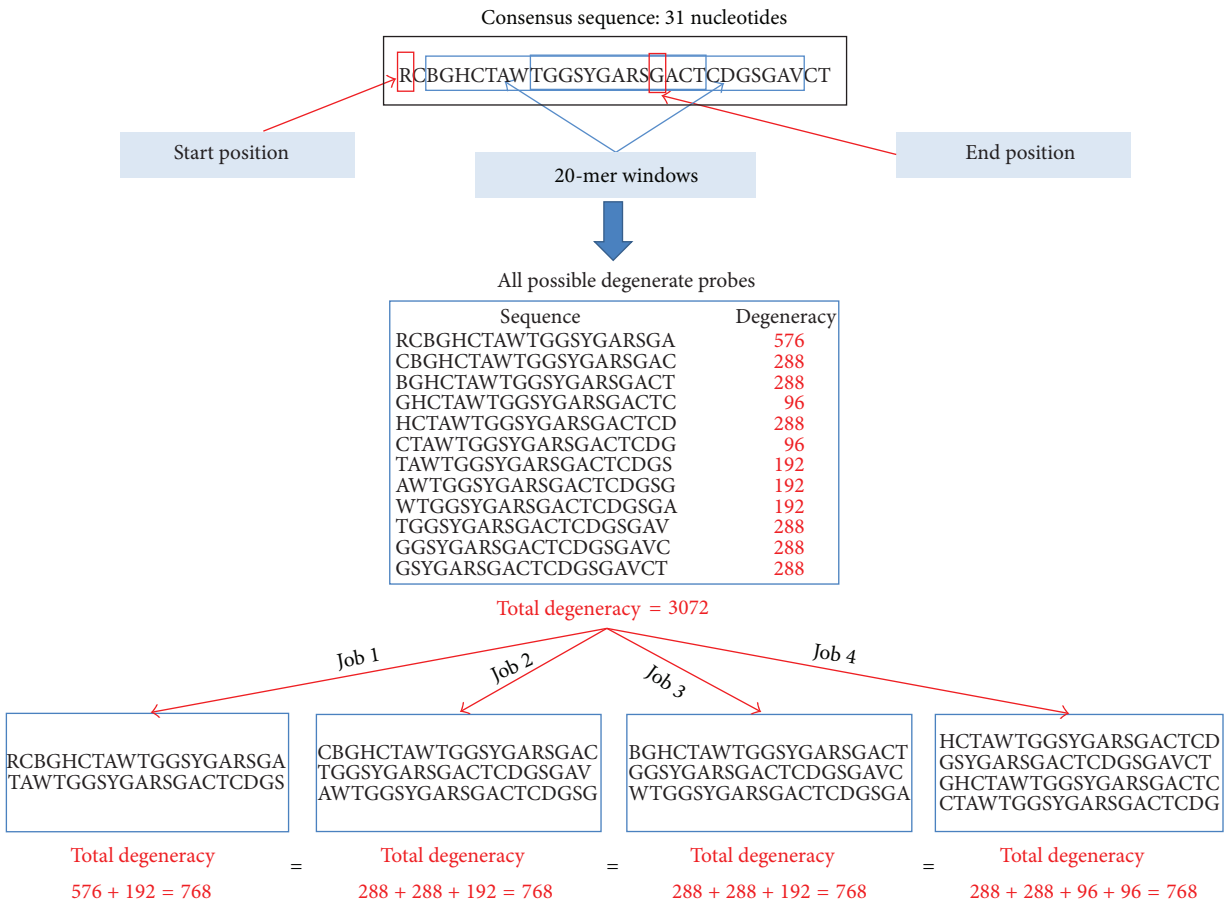

Figure 2: Parallelization strategy to define and submit jobs over the grid.

of each group of sequences, is read to extract all possible degenerate probes that do not contain gaps ("-"), based on the probe length set by the user. The degeneracy of each degenerate probe is calculated. If this degeneracy is less than "maximum degeneracy authorized by the user" (MaxDeg), the degenerate probe is saved. A weight value is calculated for each saved degenerate probe based on its degeneracy.

Once this step is performed, all valid degenerate probes saved are collected and put in the same file. This file must then be cut into "N" subfiles (N is the number of jobs set by the user) depending on the weight value of each degenerate probe and the sum of all the weight values. First, all the degenerate probes are sorted in descending order based on their weights. The mean degeneracy per subfile is then calculated based on the sum of all the weight values and the number of jobs desired. Finally, a "worst fit" algorithm [35] is used to put each degenerate probe in the largest possible free block in which this degenerate probe can be saved according to its weight. This method allows avoiding the creation of small unusable blocks by making the remainder as large as possible with the aim of making this remainder able to contain other degenerate probes. The subfiles created will have almost the same weight (Figure 2) and the same number of potential probes. Each subfile represents a job that will be submitted to the EGI computing grid.

Moreover, we have developed job monitoring scripts, with resubmission in case of failure to improve the reliability of our grid software. Three cases can be distinguished.

(i) The job submission failed: the job is resubmitted when a network route is found.

(ii) The job is submitted successfully and failed when executed: a new job is created and submitted.

(iii) The job is submitted successfully and done successfully but the other jobs are not finished: the program waits for all jobs and then merges all results in a single output file.

For running conditions, the database is copied on grid Storage Elements (SEs) accessible to all the grid jobs of a probe design. Regarding submission time, it is important not to overload the Workload Management System (WMS). Otherwise, the program may wait until each job is entirely associated with a CE of the EGI grid before submitting the next job. The following elementary configuration files are necessary to submit jobs successfully on the EGI grid.

Table 1: A comparison of the performance of the alignment method used in our software with that used in PhylArray [29], using 100 cores.

Aligned group	Number of sequences	Number of subgroups	Alignment time (seconds)		Speedup
			PhylArray	PhylGrid 2.0	
Vibrio	1,174	37	2,542	1,247	2.03
Bacillus	3,947	58	12,586	3,130	4.02

(i) JDL files: each job needs a job description language (JDL) file to be submitted on the Grid.

(ii) Script files: such files describe the elementary tasks that will be executed on the grid. The scripts contain operating system commands and Perl scripts called to perform probe design among all extracted degenerate probes. During execution, SSU rRNA database and subfiles containing degenerate probes are copied on the CE in which the job is running, and Blastn analysis is launched to test cross-hybridization.

Finally, the program is designed to be extensible by separating independently jobs in distinct designs. It creates a single data identifier for each probe design.

4. Results

In this section, we present the results obtained by our software on real data sets. We show the performance of our parallelization method compared to the original program PhylArray [29].

4.1. Database Building. We developed a new algorithm for the construction of a high-quality curated phylogenetic database, as described above. Our algorithm can be easily adapted and used to build high-quality SSU rRNA databases for different taxonomic ranks (genus, family, order, class, etc.). We used this algorithm to build a SSU rRNA database at the genus level. We obtained about 66,000 16S rRNA gene sequences representing 2,069 prokaryotic genera; each is composed of a set of homogeneous sequences representing the whole diversity. We used PhylGrid 2.0 and this database to create a complete phylogenetic oligonucleotide database composed of about 20,000 probes targeting 2,069 prokaryotic genera.

4.2. Alignment of Alignments. Dealing with the multiple sequence alignment for large groups of sequences, an alignment of alignments is achieved to improve the quality and speed of alignment. The alignment time is given in Table 1 for different groups with a varying number of sequences.

For instance, the performance of this method is 4 times faster than a simple multiple alignment when aligning the bacteria genus group "*Bacillus*."

4.3. Load Balancing Method. To distribute the probe design task on all used jobs equitably, we developed a load balancing method based on the degeneracy of all possible degenerate probes extracted from the consensus sequence constructed. To test the efficiency of our method, we compared it to the load balancing method used in the original algorithm PhylArray [29] that selects probes on a computing cluster. To distribute the computation on N processors, PhylArray splits the consensus sequence into N equal parts. Each part is then processed on a processor.

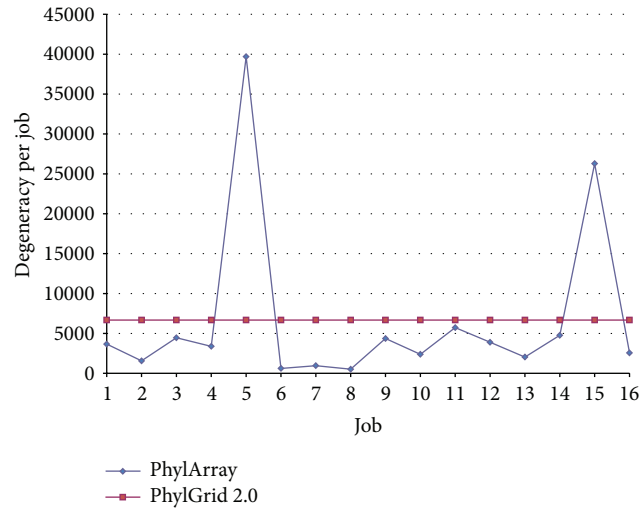

Figure 3: A comparison of our load balancing method with PhylArray [29] using 16 processors to select probes for "*Citrobacter*" group.

The comparison tests were made on real data sets, using respectively 16 jobs to select probes for the genus group "*Citrobacter*" (Figure 3), 8 jobs to select jobs for the genus group "*Haemophilus*" (Figure 4), and finally using 4 jobs to select jobs for 3 genus groups: "*Citrobacter*," "*Eubacterium,*" and "*Haemophilus*" (Table 2). This comparison shows that our method is more efficient than PhylArray. Using our method the different parts of the probe design, which processed each one on a processor, have almost the same value of degeneracy that is very close to the value of the mean degeneracy per job. For instance, as showed in Table 2, the load standard deviation between jobs is very small (0.5 probe) when using PhylGrid 2.0 compared to the high standard deviation obtained when using PhylArray (18,647.85 probes).

4.4. Use of the European Grid EGI. Our software allows users to submit parallel jobs to the EGI computing grid from Biomed Virtual Organization for the purpose of designing probes. To test the performance of our approach, we launched probes design for 10 prokaryotic genus groups simultaneously ("*Eubacterium*," "*Citrobacter*," "*Propionibacterium*," "*Neisseria*," "*Campylobacter*," "*Arcanobacterium*," "*Haemophilus*,"

Table 2: A Comparison of our load balancing method with PhylArray [29] using 4 processors to select probes for 3 genus groups.

Group	*Citrobacter*		*Eubacterium*		*Haemophilus*	
Software	PhylArray	PhylGrid 2.0	PhylArray	PhylGrid 2.0	PhylArray	PhylGrid 2.0
Mean degeneracy	26,722.75	26,722.75	37,132.25	37,132.25	20,100.75	20,100.75
Degeneracy job 1	13,068	26,723	41,435	37,133	28,600	20,101
Degeneracy job 2	41,782	26,723	43,466	37,132	32,335	20,101
Degeneracy job 3	16,381	26,723	10,273	37,132	4,314	20,101
Degeneracy job 4	35,660	26,722	53,355	37,132	15,154	20,100
Standard deviation	**14,142.836**	**0.5**	**18,647.85**	**0.5**	**12,853.09**	**0.5**

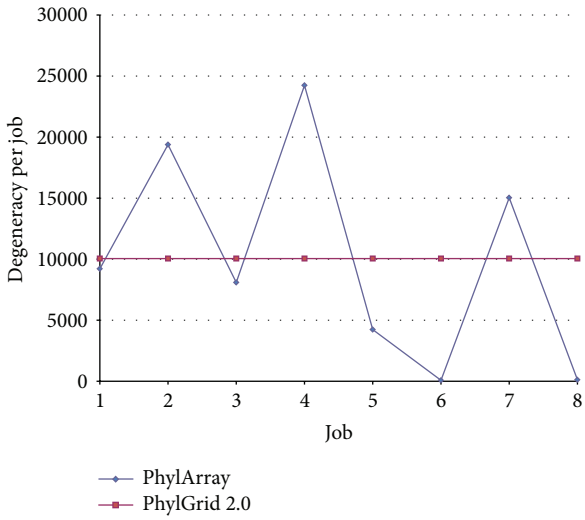

Figure 4: A comparison of our load balancing method with PhylArray [29] using 8 processors to select probes for "*Haemophilus*" group.

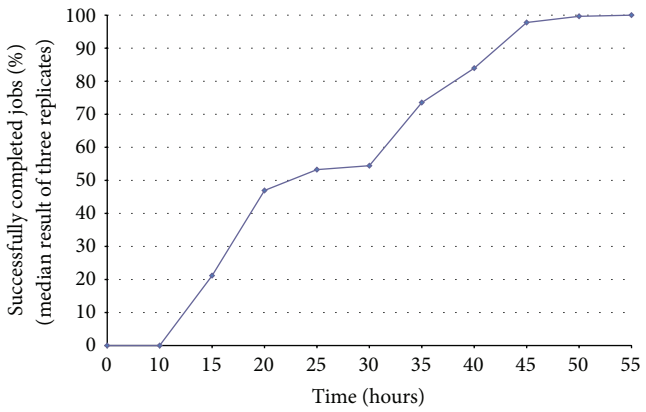

Figure 5: The median execution result of probe selection for 10 genus groups on the EGI grid using 586 jobs.

"*Kaistobacter*," "*Bacteriovorax*," and "*Riemerella*"), using the following parameters:

(i) probe length = 25;

(ii) specificity threshold = 0.88 (the probe must not have a similarity greater than or equal to 88%, with a nontargeted sequence);

(iii) maximum number of cross-hybridizations = 100;

(iv) maximum degeneracy = 2000.

This task needs more than 8 months to be processed on a single CPU core. We have launched probe designs for these groups on the EGI grid using a total of 586 jobs. We have repeated this test 3 times and the median result in terms of computational time was considered. Finally, we obtained all results successfully after less than 55 hours (with submission and waiting latency). Results are illustrated in Figure 5.

The obtained performance is here of about 106x for 586 jobs despite the submission and waiting latency of the EGI grid. Jobs submitted to a grid spend hours waiting in queues. The unavailability of some grid resources such as a Computing Element or a Storage Element can also cause the loss or blockage of jobs. This can of course increase the global computing time of our software which will however resubmit failed and lost jobs. For instance, in Figure 5, we can see a small decrease in throughput of returned completed jobs in the time window between 20 and 30 hours. This is due to the important resubmission of failed jobs at this computing phase. These jobs were submitted successfully at the beginning, but they failed or were blocked when executed.

5. Conclusions

In this work, we show that it is possible to select probes at large scale on a grid infrastructure with significant performance gains, without any particular grid submission optimizations (such as using pilot jobs). Our software allows selecting both specific and explorative (discovery of possible new species) probes with respect to excellent sensitivity and specificity. It takes advantage of the computing power offered by the EGI grid to propose at once probe design for thousands of groups. We also developed job monitoring scripts to improve the reliability and efficiency of our grid software.

The design of oligonucleotide probe on a computing grid requires optimizing the distribution of the probe design algorithm. This is why we developed an efficient parallelization method based on the degeneracy of all possible degenerate probes extracted from the consensus sequence that represents the input group. The probe selection is equally distributed

over a given number of jobs. We have compared our parallelization method with the original algorithm PhylArray [29]. We have shown that our approach is more efficient and allows a fine load balancing by sharing equitably the processing of probe selection for the input group across jobs. The comparison results of our load balancing method with that used in PhylArray—for a probe design with a mean degeneracy per job equal to 37,132.25 probes—showed that our software allowed creating jobs with a small load standard deviation of only 0.5 probe while PhylGrid generated a high load standard deviation of 18,647.85 probes between jobs. The experimental results obtained have shown that the parallel implementation of our software had significantly increased performance up to 106x when running around 600 jobs on the European Computing Grid (with submission and waiting latency). The performance of our software depends on the grid resource availability and also on the number and the size of designs that can be simultaneously launched. Hence, we have to consider Grid Computing only for large designs; otherwise, the queue waiting time and the time of data transfer on and to the grid can far exceed the computing time. For small groups of sequences, the use of a computing cluster or a multiprocessor will be more efficient than the use of a grid infrastructure for latency reasons. In our case, if we do not have tens of jobs with a job running time around 12 hours, we estimate that it is not worth submitting jobs to a computing grid where our jobs may queue for hours; instead our software suggests to consider local submissions to computing clusters.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors thank the Auvergne Regional Council and the European Regional Development Fund for the funding of Faouzi Jaziri scholarships. Nicolas Parisot was supported by the French "Direction Générale de l'Armement" (DGA). This work was also supported by the program Investissements d'avenir AMI 2011 VALTEX.

References

[1] C. Mora, D. P. Tittensor, S. Adl, A. G. B. Simpson, and B. Worm, "How many species are there on Earth and in the ocean?" *PLoS Biology*, vol. 9, no. 8, Article ID e1001127, 2011.

[2] P. G. Falkowski, T. Fenchel, and E. F. Delong, "The microbial engines that drive Earth's biogeochemical cycles," *Science*, vol. 320, no. 5879, pp. 1034–1039, 2008.

[3] E. Dugat-Bony, E. Peyretaillade, N. Parisot et al., "Detecting unknown sequences with DNA microarrays: explorative probe design strategies," *Environmental Microbiology*, vol. 14, no. 2, pp. 356–371, 2012.

[4] J. Zhou and D. K. Thompson, "Challenges in applying microarrays to environmental studies," *Current Opinion in Biotechnology*, vol. 13, no. 3, pp. 204–207, 2002.

[5] M. D. Kane, T. A. Jatkoe, C. R. Stumpf, J. Lu, J. D. Thomas, and S. J. Madore, "Assessment of the sensitivity and specificity of oligonucleotide (50mer) microarrays," *Nucleic Acids Research*, vol. 28, no. 22, pp. 4552–4557, 2000.

[6] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, no. 3, pp. 403–410, 1990.

[7] D. Zhu, Y. Fofanov, R. C. Willson, and G. E. Fox, "A parallel computing algorithm for 16S rRNA probe design," *Journal of Parallel and Distributed Computing*, vol. 66, no. 12, pp. 1546–1551, 2006.

[8] E.-G. Talbi and A. Y. Zomaya, *Grid Computing For Bioinformatics and Computational Biology*, vol. 1 of *Wiley Series in Bioinformatics*, John Wiley & Sons, New York, NY, USA, 2007.

[9] I. Foster and C. Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure*, vol. 1, Morgan Kaufmann, Boston, Mass, USA, 2004.

[10] N. Jacq, C. Blanchet, C. Combet et al., "Grid as a bioinformatic tool," *Parallel Computing*, vol. 30, no. 9-10, pp. 1093–1107, 2004.

[11] M. Wagner, H. Smidt, A. Loy, and J. Zhou, "Unravelling microbial communities with DNA-microarrays: challenges and future directions," *Microbial Ecology*, vol. 53, no. 3, pp. 498–506, 2007.

[12] A. Loy, A. Lehner, N. Lee et al., "Oligonucleotide microarray for 16S rRNA gene-based detection of all recognized lineages of sulfate-reducing prokaryotes in the environment," *Applied and Environmental Microbiology*, vol. 68, no. 10, pp. 5064–5081, 2002.

[13] E. L. Brodie, T. Z. DeSantis, D. C. Joyner et al., "Application of a high-density oligonucleotide microarray approach to study bacterial population dynamics during uranium reduction and reoxidation," *Applied and Environmental Microbiology*, vol. 72, no. 9, pp. 6288–6298, 2006.

[14] E. L. Brodie, T. Z. DeSantis, J. P. M. Parker, I. X. Zubietta, Y. M. Piceno, and G. L. Andersen, "Urban aerosols harbor diverse and dynamic bacterial populations," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 1, pp. 299–304, 2007.

[15] T. C. Hazen, E. A. Dubinsky, T. Z. DeSantis et al., "Deep-sea oil plume enriches indigenous oil-degrading bacteria," *Science*, vol. 330, no. 6001, pp. 204–208, 2010.

[16] N. Weinert, Y. Piceno, G.-C. Ding et al., "PhyloChip hybridization uncovered an enormous bacterial diversity in the rhizosphere of different potato cultivars: many common and few cultivar-dependent taxa," *FEMS Microbiology Ecology*, vol. 75, no. 3, pp. 497–506, 2011.

[17] K. M. Handley, K. C. Wrighton, Y. M. Piceno et al., "High-density PhyloChip profiling of stimulated aquifer microbial communities reveals a complex response to acetate amendment," *FEMS Microbiology Ecology*, vol. 81, no. 1, pp. 188–204, 2012.

[18] K. E. Ashelford, A. J. Weightman, and J. C. Fry, "PRIMROSE: a computer program for generating and estimating the phylogenetic range of 16S rRNA oligonucleotide probes and primers in conjunction with the RDP-II database," *Nucleic Acids Research*, vol. 30, no. 15, pp. 3481–3489, 2002.

[19] S. Fraune, R. Augustin, F. Anton-Erxleben et al., "In an early branching metazoan, bacterial colonization of the embryo is controlled by maternal antimicrobial peptides," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 107, no. 42, pp. 18067–18072, 2010.

[20] K. Bers, K. Sniegowski, P. Albers et al., "A molecular toolbox to estimate the number and diversity of Variovorax in the environment: application in soils treated with the phenylurea herbicide linuron," *FEMS Microbiology Ecology*, vol. 76, no. 1, pp. 14–25, 2011.

[21] D. Blaskovic and I. Barák, "Oligo-chip based detection of tick-borne bacteria," *FEMS Microbiology Letters*, vol. 243, no. 2, pp. 473–478, 2005.

[22] W. Ludwig, O. Strunk, R. Westram et al., "ARB: a software environment for sequence data," *Nucleic Acids Research*, vol. 32, no. 4, pp. 1363–1371, 2004.

[23] A. Loy, C. Schulz, S. Lücker et al., "16S rRNA gene-based oligonucleotide microarray for environmental monitoring of the betaproteobacterial order 'Rhodocyclales'," *Applied and Environmental Microbiology*, vol. 71, no. 3, pp. 1373–1386, 2005.

[24] H. Sanguin, A. Sarniguet, K. Gazengel, Y. Moënne-Loccoz, and G. L. Grundmann, "Rhizosphere bacterial communities associated with disease suppressiveness stages of take-all decline in wheat monoculture," *New Phytologist*, vol. 184, no. 3, pp. 694–707, 2009.

[25] M. R. Liles, O. Turkmen, B. F. Manske et al., "A phylogenetic microarray targeting 16S rRNA genes from the bacterial division Acidobacteria reveals a lineage-specific distribution in a soil clay fraction," *Soil Biology and Biochemistry*, vol. 42, no. 5, pp. 739–747, 2010.

[26] D. Zhu, Y. Fofanov, R. C. Willson, and G. E. Fox, "ProkProbePicker (PPP): a fast program to extract 16S rRNA-targeted probes for prokaryotes," in *Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS '05)*, pp. 41–47, Las Vegas, Nev, USA, June 2005.

[27] M. Severgnini, P. Cremonesi, C. Consolandi, G. Caredda, G. De bellis, and B. Castiglioni, "ORMA: a tool for identification of species-specific variations in 16S rRNA gene and oligonucleotides design," *Nucleic Acids Research*, vol. 37, no. 16, p. e109, 2009.

[28] M. Candela, C. Consolandi, M. Severgnini et al., "High taxonomic level fingerprint of the human intestinal microbiota by Ligase Detection Reaction—Universal Array approach," *BMC Microbiology*, vol. 10, article 116, 2010.

[29] C. Militon, S. Rimour, M. Missaoui et al., "PhylArray: phylogenetic probe design algorithm for microarray," *Bioinformatics*, vol. 23, no. 19, pp. 2550–2557, 2007.

[30] K.-B. Li, "ClustalW-MPI: ClustalW analysis using distributed and parallel computing," *Bioinformatics*, vol. 19, no. 12, pp. 1585–1586, 2003.

[31] T. J. Wheeler and J. D. Kececioglu, "Multiple alignment by aligning alignments," *Bioinformatics*, vol. 23, no. 13, pp. i559–i568, 2007.

[32] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acids Research*, vol. 22, no. 22, pp. 4673–4680, 1994.

[33] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithms: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[34] A. Cornish-Bowden, "Nomenclature for incompletely specified bases in nucleic acid sequences: rcommendations 1984," *Nucleic Acids Research*, vol. 13, no. 9, pp. 3021–3030, 1985.

[35] D. S. Johnson, "Fast algorithms for bin packing," *Journal of Computer and System Sciences*, vol. 8, no. 3, pp. 272–314, 1974.