ISSN 2047-3338

Finding the motif from DNA Sequences using Grid Computing System

Trang Hong Son[1], Tran Van Lang[2] and Le Van Vinh[3]
[1]Hoa Sen University, Vietnam Ministry of Education and Training
[2]Institute of Applied Mechanics and Informatics, Vietnam Academy of Science and Technology
[3]HCM City University of Technical Education, Vietnam Ministry of Education and Training
[1]trangson2211@yahoo.com, [2]tvlang@vast-hcm.ac.vn, [3]vinhlv@fit.hcmute.edu.vn

***Abstract*—The motif finding problem is one of high complex problems, as well as needs much time. Therefore, the accuracy and time consuming are two important goals in this problem. Many algorithms were proposed for solving the problem. However, execution time is still a challenge needing more research. In this paper, we proposed a parallel solution based on improving the Uniform Projection algorithm. Moreover, we implemented the program on the Grid computing environment to get better performance.**

***Index Terms*—Biology Sequence, Finding Motif and Grid Computing**

I. Introduction

THE number of DNA and protein sequences is increasingly being uncovered. In this sequence, the number of nucleotides or amino acids is usually very large. Thus, we need to know which the subsequences that are the same or nearly the same in the biological sequences. This is a problem with high complexity and spending a lot of processor time. The subsequence is known as a motif. Motif finding problem is often used for finding transcription factor binding sites which are helpful to decipher the regulation of gene expression.

The problem is high complexity because we could not get a set of template motifs, as well as not know its position in the sequences. There were many algorithms proposed in the past. Each algorithm has its own advantages and disadvantages. In general, the motif finding algorithms are based on two general approaches: using the Sample Driven Approach – SDA method (Consensus [1], Gibbs sampling [2], MEME [3]) to check on a set of given sequences; and using the Pattern Driven Approach – PDA method (Teiresias algorithm [4], MITRA [5]) to draw the motif in the space that contains the result set.

Among the existing algorithms, the Brute Force is the simplest algorithm. The algorithm's idea is using an exhaustive search that always finds the best results. However, the execution time required too much. It is not suitable for long sequences or the large number of sequences. In addition, there are some following finding motif algorithms:

The Consensus algorithms, proposed by Hertz [1], based on the Brute Force method, but it removes the cases which are predicted as not well one. As a result, the search time is greatly reduced. The weakness of the algorithm is that when it removes search space too early, some good results could be missed, leading to local interference.

The Gibbs Sampling [2] and MEME algorithm [3] based on probabilistic models. Those approaches start with a certain probability distribution. After some iteration, they use the criterions of probability so that the probability model converged to a better state. The algorithms are said to have less "interference" than the Consensus algorithm. Especially, the speed of algorithms is faster and it could search the more complex motif than the Consensus algorithm. The downside of this algorithm is the dependence on the starting model. If the initial model is not good, the algorithm may be converge slowly and return unexpected results.

The Teiresias algorithm's idea [4] is completely different from the three algorithms mentioned above. In the algorithm, the motif is considered as a vocabulary. The small size words are found firstly. Then, they are combined to be larger and larger motif until we have the expected motif. This method was proved to be very effective and could find the complex motif. But, because it is special approach, the method is difficult to improve and less common than other algorithms.

Another PDA-based approach, MITRA [5] checks on the entire space of possible motifs, and store the data in a tree structure. Furthermore, the algorithm uses artificial intelligence approach in the searching process. In each of iteration, the search space's size decreases, so searching time is reduced significantly. In addition, this algorithm can easily to be parallelized for faster search.

Recently, some novel algorithms were proposed, such as qPMS7 [6], PMS6 [7] also aiming to decrease the computational cost. Other algorithms use parallel techniques to get better performance, for examples, Christopher or Liu's algorithm [8], [9]. The parallel techniques often used in kind of algorithms are MPI (Message Passing Interface) and GPU (Graphics Processing Unit).

This paper presents our results in parallelization of the algorithm proposed by Raphael: Uniform Projection [10], the

algorithm is a combination of Radom Projection algorithm [11] and Expectation Maximization method. In addition, the program was implemented on the Grid computing system in order to achieve better performance.

The rest of paper was organized as follows. The section 2 was introduced briefly to finding motif problem. The approaches to solve the problem effectively and quickly on a grid computing environment presented in section 3. The section 4 presents some experimental results, and the Section 5 for conclusions and future work.

II. MOTIF FINDING PROBLEM

A. The Motif

A sequence motif is a nucleotide sequence or an amino acid that contains (or can contain) a certain biological functions.

The finding motif problem can be defined as follows: Given a set of DNA or protein sequences, find the subsequences which are the same, or nearly the same (have a mutation in a few nucleotides) occurring in all sequences (being implanted in each of the individual sequence).

For example, there are five following sequences. The case that there is not mutation in motif:

1. cctgatagacgctatctggctatcc**acgtacgt**aggtcctctgtgcga atctatgcgtttccaaccat
2. agtactggtgtacatttgat**acgtacgt**acaccggcaacctgaaacaa acgctcagaaccagaagtgc
3. aa**acgtacgt**gcaccctctttcttcgtggctctggccaacgagggctg atgtataagacgaaaatttt
4. agcctccgatgtaagtcatagctgtaactattacctgccacccctatt acatctt**acgtacgt**ataca
5. ctgttatacaacgcgtcatggcggggtatgcgttttggtcgtcgtacg ctcgatcgtta**acgtacgt**c

We found the motif in five sequences is: **acgtacgt**.

In case with two mutations, consider five DNA sequences as follows:

1. cctgatagacgctatctggctatcc**aGgtacTt**aggtcctctgtgcga atctatgcgtttccaaccat
2. agtactggtgtacatttgat**CcAtacgt**acaccggcaacctgaaacaa acgctcagaaccagaagtgc
3. aa**acgtTAgt**gcaccctctttcttcgtggctctggccaacgagggctg atgtataagacgaaaatttt
4. agcctccgatgtaagtcatagctgtaactattacctgccacccctatt acatctt**acgtCcAt**ataca
5. ctgttatacaacgcgtcatggcggggtatgcgttttggtcgtcgtacg ctcgatcgtta**CcgtacgG**c

We found similar motifs as nucleotide sequences shown in bold in the above.

The motif finding problem has some following difficulties:

- Do not know the pattern of motif.
- Do not know where motif appears in the sequence.
- The motif may differ from the sequences.

B. The alignment, profile matrix and consensus

Let a set S with t DNA sequences s_1, s_2, …, s_t. In each sequence, randomly choose l nucleotide subsequences. Aligning the subsequences, we obtain a matrix, called "alignment matrix". The alignment matrix includes t x l elements. With the above example, we have:

```
......aGgtacTt.........
...CcAtacgt............
.......acgtTAgt........
...acgtCcAt............
.......CcgtacgG........
```

The following alignment matrix includes 5 x 8 elements:

```
aGgtacTt
CcAtacgt
acgtTAgt
acgtCcAt
CcgtacgG
```

From the alignment matrix, we can derive the number of occurrences of each nucleotide (or amino acid) in each column. And the profile matrix will be created. Given alignment matrix (t x l), we obtain 4 x l profile matrix. For example, the above alignment matrix, the profile matrix is:

```
A   3 0 1 0 3 1 1 0
C   2 4 0 0 1 4 0 0
G   0 1 4 0 0 0 3 1
T   0 0 0 5 1 0 1 4
```

With profile matrix, we can obtain a sequence, each nucleotide in this sequence is chosen so that it matched the highest number in the column of corresponding profile matrix. This sequence is called Consensus. The Consensus score is calculated by the following formula, where Pe is profile matrix:

$$Score = \sum_{j=1}^{l} \max_{1 \le i \le 4} Pe_{ij}$$

For example, the Consensus of above profile matrix is **ACGTACGT**.

In summary, with the subsequences were selected randomly from five sequences in above template, the alignment, profile matrix and Consensus as follows Table 1.

Table 1: Example for alignment, profile matrix and Consensus

Aligment		A	G	g	T	a	C	T	T
		C	c	A	T	a	c	G	T
		A	c	g	T	T	A	G	T
		A	c	g	T	C	c	A	T
		C	c	g	T	a	c	g	G
Profile	**A**	3	0	1	0	3	1	1	0
	C	2	4	0	0	1	4	0	0
	G	0	1	4	0	0	0	3	1
	T	0	0	0	5	1	0	1	4
Consensus		A	C	G	T	A	C	G	T

From above definitions and concepts, the motif finding problem can be stated as follows:

Given a set of t DNA or protein sequences, find a set of t sequences have the length l (called l-mer), such that each l-mer appear in a sequence, and Consensus of these sequences have score is highest.

III. THE SOLUTION METHODS

Our research aims to parallelize Uniform Projection method. The algorithm is combined of Random Projection and Expectation maximization algorithm.

A. The Expectation Maximization algorithm

This algorithm is based on probabilistic model. It helps to find the set of motif better than the original set only through some iterations. The main idea of algorithm as follows:

With a set S of t positions, $S = \{s_1, s_2, ..., s_t\}$ is the input, by using probabilistic model, after a few iterations, it will take this position set into better state. Here, the paper uses s_i as the location of the subsequence in a sequence.

This algorithm has advantages is that it converges fast and returns a set of motif location much better than the initial set. However, it much depends on the initial set. This means that the algorithm may be converge slowly, and return unexpected results if initial set S is not good.

B. The Random Projection algorithm

As the mention above, Expectation Maximization (EM) algorithm converges fast, and can find motif effectively, but it depends a lot on the input set S. The Random Projection can find a good initial set for the EM algorithm. The combination of these two algorithms will improve the efficiency in motif finding.

The algorithm assumes that l-mer could be a motif, if they must be the same in at least k positions. The algorithm tries to take out all l-mer which are the same at least k positions, and hashes them into a bucket. The bucket is a set S of l-mer's positions in the sequences. The set S is used in the EM algorithm. This algorithm could create a lot of good initial set S for the EM algorithm. But, the creation of the entire space of all projections can lead to excess. In addition, because the computational time of a projection is very long, the algorithm's execution time increases significantly with a large number of projections.

C. The Uniform Projection algorithm

The Random Projection algorithm is used to sample the space of all projections. Each good projection passing the threshold will produce an enriched bucket that is refined to create the motif. However, the random sampling in the space of all projections is not effective way. Compared with Random Projection algorithm, the Uniform Projection algorithm reduced from 20% to 50% of the projections, but it does not affect the success rate of motif finding problem.

In this algorithm, instead of random sampling, it selects projections for sampling. The projections P_1, P_2, ..., P_N are made many times, as follows:

- First, building the projections P_1, P_2, ..., P_{M1} covering all 1-mer.
- Adding the projection P_{M1+1}, P_{M1+2}, ..., P_{M2} so that P_1, P_2, ..., P_{M2} covering all the 2-mer.
- Continue until all j-mer are covered.
- In this process, a new projection P_{M+1} is added when it is distinguished from the P_1, P_2, ..., P_M.

Parameter j is not selected priority. Instead, the algorithm selects m projections so that when m increases, all the j-mers are covered.

C. Parallelization of the Uniform Projection algorithm

With the problems needing big data, the computational cost is usually very large. One of the effective approaches is parallelization of the algorithm and implement on parallel or distributed systems. The motif finding problem is one of kind of those problems. The Uniform Projection algorithm was described by the flowchart at Fig. 1.

As shown in the flowchart, there are two main works are done in the algorithm:

- Create a set of projections. (Step 1)
- Refined the set S to be used in the EM algorithm. (Step 2)

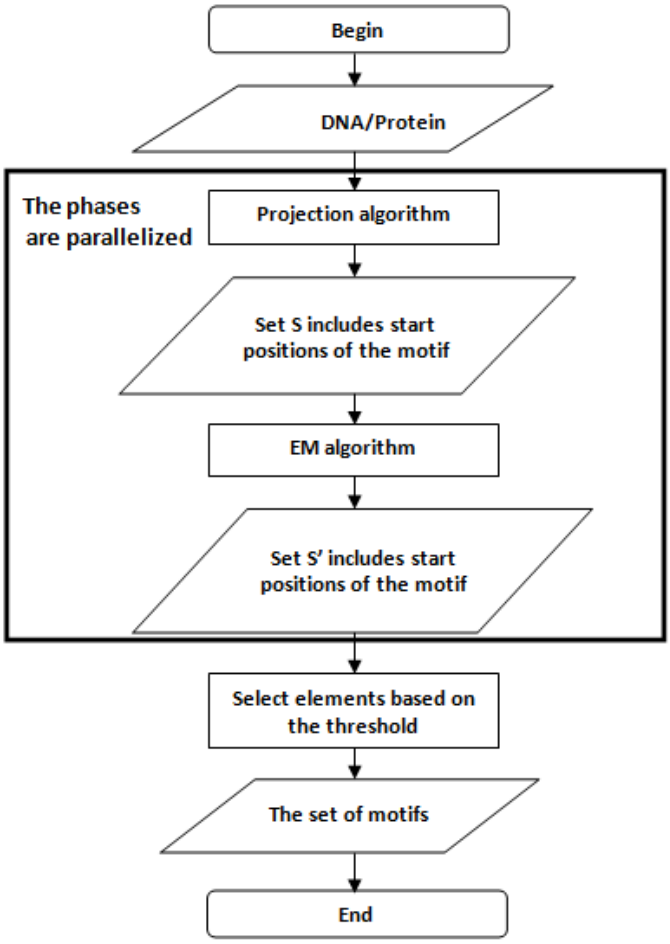

Figure 1. Flowchart of sequence algorithm to find motif

Through testing in the first step, the algorithm does not depend on the number of input data, and the process of finding the projection is done relatively quickly. However, in step 2, the time to refine the set S takes a long time. Meanwhile, with a projection P_i, there are countless set S must be refined. Thus, the computational time of the problem depends entirely on the refining process.

Besides, the finding solution of the problem from the certain projection function P_i is completely independent of the other P_i. As shown in Fig. 1, after the set of projection functions $P=\{P_i\}$ are refined, the new set S' corresponding to each projection function be created. The process of calculation to find results for each projection function does not depend on

each other. Based on the characteristic, the Uniform Projection algorithm could be parallelized for saving computational time. The parallel model is presented in Fig. 2.

Data sets (sets of DNA/Protein sequences) are transmitted to each computer (*Node*) at the initial time, and there is not the data transmission between the *Nodes* at the running time. Besides, a set of projection function $P = \{P_1, P_2, ..., P_n\}$ are divided to each *Node*. Suppose, there are m computers ($m<n$), each *Node* is received n/m projection functions and execute

L	*d*	1 process	2 processes	Ratios	Performance
14	4	6 (s)	3 (s)	6/3 = 2	2/2 = 1
15	4	9 (s)	5 (s)	9/5 = 1.8	1.8/2 = 0.9
16	5	15 (s)	8 (s)	15/8 = 1.87	1.87/2 = 0.935
17	5	23 (s)	12 (s)	23/12 = 1.92	1.92/2 = 0.96
18	6	36 (s)	19 (s)	36/19 = 1.89	1.89/2 = 0.945

Table 1: Results in testing in parallel and sequential mode

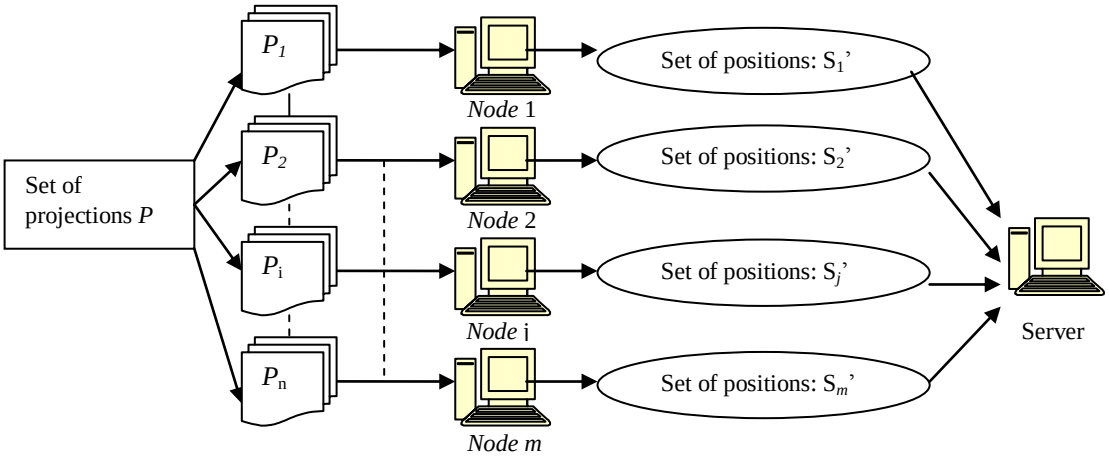

Figure 2. Parallel model of algorithm

them independently. After all tasks are finished, results are transmitted to the *Server*. At this computer, final results will be computed from the output data of the *Nodes*.

IV. EXPERIMENTAL RESULTS

The program was deployed on IOIT-HCM grid system at Institute of Applied Mechanics and Informatics (IAMI), Vietnam Academy of Science and Technology (Fig. 3) [12]. The data was downloaded from the Data Bank NCBI (National Center for Biotechnology Information).

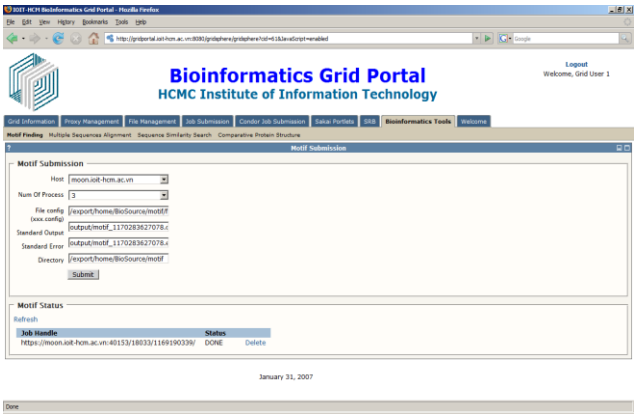

Figure 3. Grid Portal at IAMI

A. First data

The program was tested with the synthetic data: **@synthetic_05_500.fasta**. Number of sequences $t = 5$, with length $n= 500$ nucleotides, the projection size $k=7$. *(l, d)* pairs were used: (14, 4), (15, 4), (16, 5), (17, 5), (18, 6).
The time in parallel mode (2 processes) was reduced nearly by half compared with in sequential mode (1 process). Results were shown in Table 1 and Fig. 4.

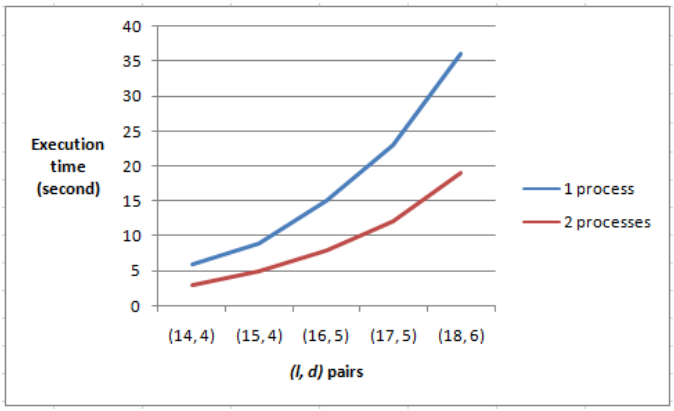

Figure 4. Testing with 1 and 2 processes

B. Second data

The program was tested with the yeast data: **@yeast_03_500.fasta**. Number of sequences $t = 3$, with length $n= 500$ nucleotides, the projection size $k=7$, motif length $l=20$, number of mutation $d=4$. The number of processes was used: 2, 3, 6, 9, 12, 15, 30, 45, 60, 90. (Table 2). Some figures of the program (Fig. 4 and Fig. 5) are showed in the Appendix section.

Number of process	Execution time (second)	Number of process	Execution time (second)
2	143	15	75
3	104	30	80
6	71	45	81
9	72	60	81
12	73	90	84

Table 2: Results in testing with the yeast data

V. CONCLUSIONS

The motif finding is one of the problems that need very large execution time. Using the high-performance environment to reduce computational time is an inevitable solution. In our research works, the Uniform Projection algorithm was parallelized and implemented on the Grid computing system. It showed that while the accuracy of algorithm was not changed, the execution time was reduced significantly. In the future, we aims to continue improve and apply the motif finding problem to solve other related problems in bioinformatics.

REFERENCES

[1] Hertz GZ, Hartzell GW, Stormo GD: Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Comput Appl Biosci*, 1990, 6:81-92

[2] Lawrence CE, Altschul SF, Boguski MS, Liu JS, Neuwald AF, Wootton JC: Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 1993, 262:208-214.

[3] Bailey TL, Elkan C: Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 1995, 21:51-80.

[4] I. Rigoutsos and A. Floratos, Combinatorial Pattern Discovery in Biological Sequences: The TEIRESIAS Algorithm. *Bioinformatics*, vol.14, no.1, pp.55-67, 1998.

[5] E. Eskin and P.A. Pevzner. Finding Composite Regulatory Patterns in DNA Sequences. *Bioinformatics*, vol.18, no.1, pp.354-363, 2002.

[6] Hieu Dinh, Sanguthevar Rajasekaran, and Jaime Davila. qPMS7: A fast algorithm for finding (l, d) – motifs in DNA and protein sequences. PLoS One, 2012.

[7] Shibdas Bandyopadhyay, Sartaj Sahni, and Sanguthevar Rajasekaran. PMS6: A fast algorithm for motif discovery. IEEE 2012.

[8] Christopher T. M., Jonathan G., Julian H. D., *et al*. Parallelizing Tompa's exact algorithm for finding short motifs in DNA. *PDPTA' 11*, USA, 2011.

[9] Yongchao Liu, Bertil S., Doulas L. M. An ultrafast scalable many-core motif discovery algorithm for multiple GPUs. *IEEE IPDPS 2011*.

[10] Benjamin Raphael, Lung-Tien Liu, George Varghese. A Uniform Projection Method for Motif Discovery in DNA Sequences. *IEEE Transactions on Bioinformatics and Computational Biology*, Nov, 2004

[11] Jemery Buhler, Martin Tompa, Finding Motifs Using Random Projections, Journal of computational biology, Volume 9, Number, 2002.

[12] Tran Van Lang, *Grid computing: building computing system and deploying applications* (in Vietnamese), Vietnam Education Publishing House, 2008, 196p.

APPENDIX

The results in the testing with second data:

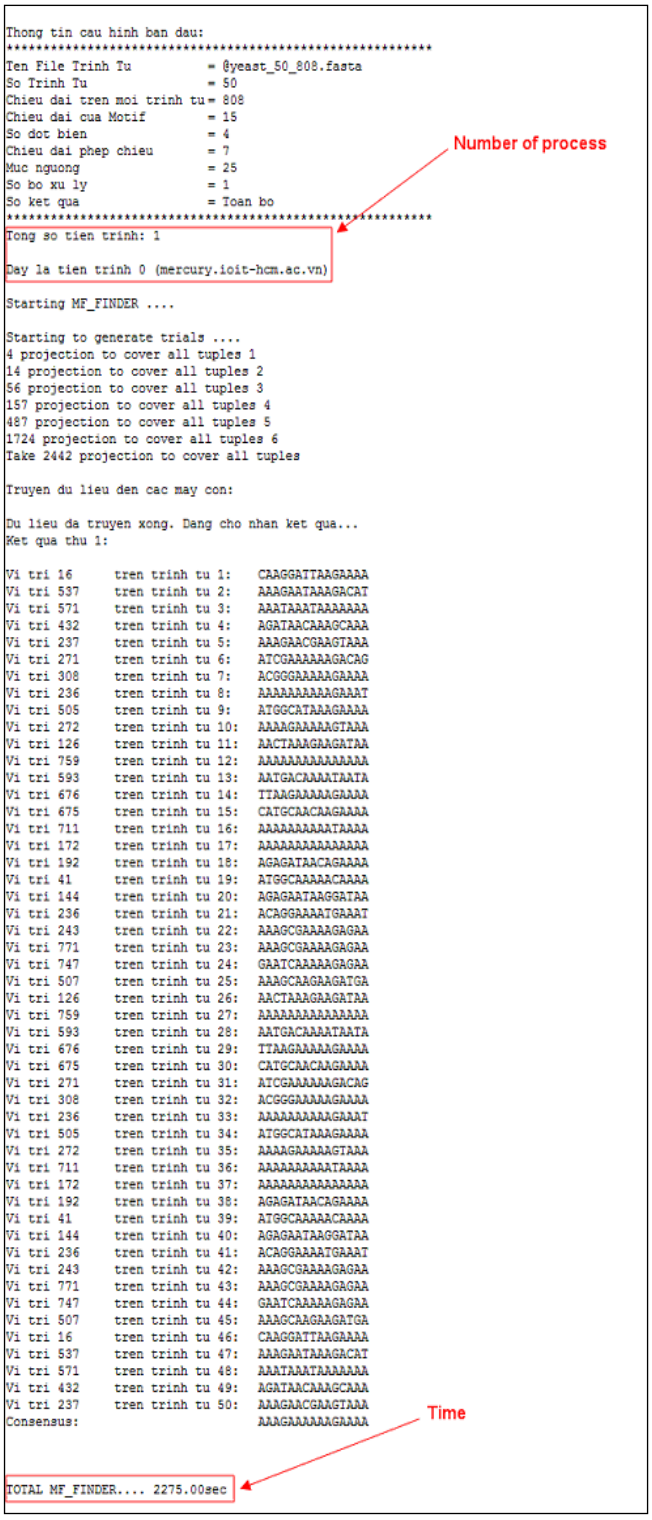

Figure 4: Using 1 process

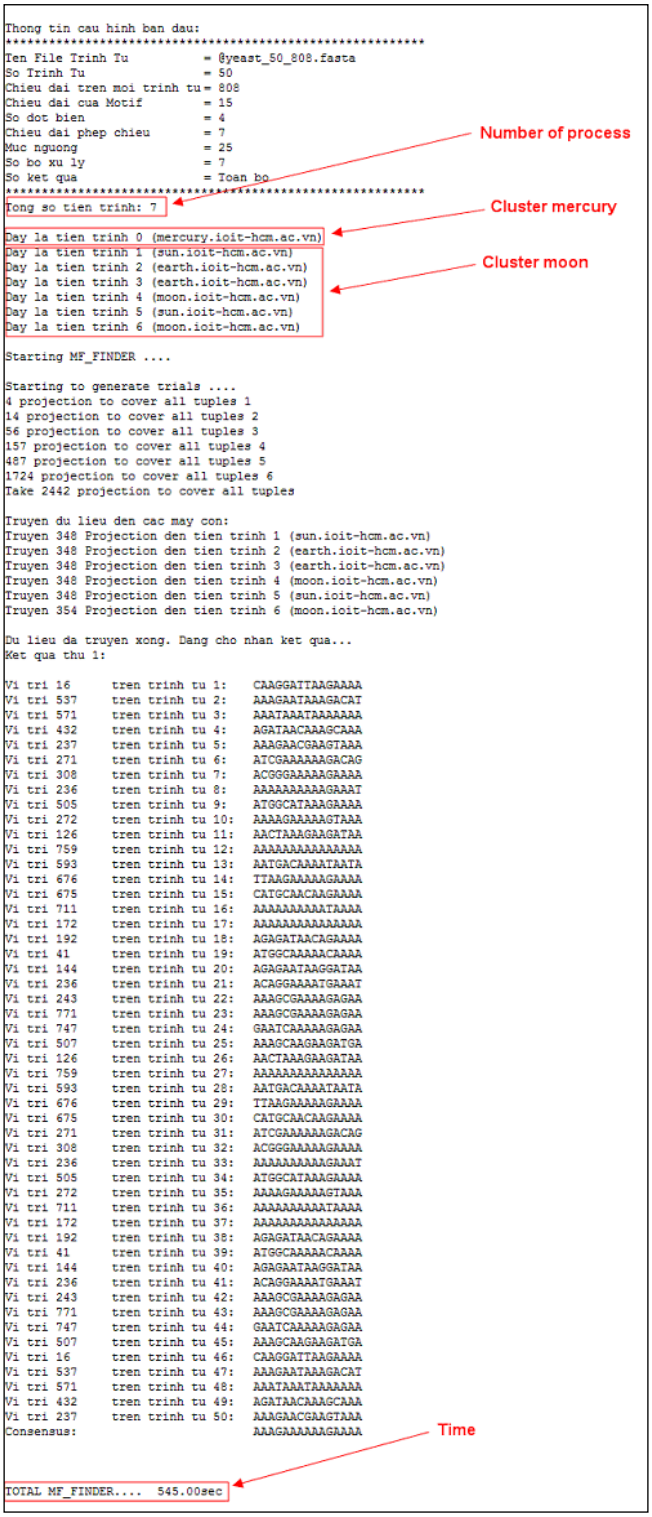

Figure 5: Using 7 process (Run on 2 clusters)