

## Section 1 - You and Your Background

### Personal Data



<b>Title</b>	Mr
<b>Name</b>	Josep
<b>Surname</b>	RIUS
<b>Position</b>	Postgraduate (student with first University degree or equivalent)
<b>Date of Birth</b>	08/09/1985
<b>Gender</b>	MALE
<b>Nationality</b>	SPAIN
<b>University/Organisation</b>	University of Lleida
<b>Department</b>	Computer Science Department
<b>Organisation Address</b>	Jaume II, 69
<b>Town</b>	Lleida
<b>post code</b>	25001
<b>Country</b>	SPAIN
<b>Email</b>	jrius@diei.udl.cat
<b>Telephone</b>	+34 676 88 20 69
<b>Fax</b>	+34 973 702 702

## Section 2: Your Organisation

### Your Organisation



<b>Organisation Legal Status</b>	University
<b>Your Scientific Background</b>	Engineering & Technology
<b>Your Research Group</b>	GCD: Distributed Computing Group
<b>Name of your Group Leader</b>	Concepció Roig Mateu
<b>Nationality of your Group Leader</b>	SPAIN
<b>Email of your Group Leader</b>	roig@diei.udl.cat
<b>URL of your research group www page</b>	http://gcd.udl.cat

**Section 3: Your Visit**

**Virtual Visit**

Is this an application for a virtual visit?



no

**When do you plan to come?**



Preferred Start Date (dd/mm/yyyy) 01/07/2010

Expected Duration in weeks 13

HPC Access Center

EPC

**Scientific host**



	Host Researcher Name	Host Researcher Email	Host Researcher Telephone	Host Department	Organisation	Country	Contacted This Host?
1.	Malcolm Atkinson	mpa@nesc.ac.uk	+44 (0)131 651 4040	School of Informatics	University of Edinburgh	UNITED KINGDOM	Yes
2.(optional)	Jano van Hemert	jvhemert@nesc.ac.uk	+44 (0)131 650 9820	School of Informatics	University of Edinburgh	UNITED KINGDOM	Yes
3.(optional)							
Briefly describe what you expect to gain from collaborating with your scientific host					<p>First of all, to access to HPC supercomputer (at EPC) to test our improvements on Parallel-TCoffee in order to evaluate the proposed ideas.</p> <p>Also, Gain knowledge about the electronic Science and discover its real applications.</p> <p>Furthermore, the data-intensive research group have agreed to provide visitor space and research facilities in the new Informatics Forum so that I can work with them.</p>		

**Section 4: Your Proposed Project**

**Project**



Project Title	Multiple sequence alignment
1. Main Field	Life Sciences & Biotech
2. Specific Discipline	Life - Other - Life Sciences & Biotech
<b>More about the code:</b>	
Is there an existing serial code?	Yes
If yes, how big is it?	more than 10000 lines
What language is it written in?	C
if other, please specify:	
How much of the code did you write yourself?	0 percent
Is there an existing parallel code?	Yes
What language is it written in?	C++
How was it parallelised?	MPI
if other, please specify:	
How big is it?	more than 10000 lines
How much of it did you write yourself?	11-25 percent
Libraries and Packages used:	mpich2 libgtop2

**Your motivation for a visit - what do you intend to do?**



Benchmarking:	Main motivation	Code development:	Secondary motivation
Collaborative project:	Secondary motivation	Consultancy:	If time permits
Data Analysis:	If time permits	Establishing Academic Link:	Secondary motivation
Optimisation:	Main motivation	Parallelisation:	Main motivation
Porting code:	Not interested	Production runs:	If time permits
Training:	Not interested	Visualisation:	Main motivation
Other:	Test the proposed improvements on Parallel-TCoffee. Grid and Web integration.		

**Tell us about your programming experience**



	<b>Level</b>	<b>Years experience</b>
--	--------------	-------------------------

<b>Unix</b>	Advanced	More than 5 years
<b>Fortran</b>	No experience	Not applicable
<b>C</b>	Advanced	2-5 years
<b>C++</b>	Advanced	2-5 years
<b>Message Passing</b>	Intermediate (low )	Less than 1 year
<b>Open MP</b>	No experience	Not applicable

**Please characterise your typical production runs**



<b>Tell us about your present computing resources</b>	
Machine architecture (please specify)	Cluster
Processor speed (please specify)	2400 MHz
Processor type (please specify)	Intel(R) Core(TM)2 Quad CPU Q6600 cache size: 4096 KB
Number of processors	65-128
Typical execution time per run	6-12 hours
<b>Please estimate the computing resources that you would expect to use during your visit</b>	
Total CPU requirements (CPU hours)	25.000-35.000
Please specify the value = (elapsed time of a single run)*(number of CPU used in a single run) * (total number of runs)	1.5 hours * 256 CPU * 75 runs
Number of processors	128-256
Total Memory requirements	512 Gb - 1 Tb
Temporary disk space requirements for a single run	1GB
Library requirements	mpich2, libgtop2
Compatible architectures (Select one or more, as applicable)	Clusters of SMP
Please justify your choice of resources (e.g. CPU requirements, no. of processors, compatible architectures)	Scalability application testing needed depending on the size of the problem and the number of processors.

**Section 5: Statement of Support**

**Statement of Support**



<b>Title</b>	Dr
<b>Name</b>	Francesc
<b>Surname</b>	Solsona
<b>Position</b>	Department Director
<b>University/Company</b>	University of Lleida
<b>Address</b>	C/ Jaume II, 69
<b>Country</b>	SPAIN
<b>Town</b>	Lleida
<b>post code</b>	25001
<b>Email</b>	francesc@diei.udl.cat
<b>Telephone</b>	+34 973 70 27 35
<b>Fax</b>	+34 973 70 27 02

## Section 6: Attachments

### Curriculum vitae



Josep Rius Torrentó has achieved the Technical Engineering in Computer Systems title by the Polytechnic School of the University of Lleida (UdL) in 2006. In 2008 he finished the Master's Degree in Engineering of Free Software and one year later he received the Engineering in Computer Science degree, all of these by the same university. Nowadays, he is pursuing the PhD degree in the Computer Science Department at the University of Lleida. His current research interests include cluster and Multicenter scheduling, and p2p computing.

On the other hand, he shows a special concern for the economic science. In this area, he received the degree in Business Sciences in 2009 from the University of Lleida, and is simultaneously studying "Business Administration and Management" degree and "Market Research and Techniques" degree, both by the Open University of Catalonia(OUC).

Referring to investigation projects, he has also taken part in "CoDiP2P: Distributed computing in Peer-to-Peer networks" project by the Indra multinational and University of Lleida agreement, from September 2007 to February 2008, whose Principal Investigator was Francesc Solsona Tehàs. Besides, he is now involved in a "Management of computational resources in heterogeneous systems" project by the Ministry of Education and Science (MEC), which started in August 2009 and is still in process with Concepció Roig Mateu as Principal Investigator.

### List of publications



CompP2P: A P2P Computing System.  
CEDI 2007: II Congreso Español de Informática.

CoDiP2P: a Distributed Computing Architecture Oriented to P2P Environments.  
XIX Jornadas de Paralelismo, 2008.

CoDiP2P: A Peer-to-Peer Architecture for Sharing Computing Resources.

DCAI 2008: International Symposium on Distributed Computing and Artificial Intelligence (2008) 293–303.

A New Reliable Proposal to Manage Dynamic Resources in a Computing P2P System.

PDP 2009: 17th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (2009).

A Reinvestment Mechanism for Incentive Collaborators and Discouraging Free Riding in Peer-to-Peer Computing.

Minisymposium on High Performance Computing applied to Computational Problems in Science and Engineering (2009).

A new credit-based incentive mechanism for p2p scheduling with user modeling.

AP2PS 2009: The First International Conference on Advances in P2P Systems. pages 85–91, 2009.

## Project Proposal



### 1. BACKGROUND INFORMATION.

#### 1.1. T-COFFEE (TC).

TC [1] is a multiple sequence aligner method that combines the consistency-based scoring function COFFEE [2] with the progressive alignment algorithm. TC provides an improvement in accuracy compared to most methods based on a progressive strategy, due to errors made in the initial alignments cannot be rectified later as the rest of the sequences are added in. In contrast, TC introduces a library generated using a mixture of global and local pair-wise alignments in order to reduce the greediness and increase the accuracy. However, the introduction of these improvements has penalized TC in speed as compared to the most commonly used alternatives. TC is divided into three main stages:

- \* Primary Library: The primary library contains a set of pair-wise alignments between all of the sequences to be aligned. By default, it is generated by combining the ten top-scoring non-intersecting local alignments constructed with the Lalign program and all the global pair-wise alignments obtained with the ClustalW method. In the library, each alignment is represented as a list of pair-wise residue matches and a sequence identity weight is assigned to each pair of aligned residues in order to reflect the correctness of a constraint. This stage is the most time and memory consuming, limiting its applicability to no more than 100 sequences on a typical workstation.

- \* Extended Library: The extended library allows TC to reduce the errors made in the initial alignments. The extension of the library is a re-weighting process where the new weights, for a given pair of sequences, also depend on information from the other sequences in the set.

- \* Progressive Alignment strategy: To obtain the multiple sequence alignment, first of all, pair-wise alignments are made to produce a distance matrix between all the sequences. The distance matrix is a matrix of similarity values between the pairs of sequences used to generate a guide tree by the neighbor-joining method. The guide tree is a phylogenetic tree, whose order is followed by the progressive alignment strategy to obtain the multiple sequence alignment. Progressive alignment consists of aligning the first two closest sequences using dynamic programming. This alignment uses the weights in the extended library above to align the residues in the two sequences. Then, the next two closest sequences are aligned, or a sequence is added to the existing alignment of the first two sequences, depending on the tree order. This process is repeated until all the sequences are aligned.

#### 1.2. PARALLEL-TCOFFEE (PTC)

PTC [3] is a parallel version of TC that allows to report alignments of more than hundreds of sequences, which is far beyond the capability of the sequential version. PTC is implemented on version 3.79 of TC and supports most of the options provided by this. The implementation of PTC uses a distributed master-worker architecture and a message passing paradigm employing one-sided communication primitives [4]. Basically, PTC parallelizes the library generation, the progressive alignment, which are the two main and difficult stages of TC, and the distance matrix computation.

- \* Distance Matrix Computation: In TC, the progressive alignment strategy is guided by a neighbor-joining tree. This is generated using some measure of sequence similarity expressed with a distance matrix. The computation of the distance matrix requires  $\binom{n}{2}$  sequence comparisons and each comparison is a totally independent task. This is why PTC parallelizes it by a master-worker paradigm implementing a Guided Self Scheduling method (GSS) [5] to distribute the computations (tasks) between workers. Each worker computes its part of the distance matrix, calculates the time required and returns the results to the master.

- \* Library Generation: This stage is the most time and memory consuming, limiting its applicability to no more than 100 sequences on a typical workstation. The library generation consists in three phases: (a)

Generation of all pair-wise constraints, where half of the total number of pair-wise alignments is distributed proportionally based on worker efficiency, and the other part is distributed using GSS. (b) Deletion, association and re-weighting of duplicated pair-wise constraints, where each host merges its duplicate constraints locally. And, (c) Transformation of the library into a three-dimensional look-up table, where the rows are indexed by sequences and the columns by residues. PTC implements the table using one-sided Remote Memory Access mechanisms (RMA) distributed on the workers, that also implements a cache system managed by a Last Recently Used policy to store all the frequent requests to remote memory.

\*Progressive Alignment: The computations of the progressive alignment stage follow a tree order, and its parallelization can be reduced to a Directed Acyclic Graph (DAG) scheduling problem. This is why this stage is the most difficult to parallelize. PTC implements a strategy similar to the HLFET (Highest Level First with Estimated Times) algorithms [6]. It launches the graph nodes that have no precedence dependencies and allow the earliest start time, until all graph nodes (alignments) are computed.

### 1.3. BALANCED PARALLEL-TCOFFEE

To improve PTC, we analyzed the scalability of it by running some tests and varying the numbers of processors between 16 to 120 processors. In this study, the Pfam data set PF000231 was used [7].

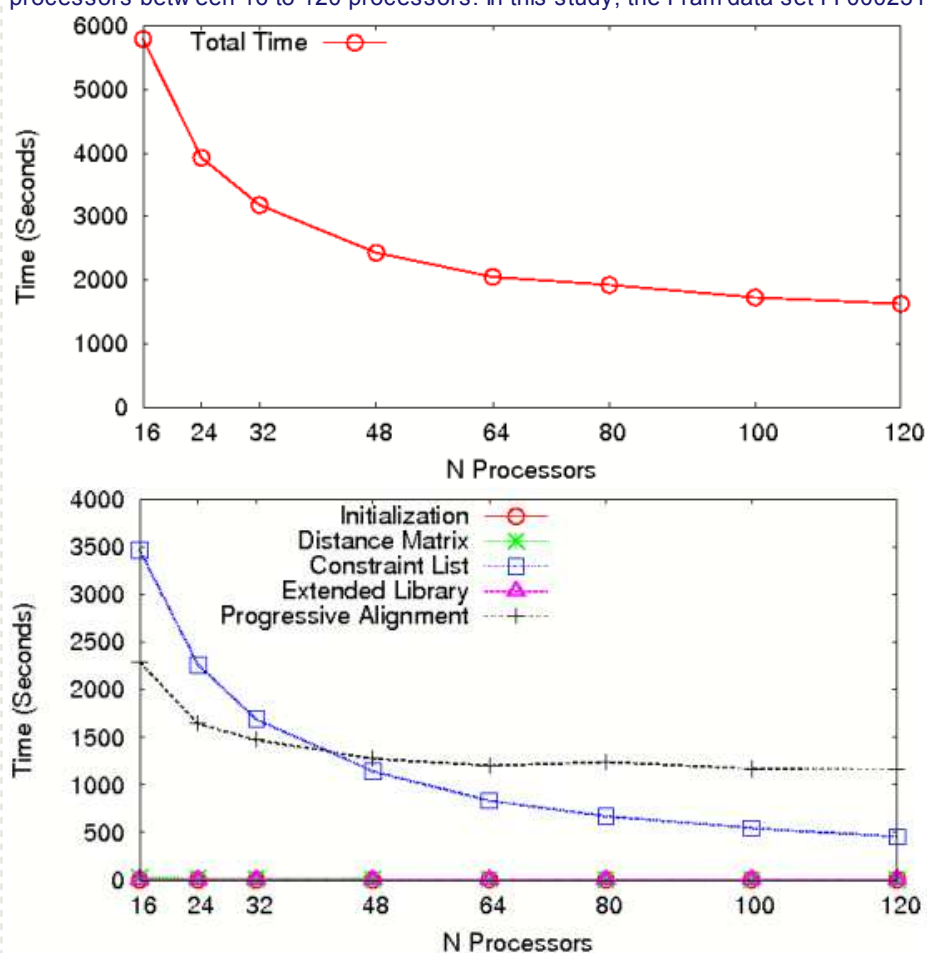


Figure 1: Parallel-TCoffee performance analysis. (a) Total execution time. (b) Stages execution time.

Figure 1a shows that PTC generally improves the execution time as the number of processors increases. It can be observed in Figure 1b, that the only stage that is improved is the library generation while the progressive alignment stage keeps linear on the number of processors, avoiding the scalability. Given these results it can be deduced that the optimization must be focused on the progressive alignment. This stage is driven by the guide tree because it determines the order of the partial alignments and defines the dependences among parallel tasks.

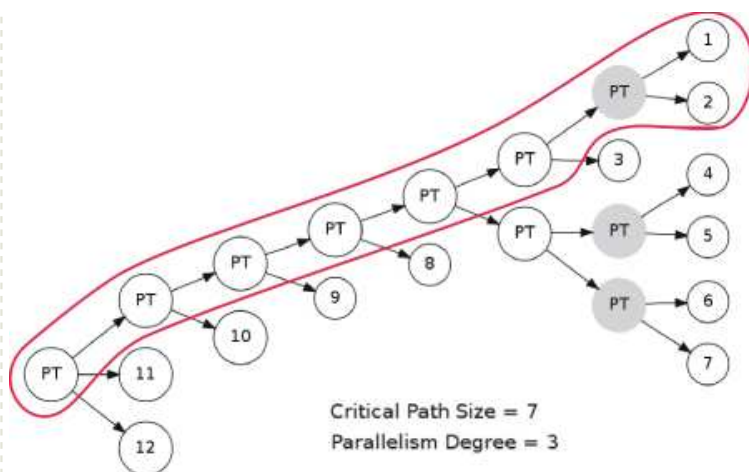


Figure 2: Guide tree generated with TC heuristic.

To analyze the characteristics of the guide trees generated by PTC, Figure 2 shows an example of them. In the tree, the PT-nodes (internal nodes) define the progressive alignment tasks. The leaf nodes are the different sequences to align (12 in this example) and the tree represents the order in which such progressive partial alignments can be performed.

From the point of view of parallelism, only PT-nodes with all dependencies resolved (all children nodes are leaves) can be executed as independent tasks. In the example, there are three initial tasks, grey PT-nodes, which can be executed in parallel, defining the maximum parallelism degree. Another important parameter is the critical path -the longest path through the guide tree to obtain the final alignment-. The critical path defines the number of sequential iterations that the alignment algorithm has to perform. As more sequential iterations, lower parallelism, lower performance and high execution times. In the example, the maximum degree of parallelism is three parallel tasks and the length of the critical path is 7 iterations.

In general, the guide tree generated by the PTC method is unbalanced and it generates too many precedence relations between the tree nodes and then generates longer critical paths. Unbalanced trees also affects to the degree of parallelism. The more unbalanced a tree is, the fewer the tasks can be launched in parallel and in consequence the lower degree of parallelism is.

### 1.3.1. BGT: Balancing Guide Tree

To solve the problems presented above, the proposal is to modify the tree generation method to take into account not only the similarity between sequences, but also balancing features. Our main goal is to reduce the number of precedence relations, reducing the critical path and increasing the degree of parallelism without losing the alignment accuracy.

The new heuristic called BGT tries to join the maximum number of pairs of sequences and locate them at the base of the tree in order to reduce the number of tree levels and thus reduce the critical path. In order to maintain the accuracy of the T-Coffee algorithm, our balancing heuristic is only applied if two sequences are quite similar. The main heuristic steps are:

\*Calculate the similarity threshold: This is the average value between the similarity values of the upper diagonal in the distance matrix (Figure 3 step 1). It is used to decide which method to use to group a pair of sequences: BGT heuristic or the original PTC heuristic.

\*Search and group the nearest pair of sequences: Like the original method, the nearest pair of sequences is grouped. The main difference is that if the intersection value between the pair of sequences in the distance matrix satisfies the constraint imposed by the similarity threshold, this pair of sequences is grouped and their respective columns and rows in the distance matrix are deleted (Figure 3 step 2). If not, they are grouped using the original PTC heuristic, where only one column and row is deleted and the other is filled with a new similarity value that represents the joining between this pair of sequences.

\* Replace pair by joining similarity: At each iteration the joining similarity values that represent the nodes that links the pair of sequences are calculated. These similarity values will be replaced when one of this conditions are achieved: 1) all the matrix has been deleted or, 2) the intersection value between the pair of sequences fails to satisfy the constraint imposed by the similarity threshold.



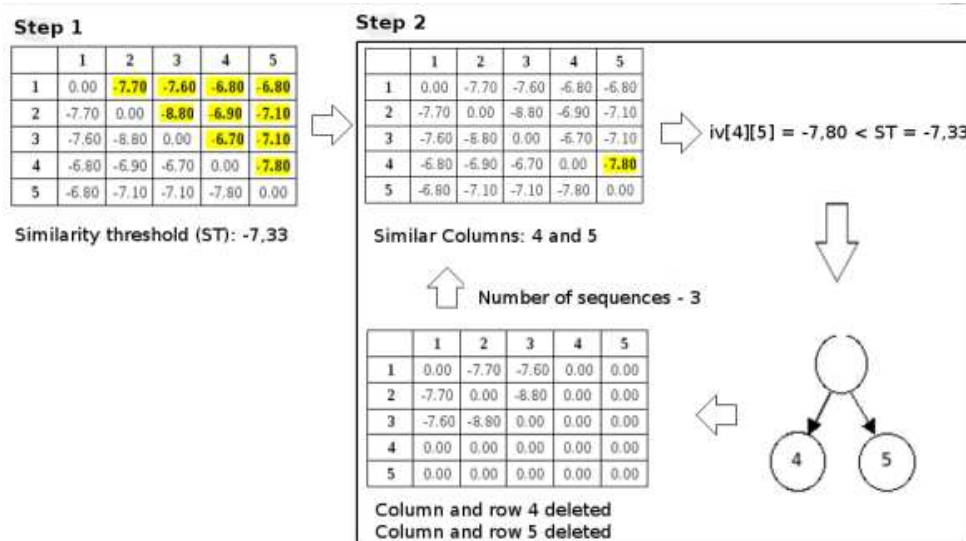


Figure 3: Guide tree generated with BGT heuristic.

Figure 4 shows the same tree as Figure 2, but in this case it has been generated with BGT. This tree is more balanced than the previous one. Thus, comparing it with Figure 2 tree, it can be noticed that the critical path is shorter and the parallelism degree is greater. The critical path length is reduced by 50%, from 7 to 4 iterations, and the parallelism degree is increased from 3 to 5 tasks, 66% greater.

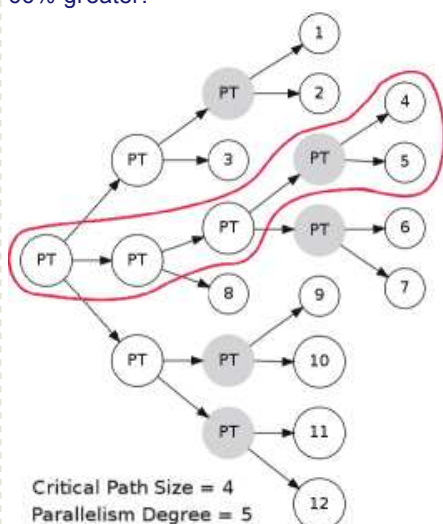


Figure 4: Guide tree generated with BGT heuristic.

A tree balancing study, shown in Table 1, was done using the sets of sequences PF00074, PF00200, PF00231, PF00349, PF01057 and PF08443 from the Pfam database. In this study, the critical path and the parallelism degree of each tree are compared between PTC heuristic and BGT heuristic.

SEQ	PTC Critical Path length	BGT	PTC Maximum Parallel Degree	BGT
PF00074	28	11	107	172
PF00200	29	17	173	295
PF00231	26	18	164	270
PF00349	23	19	143	252
PF01057	94	14	71	187
PF08443	68	29	215	365

Table 1: Critical Path and Parallel Degree study for Pfam.

The results in Table 1 show that BGT heuristic generates the guide trees with the shortest critical path and the greater degree of parallelism. On average BGT reduces the critical path by 59.71% and increases the degree of parallelism by 76.52% compared with PTC. The limit case appears with the data set PF01057 where BGT reduces the critical path by 85.11% and increases the degree of

parallelism by 163.38% against PTC. In conclusion, the guide trees generated with the BGT approach are more balanced than the guide trees obtained with the Parallel-TCoffee.

2. CASE FOR HPC-EUROPA2 FUNDING (OBJECTIVES, JUSTIFICATION & VISIT EXPECTATIONS)  
I will try to cover two main objectives during the stay.

### 2.1. PERFORMING A COMPREHENSIVE STUDY OF THE SCALABILITY OF THE PARALLEL VERSION OF THE APPLICATION TCOFFEE.

It is necessary to scale the number of alignments performed in our cluster and thus it must be tested in bigger cluster. This study will analyze the performance scalability of the application by scaling the problem (align a larger number of sequences or that they have a greater length). Some previous studies of the underlying algorithm TCoffee have shown that two main factors that limit its scalability are the data dependencies imposed by the guide tree that defines the order in perfect alignment between sequences and the huge Memory requirements of the extended library to maintain consistency in the alignment. These two parameters have a quadratic cost in the algorithm ( $O(N^2L^2)$ ), depending on the number of sequences (N) and size (L), which greatly difficult its scalability. In this framework, we plan to carry out a set of performance experiments, depending on the size of the problem, the number of processors and Memory available. In each experiment we will test the original version of the algorithm TCoffee and some alternative proposals for improving the efficiency of the parallel version (BGT-Coffee). From these experiments, the current limits of scalability of the application under study will be analyzed and new proposals in the algorithm to improve its efficiency and scalability will be performed.

### 2.2. TO PROVIDE EASY ACCESS TO THE BIOTECHNOLOGY COMMUNITY TO AVAILABLE HPC RESOURCES TO CONDUCT LARGE-SCALE ALIGNMENTS BY MEANS OF A GUI ALIGNMENT TOOL.

Nowadays, researchers in the field of genomics increasingly require greater computational resources to perform their experiments.

Until recently, to align few tens of sequences it was enough a desktop computer, which in few hours of work could obtain a satisfactory result.

But as the problems grows, it is more evident the need for an alignment tool for large-scale, enabling the processing of thousands or even tens of thousands of sequences. The computational requirements of these problems (months and terabytes of computer time and memory respectively) requires HPC systems with thousands of processors. However, the difficulty of using HPC platforms have limited the use of these resources in the field of biotechnology.

For this reason, we propose as part of this project to develop a portal or web application allowing access in an easy, efficient and controlled alignment tool (Parallel TCoffee) to the computing resources needed (Cluster or Grid) to perform large-scale alignments. In order to achieve this objective, we will use a specific tool called Rapid [8]. This tool is a cost-effective and efficient way of designing and delivering portal interfaces to applications that require remote compute resources. The aim of Rapid is to make completing these tasks as simple as ordering a DVD or booking a flight on the web. This customised interfaces allow tasks to be performed without referring to terminology about the underlying computational infrastructure. Moreover, the system allows to expose particular features of applications as not to overwhelm the user.

### 3. REFERENCES

- [1] C. Notredame, D.G. Higgins, J. Heringa. T-Coffee: A Novel Method for Fast, Accurate Multiple Sequence Alignment, *J.Mol. Biol.*, 302(1) (2000) 205--217.
- [2] C. Notredame, L. Holm, D.G. Higgins. COFFEE: an objective function for multiple sequence alignments. *Bioinformatics*, 14(5)(1998) 407--422.
- [3] J. Zola, X. Yang, S. Rospondek and S. Aluru. Parallel T-Coffee: A Parallel Multiple Sequence Aligner. In *Proc. of ISCA PDCS-2007*, (2007) 248-253.
- [4] MPI-2: Message-Passing Interface Standard version 2. <http://www.open-mpi.org>
- [5] C. Polychronopoulos and D. Kuck. Guided self-scheduling: A practical scheduling scheme for parallel supercomputers. *IEEE Trans. on Computers*, 36(12) (1997) 1425-1439.
- [6] Y.K. Kwok and I. Ahmad. Benchmarking, comparison of the task graph scheduling algorithms. *J. Par. Dist. Comp.*, 59(3) (1999) 381--422.
- [7] A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Ewinger, S.R. Eddy, S. Griffiths-Jones, K.L. Howe, M. Marshall and E.L.L. Sonnhammer. The Pfam protein families database. *Nucl. Acids Res.*, vol.30, no.1, pp.276-280 (2002).
- [8] Rapid: Giving Computational Science a Friendly Face. <http://research.nesc.ac.uk/rapid/>

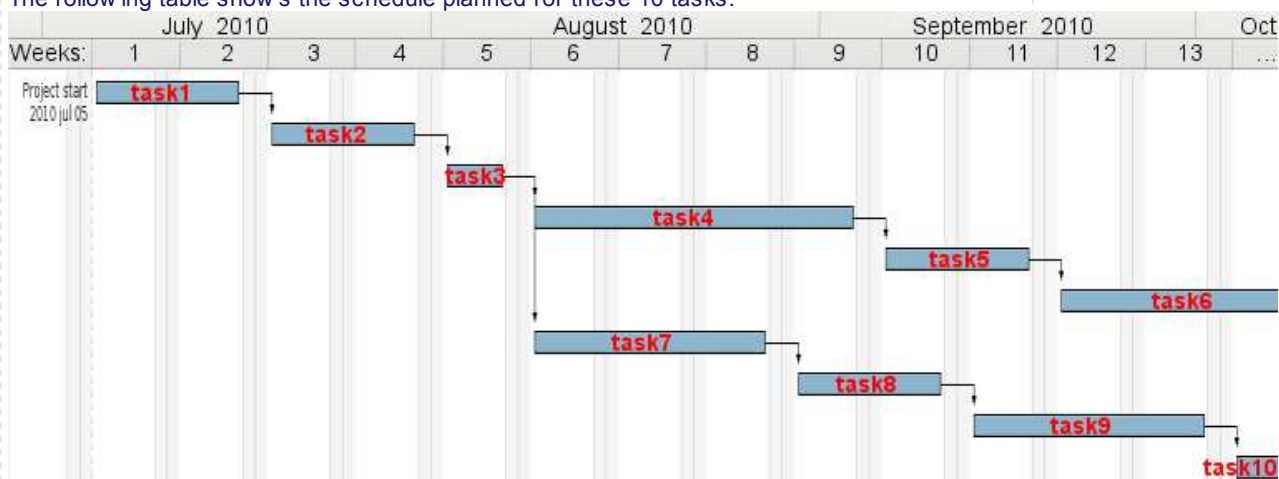
## Project Workplan



In order to achieve the main objectives defined on the project proposal, my project workplan is divided into the following tasks:

- Task 1: Modelling Parallel-TCoffee resource requirements (cpu and memory) with regard to number and size of sequences.
- Task 2: Evaluation and analysis of the scalability requirements of T-Coffee application for large-scale alignments.
- Task 3: Design and planning of the experimentation tests.
- Task 4: Evaluation of the proposed improvements for Parallel-TCoffee (This task includes repeat some tests if necessary).
- Task 5: Extensive analysis of the results and synthesize major conclusions.
- Task 6: Results dissemination.
- Task 7: Design and Development of Web Framework to access Parallel TCoffee on Grid.
- Task 8: Test of Parallel-TCoffee on Web Framework.
- Task 9: Integration and validation of Parallel-TCoffee resource requirements on the web Framework.
- Task 10: Results dissemination.

The following table shows the schedule planned for these 10 tasks:



## Section 7: Marketing HPC-Europa2

### About HPC-EUROPA2



**Where did you hear about HPC-EUROPA2?**

Previous visitor

If possible, please provide us with more details. For example, if you heard about us from a direct boss or colleague, how did they hear about us? If you found us from a link on another web site, tell us the URL of that site. If you selected "Other", please let us know where!

I don't know how Dr. Javier Fernández Muñoz (from University Carlos III of Madrid) heard about you, but he strongly recommended me this program.

### **Any further suggestions for a more effective marketing of the HPC-Europa2 Programme**



### **Data protection and privacy of personal information**

The collection of personal data is conducted in accordance with Italian laws and regulations. Such data will only be used for purposes connected to the fulfilment of the contract/service. Any information provided to CINECA during the supply/service will be treated as strictly confidential and in accordance to the terms of law.

As the form is closed you automatically authorise CINECA to use all your personal data for the selection procedure of the HPC-EUROPA2 Project, and for any further utilisation in the frame of the project, (according to the D.lgs. 196/2003 of 30/06/2003 about "Personal data protection").