



# *Proteomics applications in grid*

**Ivan Merelli**

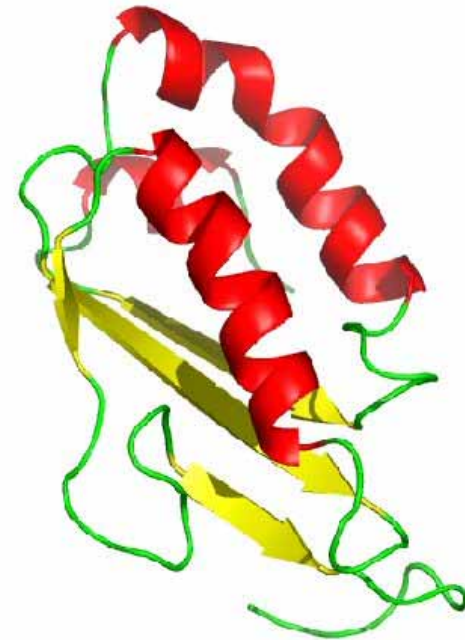
Institute for Biomedical Technology

National Research Council



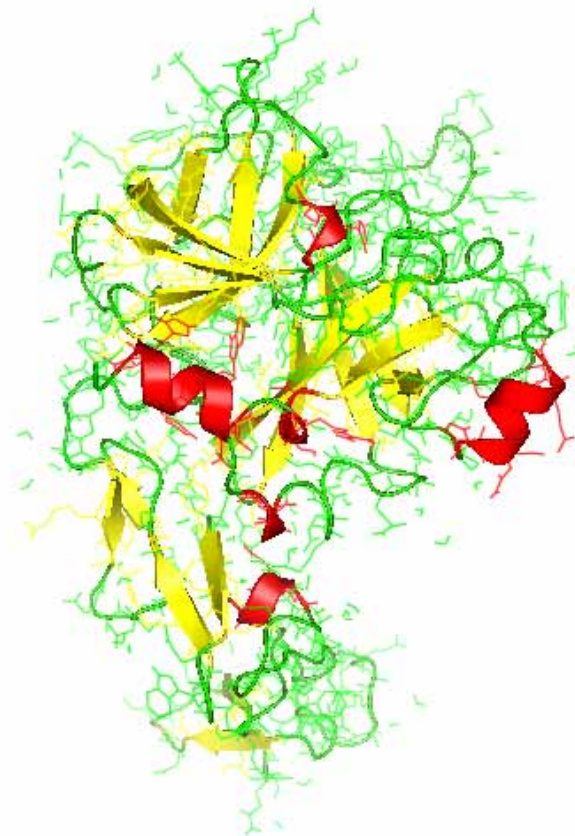


- Proteomics Work Package
- Sequence based analysis
  - Analysis software
  - Distribute approach
  - Relational job control
  - Database management
- Structural studies
  - Post Processing
- Web Interface
- Conclusions
- Acknowledgement





- The main task of the proteomics Work Package is the evaluation of different programs and databases to perform high throughput proteomics analysis in grid to face genome scale analysis.
- This Work Package is interested both in sequence based functional identification and in structural studies related to the surface atoms configuration.





- The objectives of the proteomics Work Package are:
  - an evaluation of the computational load needed by the most used proteomics software
  - a study on the possible strategy for porting on grid time consuming application
  - an analysis of the grid scalability for bioinformatics application.
- In order to accomplish this task the creation of a suitable infrastructure to perform programs and databases in grid plays a crucial role.





- The main tasks of the sequence-based proteomics are the discovery of the protein functionality and the identification of the key role residues.
- The first approach when dealing with a new protein sequence is to perform a similarity search in order to compare the sequence against protein and nucleotide sequence database.

Sequences producing significant alignments: (bits) Score E Value

pdb 1F7S A Chain A, Crystal Structure Of Adf1 From Arabidopsis T...	51	2e-07
pdb 1AHQ  Recombinant Actophorin	47	3e-06
pdb 1CNU A Chain A, Phosphorylated Actophorin From Acanthamoeba P...	47	5e-06

>pdb|1F7S|A Chain A, Crystal Structure Of Adf1 From Arabidopsis Thaliana  
Length = 139

Score = 51.2 bits (121), Expect = 2e-07

Identities = 33/130 (25%), Positives = 65/130 (50%), Gaps = 5/130 (3%)

Query: 5 TGIQASEDVKEIFARA---RNGKYRLLKISIENEQLVIGSYSQPSDSWDKDYDSFVLPLL 61  
+G+ +D K F R +++ KI ++Q+V+ QP ++++ + LP

Sbjct: 6 SGMVHDDCKLRFLELKAKRTHRFRIVYKIEEKQKQVVVEKVGQPIQTYEEF--ACL PAD 63

Query: 62 EDKQPCYILFRLDSQNAQGYEWIFIAWSPDHSVHRQKMLYAATRATLKKEFGGGHIKDEV 121  
E+ Y +++N Q + FIAW PD + VR KM+YA+++ K+E G +++

Sbjct: 64 ECRYAIYDFDFVTAENCQKSKIFFIAWCPDIKVRSKMIYASSKDRFKRELDGIQVELQA 123

Query: 122 FGTVKEDVSL 131  
+ D+ +

Sbjct: 124 TDPTEMDLDV 133

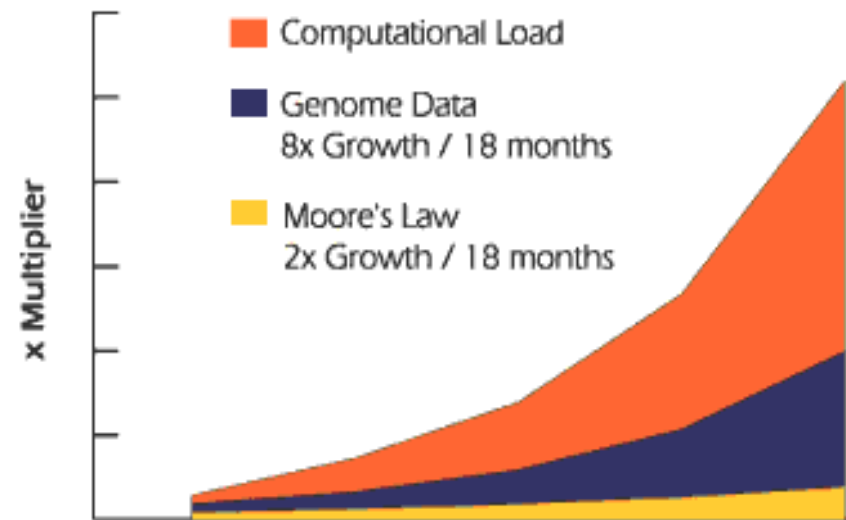


- Moreover, motif and profile-based search tools like PROSITE and PFAM can be used to perform functional annotation of protein sequences.
- In fact, protein domain patterns and Hidden Markov Model profiles are more specific for correlating protein structure and functionality
- For this reason these tools are assuming high importance in the analysis of the macromolecular functionality.





- The computational load needed for the protein analysis applications is quite important:
  - To perform an HMM analysis with hmmer against the Pfam database, for example, we need 0.3 sec for each amino acid
  - To perform a local alignment with blast against the NR database we need nearly 0.8 sec for each character.

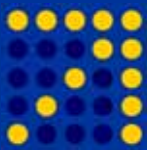




- BLAST (Basic Local Alignment Search Tool) is the most used software for the sequence-based analysis.
  - It is clear that the possibility to perform this application in grid is crucial for the functional annotation of protein sequences
- BLAST is the heuristic search algorithm based on a statistical methods employed by different programs in relation to the type of sequence presented as input ad reference:
  - **blastp**: compares an amino acid query sequence against a protein sequence database
  - **blastn**: compares a nucleotide query sequence against a nucleotide sequence database
  - **blastx**: compares a nucleotide query sequence translated in all 6 reading frames (3 on each strand) against a protein sequence database
  - **tblastn**: compares an amino acid query sequence against a nucleotide sequence database translated in all 6 reading frames.



- Using a number of signatures databases and their associated scanning tools the computation of a protein domain analysis can be very time-consuming:
  - **BlastProDom** is a wrapper script on top of Blast used to search against PRODOM database of protein domain families obtained by automated analysis of the SWISS-PROT and TrEMBL protein sequences
  - **FingerPrintScan** is used to search against the PRINTS collection of protein signatures which is very useful to detect similarities in highly divergent protein super-families
  - **HMMPiR** is a script based on hmmer that performs a wide range analysis on PIR SuperFamily, a classification database based on evolutionary relationship
  - **HMMPfam** is a script based on hmmer used to search against the Pfam database that contains curated multiple sequence alignments for each family and the corresponding hidden Markov models (HMMs)
  - **HMMSmart** is a script based on hmmer for the identification of genetically mobile domains and for the analysis of their architectures against the SMART database
  - **TIGRfam** is a script based on hmmer that implements a full alignment against TIGRFAM, a collection of protein families curated multiple sequence alignments, Hidden Markov Models (HMMs) and associated information designed to support the automated functional identification of proteins by sequence homology.



- **ProfileScan** is used to search against the PROSITE profiles database, a set of position-specific table of amino acid weights and gap costs, to identify protein family with very divergent sequences
- **ScanRegExp** is used to search against the PROSITE patterns collection of regular expression and verify the matches by statistically significant confirm patterns
- **Superfamily** is a script based on hmmer used to search against the SUPERFAMILY library of Hidden Markov Models that represent all the proteins of known structure, based on SCOP
- **SignalPHMM** performs prediction of signal peptide cleavage sites, using HMM.
- **TMHMM** is used to predict the transmembrane helices in proteins using HMM.
- **PANTHER** is a software which queries a unique resource that classifies genes by their functions, using published scientific experimental evidence and evolutionary relationships to predict their function
- **Seg** is used to identify and mask some low compositional complexity segments in amino acid sequences
- **Coil** is used to predict coiled-coil regions, using the Lupas algorithm.



- The most effective way to perform these sequence based analyses on grid technology is to subdivide the task into a set of small jobs.
- So it is possible to obtain a set of independent computations that can be addressed with a pure data parallel approach.
  - This means that subdividing the main task in a set of small jobs it is possible to obtain a set of independent computations
  - In this way the performance of the grid platform can be fully exploited and the distributed approach becomes an efficient solution.



- The easiest and most effective way to use the protein analysis tools on the grid platform is to subdivide the query file into a series of small *multifasta* files containing a balanced number of sequences.
- The environment we set up to perform the protein domain analysis in grid consists in creating an efficient system to coordinate the jobs submission, to check the computation status and to collect the results.

```
>gi|90110031|sp|Q15349|KS6A2_HUMAN Ribosomal protein S6 kinase alpha-2
MDLSMKKFVRRFFSVYLRRKSRSKSSLSRLEEVEGKVEIDISHHVKEGFEKADPSQFELLKVLGQGSY
GKVFLVRKVKGSDAGQLYAMKVLKATLKVDRVRSKMERDILAEVNHFFIVKLHYAFQTEGKLYLILDF
LRGGDLFTRLSKEVMFTEEDVKFYLAELALALDHLHSLGIIYRDLKPENILLDEEGHIKITDFGLSKEAI
DHDKRAYSF CGTIEYMAPEVNVNRRGHTQSADWWSFGVLMFEMLTGSLPFQGKDRKETMALILKAKLGMPQ
FLSGEAQSLLRALFKRNP CNRLGAGIDGVEEIKRHPFFVTIDWNTLYRKEIKPPFKPAVGRPEDTFHFDP
EFTARTPTDSPGVPPSANAHHLFRGFSFVASSLIQEPSQQDLHKVPVHPVIVQQLHGNNIHFTDGYEIKED
IGVGSYSVCKRCVHKATDTEYAVKIIDKSKRDPSEEIEILLRYGQHPNIIITLKD VYDDGKFVYLMELMR
GGELLDRILRQRYFSEREASDVLCITITKTMDYLHSQGVVHRDLKPSNILYRDESGSPESIRVCDFGFAKQ
LRAGNLLMTPCYTANFVAPEVLKRQGYDAACDIWSL GILLYTMLAGFTPFANGPDDTPEEILARIGSGK
YALSGGNWDSISDAAKDVVSKMLHVDPHQRLTAMQVLKHPVVNREYLSPNQLSRQDVHLVKGAMAATYF
ALNRTPQAPRLEPVLSSNLAQRRGMKRLTSTRL
```

```
>gi|56749457|sp|Q15208|STK38_HUMAN Serine/threonine-protein kinase 38
MAMTGSTPCSSMSNHTKERVTMTKVTLNFYSNLIAQHEEREMRQKKLEKVMEEGLKDEEKRLRRSAHA
RKETEFLRLKRLRGLGLED FESLKVIGRAGFGEVRLVQKKDTGHVYAMKILRKADMLEKEQVGHIRAERDI
LVEADSLWVVKMFYSFQDKNLNLYLIMEFLPGDMMTLLMKKDTL TEEETQFYIAETVLAIDSIHQLGFIH
RDIKPDNLLLD SKGHVKLSDFGLCTGLKKAHRTEFYRNLNHSLSDFTFQNMNSKRKAETWKRNRRLQAF
STVGTDPDYIAPEVFMQTGYNKLCDWWSLGVIMYEMLIGYPPFCSETPQET YKKVMNWKETLTFPPEVPI S
EKAKDLILRFCEWEHRIGAPGVVEIKSNSFFEGVDWEHIRERPAASIEIKSIDDTSNFDEFPPESDILK
PTVATSNHPETDYKNKDWFVINYTYKRFEGLTARGAIPSYMKA AK
```



- For each split input file a JDL script is dynamically produced in order to describe the job making the RB, which manages the policies for job allocation, able to route it.
- Moreover, the JDL script specifies the application reference database to be copied on the CE where the application will be performed.

```
[  
  Executable = "hmmpfam.sh";  
  Arguments = "test.seq exit.seq";  
  StdOutput = "stdout";  
  Stderr = "stderr";  
  InputData = "lfn:/grid/biomed/Pfam";  
  DataAccessProtocol = "gsiftp";  
  InputSandbox =  
    {"hmmpfam", "hmmpfam.pl",  
     "hmmpfam.sh", "input.seq"};  
  OutputSandbox =  
    {"stdout", "stderr", "output.seq"};  
]
```



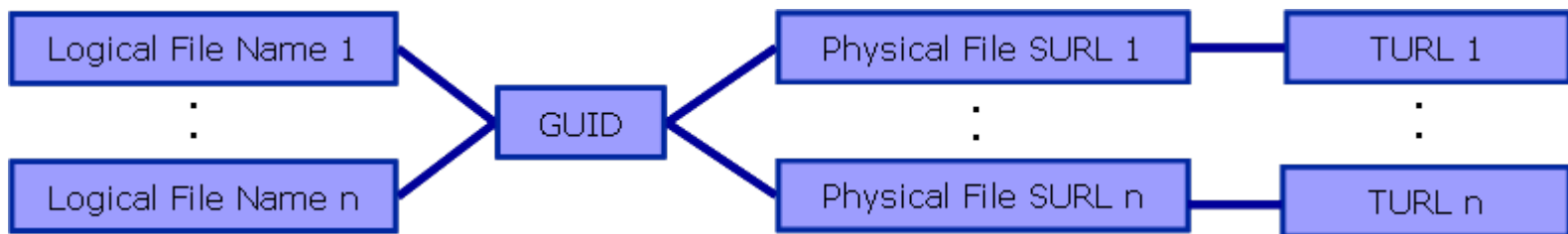
- All the JDL scripts are coordinated by an infrastructure that subdivides the input data and controls the advancement state of the computation, retrieving the output when the jobs are successfully finished or submitting them again in the case they are incorrectly completed.
- A crucial problem is how to control the submission and how to integrate the output data
  - For each analysis tool and database, a different distributed implementation has been chosen in order to maximize the job efficiency
  - This aspect is highly influenced, for example, by the size of the reference database.



- The developed infrastructure relies on a Relational Database for monitoring the jobs' execution:
  - This database is constantly updated with the computation status
  - On the top of the Database relies the software that manage each single job, masking the grid complexity to the users.
- The key information stored in the Relation Database for each job are:
  - the application to perform
  - the related execution parameters
  - the user input data
  - the reference database.



- An important challenge in porting on the grid platform the software for sequence alignment and domains identification, is the management of the database in grid.
- Our system works on the top of the EGEE grid middleware that implements a virtual distributed file system, that associate a number of database replicas to a single GUID.



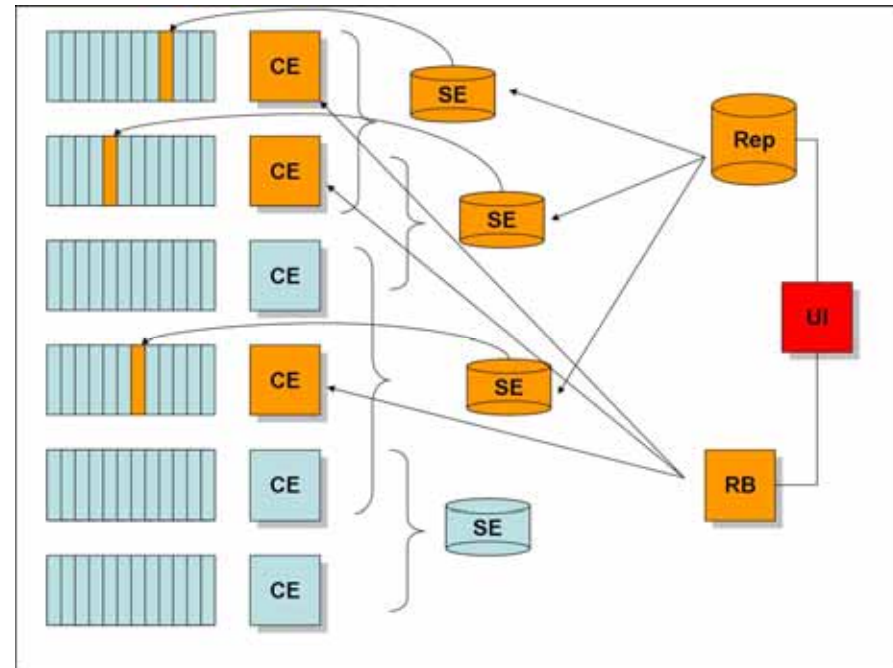


- The Automatic Updater (AU) constantly monitors FTP sites looking for newest versions of each databases to replace the older version on the grid
  - When a new timestamp on FTP sites is detected, the newest version is automatically downloaded and replaces the older version on the grid
  - Before clearing the older version, an xdelta patch is computed allowing to regenerate the old version starting from the new one.



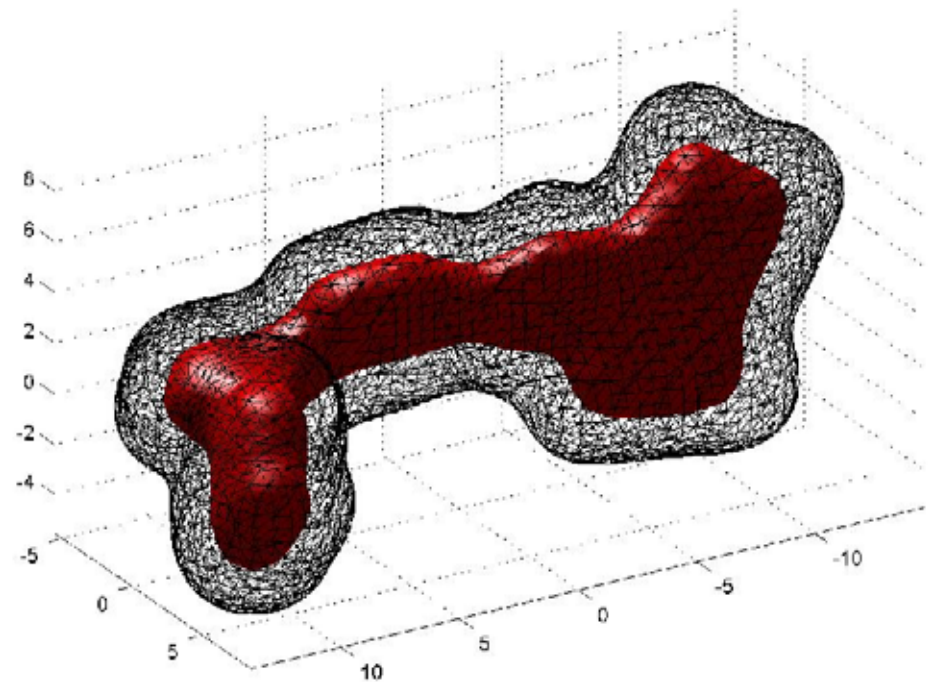


- Moreover, it allows the dynamic replication of each database in relation with its usage in order to balance the number of replicas taking into account the occupied disk space.
- This feature relies on the statistical analysis of the database usage, working on data acquired after each job execution such as queue times and overall job computation times.



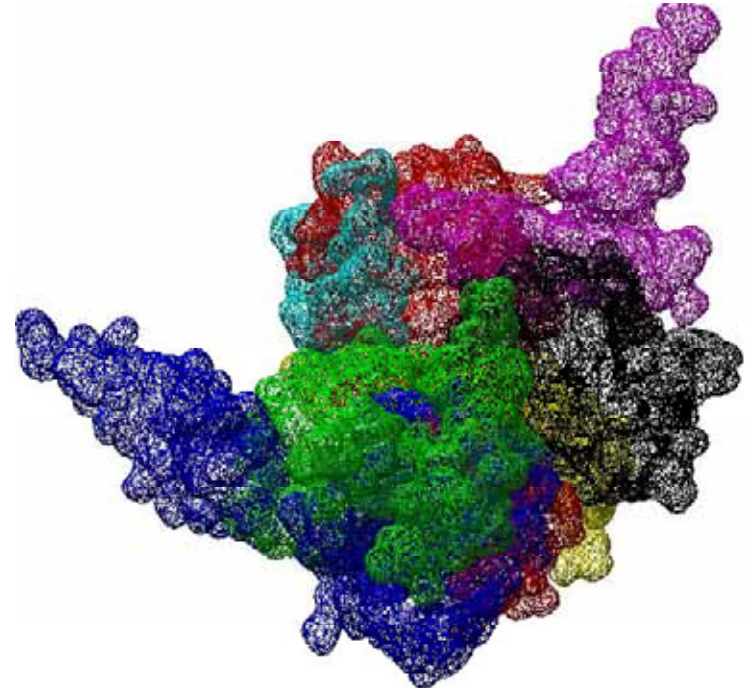


- The other main objective of the proteomics Wok Package is to address the protein functional analysis from a structural point of view.
- This kind of analysis is usually very time consuming because it relies on a pure geometrical approach.
- As for sequence based analysis, working at genome scale it is crucial to distribute the computational load on platform like grid.



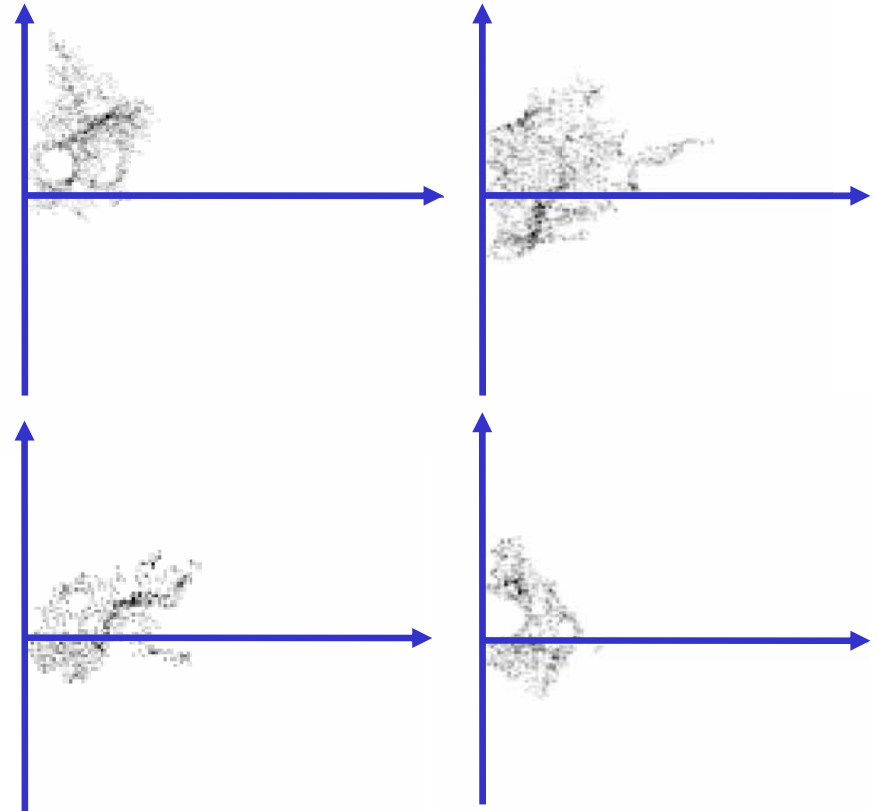


- In this project we want to test a novel approach to study the macromolecular functionality based on the analysis of protein surfaces.
- To establish possible functional correspondence among different proteins we must solve the problem of matching different surfaces.
- This task is quite complex because the description of very similar surfaces can be performed in different ways.



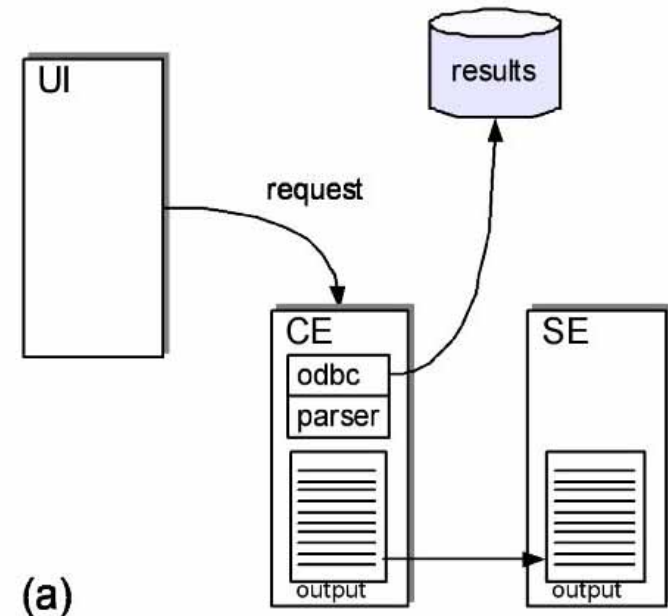


- To establish correspondence between different surfaces we will perform correlation relying on images of local description.
- Using this system we will identify surface similarities and complementariness that are very useful to describe protein functionality.
- The high number of correlations that have to be calculated represents a major issue: for this reason we will implement a grid version of this analysis system.



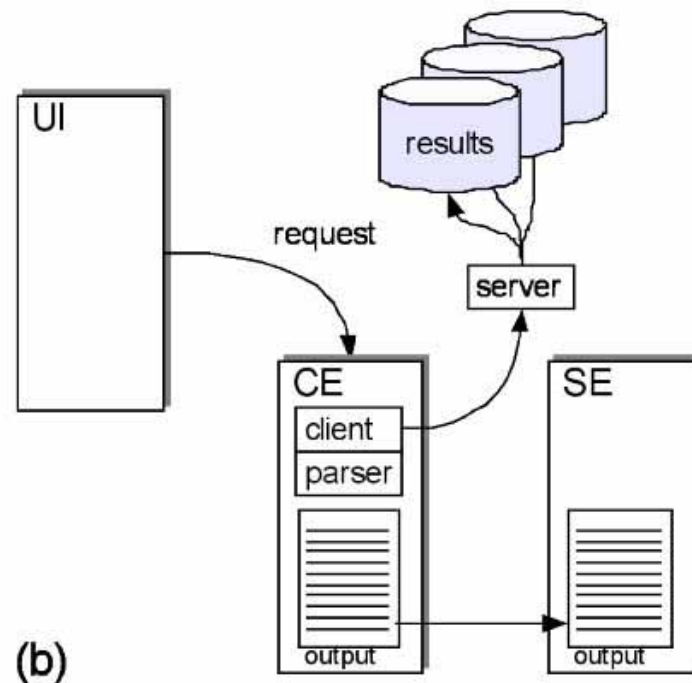


- While dealing with biological challenge in which huge quantity of data are involved the post processing of the output is a crucial problem.
- The post processing analysis of the results can be computationally very expensive if an adequate system to parse and collect the results has not been previously established.
- We face this problem both parsing the output results and storing them in an output database directly from the computational resources.



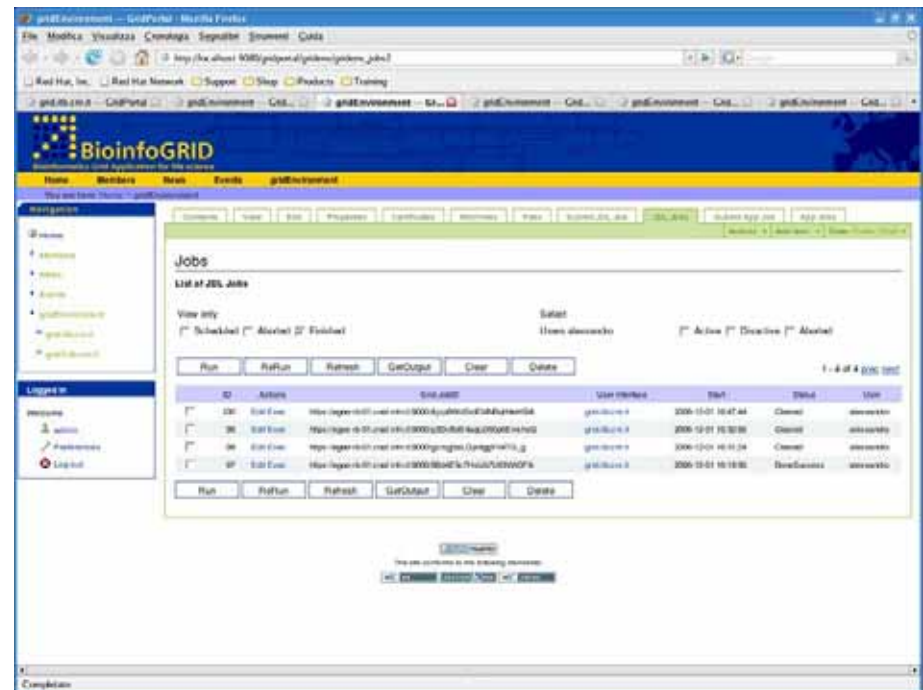


- Clearly the parsing system must be specifically designed for each application: the elaboration of the results immediately after the computation is ideal to fully exploit the grid resources and to overcome the post processing problem.
- A grid compliant solution can be implemented accessing the database using an authenticated grid service, in order to manage more efficiently different connection and different uploading section, using the GSI security infrastructure.



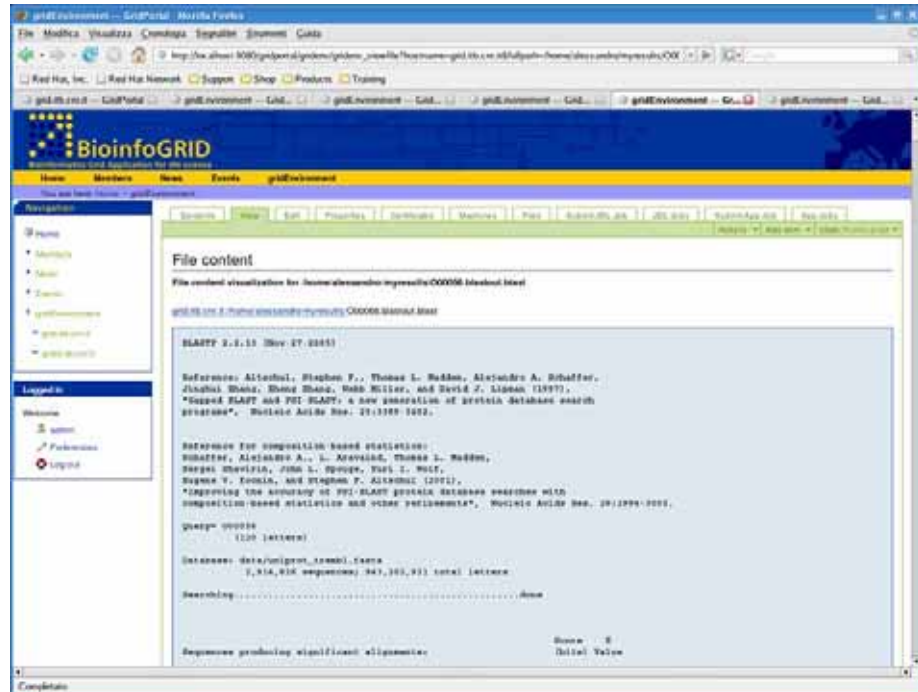


- In order to make possible the evaluation of this software for protein domain analysis a web interface has been developed.
- It is used to submit jobs to the grid infrastructure, to visualize in a clear form the obtained results and to hide the complexity of the distributed platform.



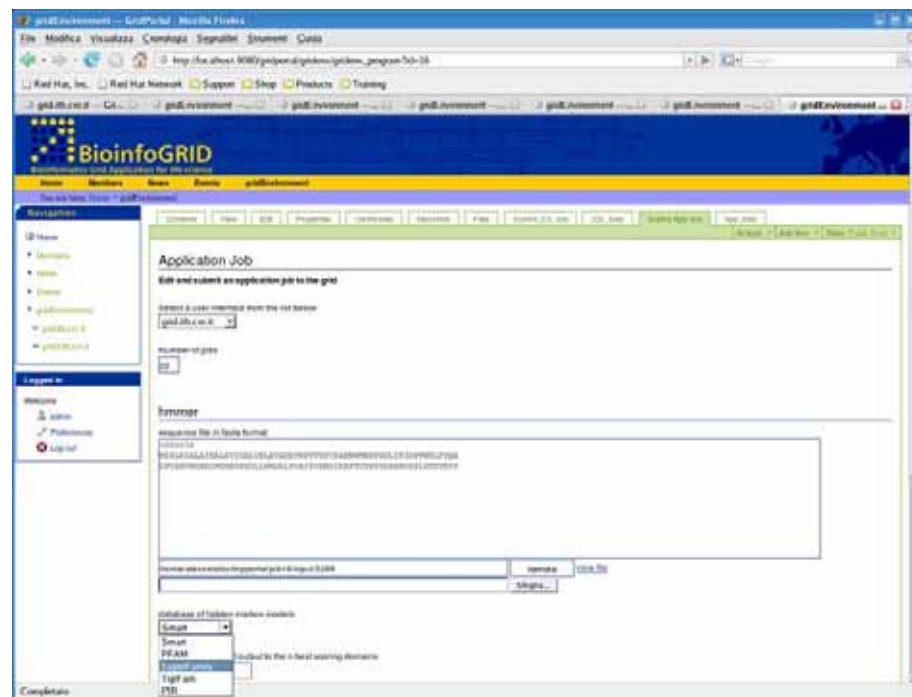


- The main feature of the portal is the possibility to hide completely the JDL scripts layer for the grid job submission.
  - While it is still possible to submit simple job to grid writing it's own JDL script, the idea is to hide this process to make the use of the grid user friendly for the bioinformatics community.
- Both at user level, for the input definition, and from the computational point of view, including the distribution policy, these applications are coordinated by the portal.





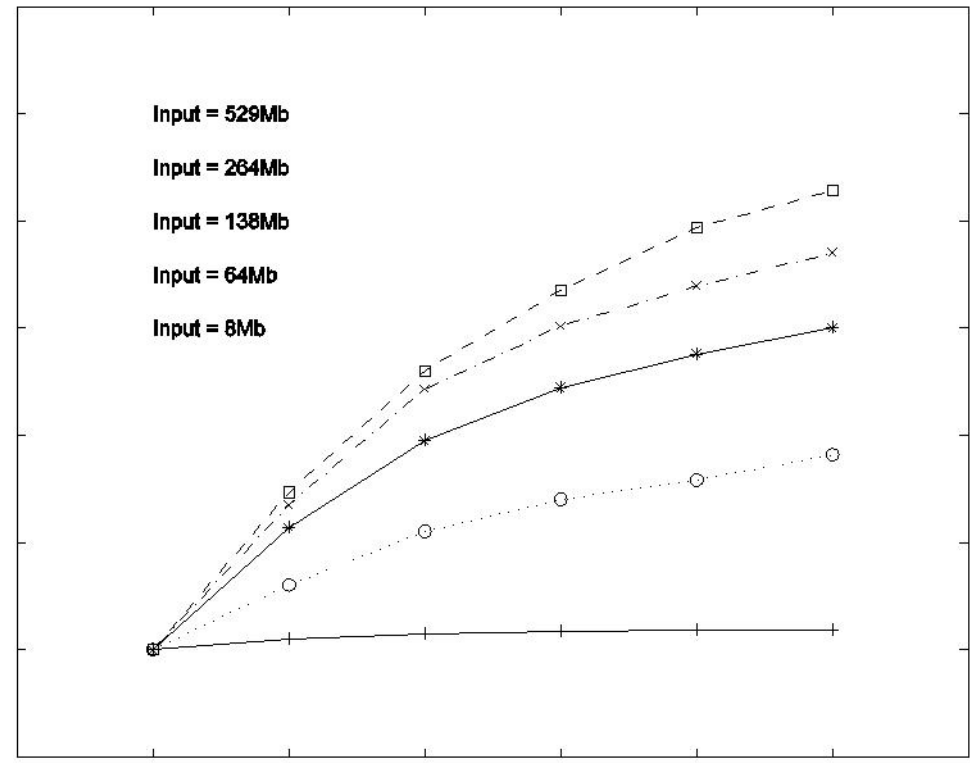
- This is possible by the definition of a couple of XML files that describe both the end user parameters and the application distribution policy.
- The file describing the web interface is converted to HTML, while the distribution policy define how to generate the JDL scripts effectively submitted to the grid.







- The main task of the second year of the project is to complete a report on the performance and the scalability of Bioinformatics software on the grid platform.
- All the tests performed until now have been studied for design solutions oriented to fully exploiting the grid's resources.





- In conclusion this study presents a number of applications for protein functional analysis and most of them are computationally very expensive.
- It is clear that working at genome scale a high throughput grid implementation for these applications is very useful to lower the execution time.
- Clearly, the performance of these computations largely depends on the available resources at the moment of submission.



BioinfoGRID

# Acknowledgments



- *BIOINFOGRID*  
<http://www.bioinfoGRID.eu>

- Alessandro Orro
- Gabriele Trombetti
- Chiara Bishop
- John Hatton
- Luciano Milanese



dkfz.

