



The discovery net system for high throughput bioinformatics

Anthony Rowe^{1,*}, Dimitrios Kalaitzopoulos^{1,2}, Michelle Osmond¹, Moustafa Ghanem¹ and Yike Guo¹

¹Department of Computing Imperial College, 180 Queens Gate, London, SW7 2RH, UK and ²The Wellcome Trust Sanger Institute, Hinxton, Cambs, CB10 1SA, UK

Received on January 6, 2003; accepted on February 20, 2003

ABSTRACT

Motivation: Bioinformatics requires Grid technologies and protocols to build high performance applications without focusing on the low level detail of how the individual Grid components operate.

Result: The Discovery Net system is a middleware that allows service developers to integrate tools based on existing and emerging Grid standards such as web services. Once integrated, these tools can be used to compose reusable workflows using these services that can later be deployed as new services for others to use. Using the Discovery Net system and a range of different bioinformatics tools, we built a Grid based application for Genome Annotation. This includes workflows for automatic nucleotide annotation, annotation of predicted proteins and text analysis based on metabolic profiles and text analysis.

Contact: asr99@doc.ic.ac.uk

Keywords: grid, E-Science, annotation, workflow, pipeline.

INTRODUCTION

Current research into fundamental Grid technologies, such as Globus (Foster and Kesselman, 1997), has concentrated on the provision of protocols, services and tools for creating co-ordinated, transparent and secure globally accessible computational systems. These technologies follow a service methodology for finding both computation and data services for performing computationally or data intensive tasks. The delivery of the low-level infrastructure is essential but does not aid end users in the creation of applications that use all of the services the Grid has to offer.

The Discovery Net project (Curcin *et al.*, 2002) has developed from the need for a higher-level layer of informatics middleware to allow scientists to create meaningful data analysis processes and then execute them using an underlying Grid infrastructure without being aware of the

protocol used by individual services. The Discovery Net system builds on top of the fundamental Grid technologies to provide a bridge between the end user of a Grid service and the developers of individual Grid tools.

Using the various tools produced as part of Discovery Net, generating a reusable Grid application becomes the task of selecting the required components and services and connecting them into a process. This is based around an XML-based language Discovery Process Mark-up Language (DPML) (Syed *et al.*, 2002). A process created in DPML is reusable and can then be encapsulated and shared as a new service on the Grid for other scientists.

As part of the Discovery Net project we have developed a number of case studies based on bioinformatics applications including genome and protein annotation. These applications have been developed to provide working models of how Grid applications can be used and to develop an understanding of the requirements of distributed heterogeneous scientific applications working in Grid environments.

DISCOVERY NET ARCHITECTURE

The aim of the Discovery Net project is to provide middleware technology to allow users to create knowledge discovery applications that use Grid-based resources. In effect, the project aims to provide a bridge between a scientific community performing analysis, such as the molecular biology community and the high-performance computing community who create the underlying Grid services. The methodology behind the Discovery Net is the development of data flow processes or pipelines that represent the transformation of data from one form to another using a variety of different services.

The Discovery Net system is designed primarily to support analysis of scientific data based on a workflow or pipeline methodology. In this framework, services can be treated as black boxes with known input and output interfaces. Services are then connected together into a sequence of operations. Typical services include: database access, clustering, homology searching or notification.

*To whom correspondence should be addressed.

Services can be combined together into workflows that represent the application. The applications have a variety of different features. These include: the dynamic retrieval and construction of required datasets; the execution of data analysis algorithms on distributed computing servers; and the dynamic integration of new servers, databases and algorithms within the knowledge discovery process.

Based on the features of discovery pipelines, the Discovery Net architecture is designed around the requirements of three main components:

Data Requirements: Data used in different scientific experiments is highly heterogeneous. Data can include tabular data, textual data, image data and spectral data. The data can be stored in a range of different database formats and technologies. The Discovery Net system supports a wide range of data access and integration methods.

Execution Requirements: Data analysis workflows represent the composition of services that can be based in many locations. The execution of a workflow requires the Discovery Net system to coordinate and manage the passage of data to each of the different services, executes the service and handles the results correctly including passing of these results to the next service. Workflows are also heterogeneous as they can represent parallel and sequential operations.

Component Requirements: The Discovery Net is based on different software components, each with associated resource constraints that must be specified and satisfied, i.e. is the component bound to a specific resource, or is the component resource free and mobile? The design of the Discovery Net system, which is built to satisfy these requirements, must therefore be open and inclusive of many different types of data accesses and integration of different types of components. It must also support distributed execution over a Grid-based network and also consider the security of the data and resources being used in the execution of a discovery workflow.

The Discovery Net system is split into a number of core services; these services can be used via several APIs to allow the user to expand and modify the system with new data types, components, clients and compose and execute services based on a workflow methodology.

Figure 1 shows the core Discovery Net services.

Component Service: The Component service manages the integration of different components and Services into the system. The nature of the service is not constrained, so that a wide variety of different protocols can be used to integrate different services into the discovery net. These include HTTP, Web services using the SOAP protocol, and OGSA Grid services.

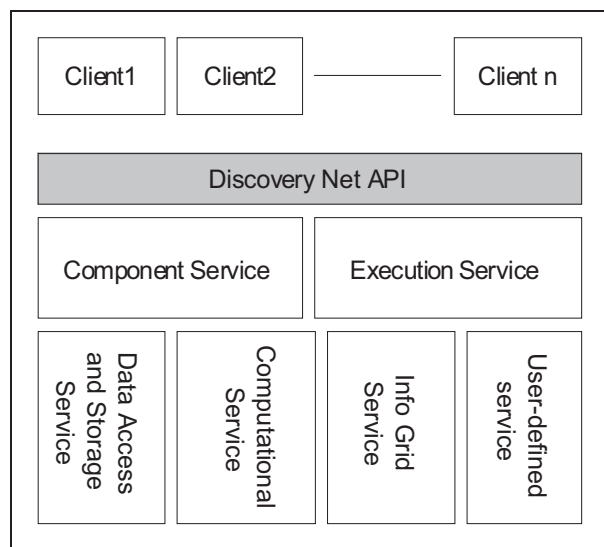


Fig. 1. The Core Discovery Services for handling integration of different components for computation and data access and an execution service for coordinating the execution of services.

Execution Service: The Execution Service manages the distributed execution of the jobs by analysing the DPML specification of the workflow, then matching the operations to the components that are available via the components service. Depending on whether a specific resource is bound to a specific component or the resource is a mobile component, the execution engine co-ordinates the scheduling of the job, passing it the correct input data and handling the output results.

Data Access and Storage Service: The Data Access and Storage service is a utility service designed to aid many of the common Data Access tasks rather than develop specific components for each methodology. The Storage Service allows data that has been accessed to be stored locally and provides storage for the representations of workflows that have been designed in the system. These can be accessed via this service or exported as reusable services into the component repository maintained by the component service.

Computational Service : The Computational Service is a utility service for integrating computational services directly into the Core Discovery Net system. These locally defined processes execute on the same resources as the core services. The heterogeneous format of Grid applications means that a local execution service can greatly enhance the performance of a specific job, if all of the required components can be made available locally.

InfoGrid Service: The InfoGrid Service provides another utility service for data access. The InfoGrid service provides a standard query interface for heterogeneous databases, such as those found in bioinformatics. The different databases are integrated into the InfoGrid component instead of being accessed as specific individual components. Access to the different databases can then be performed via this specific method. It also allows the dynamic creation of data sets from a wide variety of distributed heterogeneous platforms.

User Defined Service: The main aim of the Discovery Net platform is to make the system extensible so that the users of the systems can take advantage of new and improved services that become available. New services are added using standard interfaces provided by the Computational Service.

Discovery Net API: The Discovery Net API allows programmatic access to all of the DiscoveryNet services. It is used to write client software such as the Discovery Net Client, which allows the graphical construction of workflows from the individual components and services.

Discovery Net Clients: The Discovery Net clients provide users with graphical means of constructing their knowledge discovery workflows and providing access to data resources and result visualisation tools.

In operation, there are many Discovery Net nodes, each of them providing access to the core services and the different user-defined components. The different nodes can operate together. The different Discovery Net nodes can be combined to form a cluster of nodes, allowing users access to all of the services offered by the cluster of nodes. The combined operation of the Discovery Net nodes allows for other High Performance Computing capabilities such as automatic task farming of jobs across groups of nodes. The overall infrastructure provided by the Discovery Net is of a distributed set of nodes, each providing different services that can be accessed from any client.

COMPONENT INTEGRATION

The main structure that is available to a system integrator is a Discovery Net Component. This is integrated into the system by means of the previously mentioned Component Service.

The main feature of a component is a Service Descriptor. This provides a description of the input and output ports of the component, the type of data that can be passed to the component and parameters of the service that a user might want to change.

Once the service descriptor has been developed, the component can be added into the DiscoveryNet by registering with the component service. This will dynamically

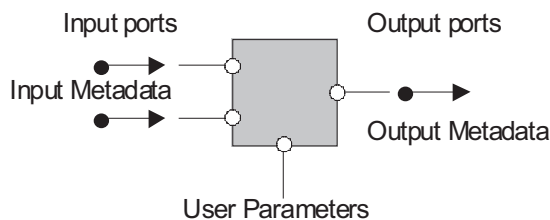


Fig. 2. The anatomy of a Discovery Net component. The component descriptor describes the inputs, output and parameters of the service. The descriptor provides a mechanism for describing its output by defining output metadata. The inputs ports can be constrained to only accept data of a specific type such as those provided by another component.

make the service available to clients so that users can take advantage of the new service.

An overview of the anatomy of a service descriptor is shown in Figure 2. The descriptor describes three main types of interface. The first is the Output port interface. This describes what will be output by the service at runtime. There are also the Input port interfaces to describe the required inputs to the service. These can be constrained to only accept connections from output ports that provide a specific metadata. This is important for describing valid discovery processes according to DPML and to help users who compose services together into a specific application make processes that make sense. The final interface is the parameters that the user is able to change to customise the service.

The DiscoveryNet infrastructure additionally provides the developers of components with the Data Access, Computational and InfoGrid services so that users can quickly build specialized services for common Grid based tasks from this existing infrastructure.

CASE STUDY: GENOME ANNOTATION

The Genome Annotation application was taken as a Discovery Net case study because it is a real world application that communities of researchers perform when a new organism is sequenced. The speed of DNA sequencing technologies is increasing, as is the number of genomes that are being published each year. However, the sequence alone provides no information about the biology of the organism. It is important to have a model of locations of the different region, including genes, regulatory elements, repeats and other interesting areas.

The main features of the Genome Annotation application that make it suitable as a Grid application are due initially to it being a highly data and computer-intensive application. There are also a large number of existing services for performing different annotations, which can be

integrated as components. Finally, annotation is not a completely automated task. The predictions produced by different software need to be evaluated by many different members of the community to curate and reach a consensus on the annotations that are being stored.

The initial steps to developing a Grid application such as this are to define the required services for the application and then to find if existing services can be used or the service needs to be developed. To be complete, we considered the three levels of Annotation as defined by Stein (2001): nucleotide-level, protein-level and process-level annotation. It was also required to consider visualisation, and storage of annotations. For each of these different tasks a range of individual tools was used which will now be summarised.

Nucleotide-level annotation: The nucleotide-level annotation is the first step in annotating a genome. All of the following publicly available tools were used in the Discovery Net case study as part of the nucleotide-level annotation workflow (Fig. 3). We chose to search for the following features.

Genes, which can be predicted by two methods: *ab initio* gene prediction and similarity-based gene prediction. The similarity of a genomic region to a sequence that has been deposited in a public database and is known to code for a protein is the most powerful way of predicting genes. The problem is that not all genes have homologues deposited in databases. Thus, a combination of the *ab initio* gene prediction methods and the similarity-based gene prediction methods provide a refined and highly accurate prediction. We chose to use a combination of the hidden Markov model (HMM) based Genscan (Burge and Karlin, 1997) for *ab initio* gene prediction and BLAST (Altschul et al., 1990) for a similarity method. Other tools that can assist in gene prediction are the EMBOSS tools cpgreport and getorf (Rice et al., 2000) that predict CpG islands, which are found in gene-rich regions, and open reading frames (ORF), regions of the genome that code for a protein, respectively.

Genetic markers, such as long restriction fragment length polymorphisms, can be identified by BLAST. Other important features of genomes are the non-coding RNAs. rRNAs and other non-coding RNAs can be identified by BLAST, while tRNAs can be predicted by tRNAScanSE (Lowe and Eddy, 1997). Regulatory regions, such as transcription-factor binding sites, can be detected by similarity searches that are specialised for short motifs. Promoter regions can be partly identified by Promoter 2.0 (Knudsen, 1999). A large eukaryotic genome contains repeated segments of DNA. Known repetitive elements, such as Alu repeats, are detected and masked using RepeatMasker (Smit and Green, unpublished). Once a family of repetitive elements is identified, other members

of that family are found by sequence similarity methods (BLAST).

Protein-Level Annotation: After the identification of genes, their encoded products have to be classified into protein families and be functionally characterised. A common workflow of protein annotation that was used in the case study is the following (Fig. 4). BLAST and PSI-BLAST (Altschul et al., 1997) are performed against the protein databases SWISS-PROT (Bairoch and Apweiler, 2000) and SWISS-PROT TrEMBL (O'Donovan et al., 2002) to identify homologues that are well characterised and of known function. Thus, functionality for the query protein can be inferred by homology to a protein of known function. Identification of functional domains is carried out by InterProScan (Apweiler et al., 2001), a tool that searches various protein domain and motif databases. Prediction of other domains, such as transmembrane helices and signal peptides (mitochondrial, chloroplastic, secretory, or other), is performed by HMMTOP (Tusnády and Simon, 2001) and TargetP (Emanuelsson et al., 2000), respectively. *Ab initio* predictions of protein function from sequence is performed by ProtFun 2.0 (Juhl Jensen et al., 2002).

Process-Level Annotation: The process of functional annotation uses the InfoGrid architecture to integrate Medline abstracts from PubMed via homologues of a specific protein. The process initially uses BLAST on a query protein against the SWISS-PROT database to find the similar known proteins, and then extracts the Medline abstract from PubMed. This is then used with a text analysis system for finding detail about metabolic processes. The text analysis engine is part of the core Discovery Net system.

Annotation Visualisation: To reduce the error rate of annotations it is required that researchers have the ability to view the annotation on the genetic sequence. Artemis (Rutherford et al., 2000) is a visualisation tool written in Java that is designed for viewing sequence features of entire bacterial genomes and small eukaryotic ones. Large eukaryotic genomes can also be visualised in smaller chromosomal fragments. Artemis was integrated into the Discovery Net client for this application

Annotation Storage: The distributed annotation system (DAS) is a client-server system in which a single client integrates information from multiple servers. It allows a single machine to gather up genome annotation data from multiple distant web sites, collate the information, and display it to the user in a single view. We used the LDAS system to store our annotations built using the pipeline. Access to the LDAS systems was via a Web service.

Having created a series of components and integrated

them into Discovery Net, users of the system can then compose the different tools into an application. Workflows for Nucleotide Annotation, Protein Annotation, and Process Annotation were built as shown in Figures 3, 4, 5 respectively.

IMPLEMENTATION AND EVALUATION

The complete case study was built by integrating the tools mentioned above for the different applications. Some components were integrated as Web Services, some as local processes, some as Grid services. The components, which were integrated without awareness of their execution environment or assumptions about the structure of the workflow, then execute the service, thus realising the aim of a high-level informatics environment. The creation of these services took less than three man-weeks.

To develop a reusable set of Discovery Net services a standard format for the input and output of each node was used. Each node for performing annotations was developed to use a FASTA format for the input data and input metadata for the different service descriptors. The output of each service was defined to be a table compatible with the DAS table structure. This required the reports of the different annotation methods to be converted into a tabular format. This process of normalizing the output format was encapsulated into each service.

With each service constrained to take input and output in the same defined format a generic set of services for bioinformatics was developed. These can be selected and composed into discovery processes such as those seen in Figures 3 and 4.

As well as the core services several utility components were developed such as a service to transform from the DAS table format to an EMBL feature format and a service to download FASTA files from a given Internet location.

The Discovery Net client was then used to combine the distributed tools together and was presented at the IEEE SC2002 Supercomputing conference in Baltimore. The annotation pipelines were running on a variety of distributed computing resources including: high performance resources hosted at the London E-Science centre, servers at Baltimore, and databases distributed around Europe and the USA.

The application was combined with a Real Time DNA sequencing platform provided by DeltaDot Ltd. (DeltaDot, 2003) to produce a real time genome sequence and annotation pipeline. This application was subsequently awarded the 'Most Innovative Data Intensive Application' at the conference's High Performance Computing Challenge.

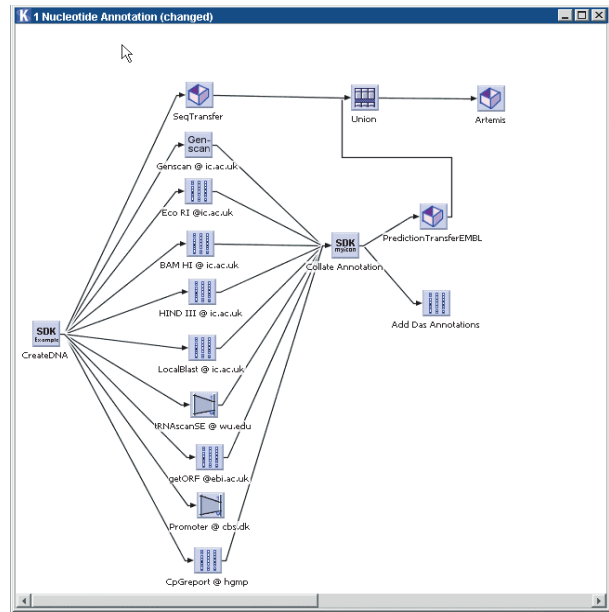


Fig. 3. Highly distributed Nucleotide annotation pipeline application automatically feeding into the Artemis visualisation and building a DAS annotation warehouse.

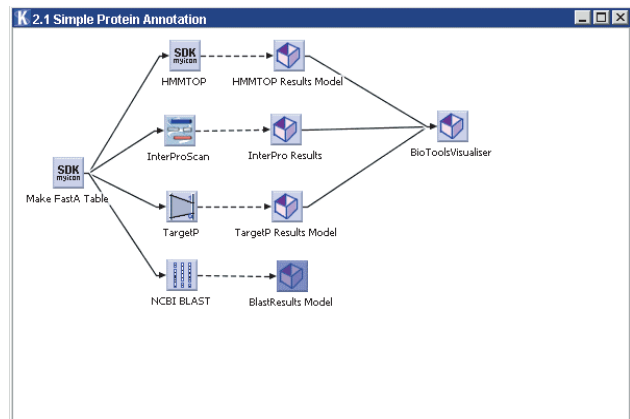


Fig. 4. Distributed protein annotation pipeline.

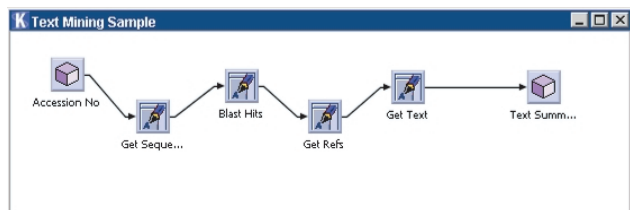


Fig. 5. Using the InfoGrid technology to find Medline abstracts about homologous proteins and then using text-mining technology to investigate the metabolic processes.

RELATED WORK

The Discovery Net project is based on building high-level services for Grid applications. Much related work is based on the Development of Grid Infrastructure. This includes the acknowledged Globus toolkit (Foster and Kesselman, 1997), which includes low-level tools for security (GSI), data transfer services (GridFTP) and resource allocation (GRAM). Beyond these tools, the Open Grid Software Architecture (OGSA) builds on the existing web services technology to provide a new standard called Grid Services. These overcome some of the drawbacks in building data-intensive processes into Web Services. The main drawback of the Globus project is the low-level abstractions. Discovery Net can be regarded as an application layer that builds on top of Globus to provide an application abstraction to the user.

In building a genome annotation pipeline there were several examples to base the case study on; one of the most popular is the Ensembl pipeline (Hubbard *et al.*, 2002), which automatically annotates the sequences of eukaryotic genomes, such as the human, mouse and zebrafish. The general pipeline of Ensembl in annotating genes proceeds by aligning known proteins deposited in the SPTREMBL database (Bairoch and Apweiler, 2000) to the genome, hence most known protein-coding genes are located. Genewise (Birney and Durbin, 2000) is used to provide a more accurate gene structure. In a similar manner, orthologous proteins and cDNAs from related organisms are aligned to the reference genome. The hidden Markov model (HMM) based program Genscan (Burge and Karlin, 1997) is subsequently used to scan the entire genome. The Genscan predicted genes that are confirmed by matches to proteins, mRNAs and UniGene clusters and that are supported by experimental evidence, such as ESTs, are assembled into genes (Hubbard *et al.*, 2002). Furthermore, Ensembl provides other nucleotide-level annotations, such as repeats, and functional annotations via InterPro (Apweiler *et al.*, 2001).

The Ensembl pipeline is more rigorous than the pipeline built in the Discovery Net case study in annotating the malarial genome. However, components from the Ensembl pipeline can be easily integrated into Discovery Net services. In addition, the users of Discovery Net can customise the pipeline to include or remove specific features, which is a major advantage since different parameters and methods are used in annotating different genomes.

Other examples include the Genotator (Harris, 1997) from the Berkeley Drosophila Genome Project and the Pipeline tool (Shah and Uberbacher, unpublished) at the Life Sciences Division of Oak Ridge National Laboratory. Both of these tools provide a specific workbench for performing genome annotation. It was based on the functionality of these types of tools and the visualisation

technology of the Artemis viewer that the requirements of our genome application systems were used.

The use of the web services model in life sciences has been popularised by the Omnigene project (Gilman, unpublished) at the Whitehead Institute. This is based around using the Web Services model to manipulate the distributed annotation servers based around the DAS protocol of the BioDas project (Stein, unpublished).

SUMMARY

We have presented the Discovery Net system as application level Grid Middleware. This is used to bridge the gap between the low level Grid technologies such as Globus and application users such as the bioinformatics community. To illustrate how this may be used we presented a case study where a genome annotation application was built within the Discovery Net Framework. This application was subsequently awarded the 'Most Innovative Data Intensive Application' at the IEEE Supercomputing 2002 High Performance Computing Challenge.

The future direction of the Discovery Net is to continue to develop the platform in light of the lessons learnt in producing our initial case study. This will include work in the Grid and High Performance Computing, Text analysis and Bioinformatics applications.

ACKNOWLEDGEMENTS

The authors would like to thank all other members of the Discovery Net team, especially Patrick Wendel and Dr Matt Howard for their help in developing the case study. We would also like to thank Dr Catherine Rice and Dr Nigel Carter at The Wellcome Trust Sanger Institute for their help and expertise. Discovery Net is an EPSRC project funded under the UK e-Science Programme.

REFERENCES

- Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
- Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res. Sep.*, **25**, 3389–3402.
- Apweiler,R., Attwood,T.K., Bairoch,A., Bateman,A., Birney,E., Biswas,M., Bucher,P., Cerutti,L., Corpet,F., Croning,M.D.R. *et al.* (2001) The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic Acids Res.*, **29**, 37–40.
- Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.*, **28**, 45–48.
- Birney,E. and Durbin,R. (2000) Using GeneWise in the *Drosophila* annotation experiment. *Genome Res.*, **10**, 547–548.
- Burge,C. and Karlin,S. (1997) Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, **268**, 78–94.

- Curcin,V., Ghanem,M., Guo,Y., Kohler,M., Rowe,A., Syed,J. and Wendel,P. (2002) *DiscoveryNet: Towards a Grid of Knowledge Discovery*, KDD-2002 Edmonton, Alberta, Canada, July 23–26. DeltaDot Ltd (2003) <http://www.deltadot.com>.
- Emanuelsson,O., Nielsen,H., Brunak,S. and von Heijne,G. (2000) Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *J. Mol. Biol.*, **300**, 1005–1016.
- Foster,I. and Kesselman,C. (1997) Globus: a metacomputing infrastructure toolkit. *Int. J. Supercomputer Appl.*, **11**(2), 115–128.
- Gilman,B. The Omnigene Project <http://omnigene.sourceforge.net/>.
- Harris,N.L. (1997) Genotator: a workbench for sequence annotation. *Genome Res.*, **7**, 754–762.
- Hubbard,T., Barker,D., Birney,E., Cameron,G., Chen,Y., Clark,L., Cox,T., Cuff,J., Curwen,V., Down,T.*et al.* (2002) The Ensembl genome database project. *Nucleic Acids Res.*, **30**, 38–41.
- Juhl Jensen,L., Gupta,R., Blom,N., Devos,D., Tamames,J., Kesmir,C., Nielsen,H., Stærfeldt,H.H., Rapacki,K., Workman,C.*et al.* (2002) *Ab initio* prediction of human orphan protein function from post-translational modifications and localization features. *J. Mol. Biol.*, **319**, 1257–1265.
- Knudsen,S. (1999) Promoter 2.0: for the recognition of PolIII promoter sequences. *Bioinformatics*, **15**, 356–361.
- Lowe,T.M. and Eddy,S.R. (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res.*, **25**, 955–964.
- O'Donovan,C., Martin,M.J., Gattiker,A., Gasteiger,E., Bairoch,A. and Apweiler,R. (2002) High-quality protein knowledge resource: SWISS-PROT and TrEMBL. *Brief. Bioinform.*, **3**, 275–284.
- Rice,P., Longden,I. and Bleasby,A. (2000) EMBOSS: the European molecular biology open software suite. *Trends Genet.*, **16**, 276–277.
- Rutherford,K., Parkhill,J., Crook,J., Horsnell,T., Rice,P., Rajandream,M.-A. and Barrell,B. (2000) Artemis: sequence visualisation and annotation. *Bioinformatics*, **16**, 944–945.
- Shah,M. and Uberbacher,E. Genome analysis pipeline at <http://compbio.ornl.gov/tools/pipeline/>.
- Smit,A.F.A. and Green,P. RepeatMasker at <http://ftp.genome.washington.edu/RM/RepeatMasker.html>.
- Stein,L. (2001) Genome annotation: from sequence to biology. *Nature Reviews Genet.*, **2**, 493–503.
- Stein,L. The BioDAS project <http://biodas.org/>.
- Syed,J., Guo,Y. and Ghanem,M. (2002) *Discovery Processes: Representation And Re-Use*, All Hands Meeting.
- Tusnády,G.E. and Simon,I. (2001) The HMMTOP transmembrane topology prediction server. *Bioinformatics*, **17**, 849–850.