GBF Monographs Volume 18

Bioinformatics: From Nucleic Acids and Proteins to Cell Metabolism

Edited by
Dietmar Schomburg
Uta Lessel

Contributions to the Conference on "Bioinformatics" October 9 to 11, 1995 Braunschweig, Germany

VCH

This Page Intentionally Left Blank

Bioinformatics: From Nucleic Acids and Proteins to Cell Metabolism

GBF Gesellschaft für
Biotechnologische Forschung mbH

Distribution:
VCH, P.O. Box 101161, D-69451 Weinheim (Federal Republic of Germany)
Switzerland: VCH, P.O. Box, CH-4020 Basel (Switzerland)
United Kingdom and Ireland: VCH, 8 Wellington Court, Cambridge CB1 1HZ (United Kingdom)
USA and Canada: VCH, 220 East 23rd Street, New York, NY 10010–4606 (USA)
Japan: VCH, Eikow Building, 10-9 Hongo 1-chome, Bunkyo-ku, Tokyo 113 (Japan)

ISBN 3-527-30072-4 ISSN 0930-4320

GBF Monographs Volume 18

Bioinformatics: From Nucleic Acids and Proteins to Cell Metabolism

Edited by
Dietmar Schomburg
Uta Lessel

Contributions to the Conference on "Bioinformatics" October 9 to 11, 1995 Braunschweig, Germany

VCH

Prof. Dr. Dietmar Schomburg
Dr. Uta Lessel
GBF
Gesellschaft für
Biotechnologische Forschung mbH
Molekulare und Instrumentelle Strukturforschung
Mascheroder Weg 1
D-38124 Braunschweig
Federal Republic of Germany

Standard Edition published jointly by
VCH Verlagsgesellschaft mbH, Weinheim (Federal Republic of Germany)
VCH Publishers, Inc., New York, NY (USA)

Copy Editor: Dr. J.-H. Walsdorff, Gesellschaft für Biotechnologische Forschung, Braunschweig
Responsible for the contents: The contributors

Cover illustration: Schroers Werbeagentur, Braunschweig

Library of Congress Card No. applied for.

A catalogue for this book is available from the British Library.

Deutsche Bibliothek Cataloguing-in-Publication Data:
Bioinformatics : from nucleic acids and proteins to cell metabolism ; contributions to the Conference on "Bioinformatics", October 9 to 11, 1995, Braunschweig, Germany / ed. by Dietmar Schomburg ; Uta Lessel. – Weinheim ; Basel ; Cambridge ; New York, NY ; Tokyo : VCH, 1995
(GBF monographs ; Vol. 18)
ISBN 3-527-30072-4
NE: Schomburg, Dietmar [Hrsg.]; Conference on Bioinformatics <1995, Braunschweig>; Gesellschaft für Biotechnologische Forschung <Braunschweig>: GBF-Monographien

Printing: betz-druck GmbH, D-64291 Darmstadt
Bookbinding: J. Schäffer, D-67269 Grünstadt
Printed in the Federal Republic of Germany

Preface

The term bioinformatics has two quite distinct meanings. It may describe information handling in living organisms, and it is widely used for the application of computer science to biological problems. It is this second area which is covered in this book. The series of articles presented here represents a selection of the papers given at an invigorating conference on Bioinformatics/Computer Application in the Biosciences, held in October 1995 in Braunschweig at the German National Laboratory for Biotechnology.

The development and use of computer applications in the biological sciences, though initiated rather late compared to the situation in physics and chemistry, has reached a high standard nowadays and has become an indispensable part of any research in this area. A strong impetus has come from modern gene sequencing projects and also from the rapid development in the field of structural biochemistry, i.e. the determination of protein and DNA/RNA 3D-structures as well as rational protein engineering and design.

This is reflected in the subjects covered in the articles in this book. They describe the present state in this field, in particular the following facts become obvious:

- The use and development of biological data bases has become an essential foundation for research in protein science and molecular biology.

- Whereas the coding regions of DNA have been the main target of research in the past, nowadays the non-coding regions and RNA are receiving closer attention.

- The sequence comparison and correct alignment of protein sequences is a prerequisite for any protein engineering. Although routinely used in almost all biochemistry laboratories, alignment of sequences with low homology still requires further intensive research so that significantly better results can be produced than those currently available.

- The description and simulation of the interactions between different biological molecules will be one of the fascinating areas of future research.

- In addition to understanding the biological processes on a molecular level, we have to simulate the metabolism in the living cell in order to achieve real metabolic design for the optimal biotechnological production of compounds.

Whereas the first development of these methods stems from the sixties and seventies, it is only recently that biologists, chemists and computer scientists have channelled their expertize into large scale collaborative projects aimed at the advancement in this exciting area. Government programs started, for example in Germany and the UK, have provided extra money for joint projects involving computer scientists and biologists. Together with the rapid progress in modern biology and biotechnology, we can expect to see wide-ranging new developments in bioinformatics in the years to come.

Dietmar Schomburg Uta Lessel October 1995

This Page Intentionally Left Blank

This Page Intentionally Left Blank

Contents

List of Authors

This Page Intentionally Left Blank

I. Biological Data Bases

This Page Intentionally Left Blank

An Integrated Services Approach to Biological Sequence Databases

Heumann K., Harris C., Kaps A., Liebl S., Maierl A., Pfeiffer F., Mewes H.W.*

MIPS at the Max-Planck-Institut für Biochemie, Am Klopferspitz, 82152 Martinsried, Germany
e-mail: heumann@mips.embnet.org mewes@mips.embnet.org
Phone: +49 89 8578 2451 FAX: +49 89 8578 2655

Abstract

Database users in molecular biology are faced with steadily increasing amounts of raw data, multiple database providers and services. Here we describe the integration of a set of previously isolated database services and demonstrate their accessibility through a uniform user interface. A multi-layered software architecture is applied to make different degrees of service integration transparent to the user. We focus on the design of specialized gateways that integrate services differing in temporal behavior and stateless or state dependent operation. Gateways may reside on heterogeneous platforms. A link layer is introduced to integrate individual query functions in order to interrelate simple, complex and state dependent services through a common, unique interface. It is possible to generate new complex services by a combination of multiple functions. We describe the application of the World Wide Web (WWW) as the implementation framework of the interface layer. To assure interoperability of services, integrity of data resources must be supervised. Consistency control is issued by a dedicated synchronization layer.

Introduction

Users of molecular biology databases that wish to benefit from multiple services provided by different resources are confronted with various user interfaces that must be mastered prior to exploring a particular service. Because these isolated services were developed independently, rarely was an interrelation with other services considered at the time of development. The user must establish this relationship in order to evaluate the results from different resources or analytical tools. Recently, the need for database interoperability and the need to develop effective mechanisms for inter-database communication have obtained increasing attention [GEO95]. Moreover, information provided by independent sites is notoriously inconsistent, i.e. information

rendered from different sites in reply to the same query is discrepant. As a data and service provider, we address the problem of incongruity by formally separating concerns relating to (1) external, public access to the database and (2) internal database management. This strategy permits us to support multiple data access methods (employing fundamentally different methodologies) and make them available as specialized services. The internal database management, which is tailored to meet the internal needs for data processing, remains untouched. This allows us to refine and enhance the conceptual database schema employed internally without interfering with data access and information retrieval. Thus new technologies and services can be incorporated and made publicly available without compromising the stability of other established database interfaces.

Database services vary widely in (1) the type of data retrieved or explored, (2) their temporal behavior, (3) their principal type of operation as stateless or state dependent, and (4) the platform on which they reside. Layered software architecture is an established concept to hide the heterogeneity of services from the user; different layers stand for different degrees of integration. Service integration is a directed process mediated by the interface layer. Therefore, constraints and limitations imposed by the interface layer must be compensated by the underlying layers. In extreme cases, applications must be reimplemented in order to match the requirements of the interface layer selected. As an example, we discuss reengineering of a service versus integration of a service by a specialized gateway.

Integrated layers make the interconnection of services possible. Critical to this approach is a formal definition of database resources. Such a definition allows the data to be transformed reliably and unambiguously into a wide variety of different physical representations by the development of simple data filter programs. Different formats (e.g. EMBL, GenBank, PIR, ASN.1), employed by various database centers, have hindered integration of macromolecular sequence data banks. Attempts to standardize sequence database formats were not successful [DOE93]. The ability to access data in a variety of forms eliminates problems associated with syntactic variability. However, this approach does not address semantic inconsistencies. The ability to crossvalidate data and to provide robust and correct paths for navigation among databases requires semantic compliance. Current retrieval systems like SRS [ETA93], ATLAS [NAT94], and hypertext based interfaces (WWW/Mosaic [LSC94] [BL94]) lack reliable, verified, and robust cross-database links. These limitations are intrinsic since cross-references do not ensure the correct semantic relation between the linked objects. We therefore propose to define cross-database relations formally within a link layer allowing for semantic mapping between different database formats.

Databases are not static, but subject to continuous change. In a heterogeneous environment of specialized services, resource databases may be spread or replicated across networks. Ensuring consistency in such an integrated and distributed database system requires distributed transaction protocols that synchronize updating of resource databases and services within a synchronization layer. This approach can be further extended to integrate the database user in a Computer Supported Cooperative Work (CSCW) approach to take advantage of the user's expert knowledge for refinement of the services [GRE94].

Concept

Many approaches in database integration are based on client server technology. We apply variants of network abstractions, including Remote Procedure Calls (RPCs) [BIR94], to join sets of services of similar type in defined layers. Thus, a multi-layered software architecture [TAN89] is applied to make different degrees of integration transparent to the user. Figure 1 gives an overview of the arrangement of distinct layers.

1 *Interface layer*: The user expects a homogeneous intuitive graphical user interface to access services in a uniform way, independent from the platform used and its location in the international network. Therefore the interface layer should apply a standardized network transparent GUI-toolkit. In order to express relations between data items of different services, the toolkit should follow hypertext concepts. The standard tool of choice is the World Wide Web (WWW). WWW is an Internet navigation tool which masks the complexities of remote operations. WWW uses hypertext, formatted text with links, known as anchors, to guide the user to other documents or programs. The WWW is an undetermined client-server application with multiple clients and multiple servers; any client may issue any request to any server at any time.

2 *Link layer*: The link layer uncouples the gateway layer from the interface layer. This takes account of the rapid development in standardized GUI-tookits, allowing for easy migration from one interface toolkit to another without affecting the gateway layer. In the link layer, relations between services are established. These links can also be expressed across networks. The link layer has knowledge of all services accessible. Thus the link layer defines the "integrated service". It therefore may also be named the "conceptual integration layer". Using the WWW on the interface layer requires the output retrieved from the gateway layer

to be mapped to documents conforming the standards of the hypertext markup language (HTML) [HTM94], expressing links as anchors for the WWW.

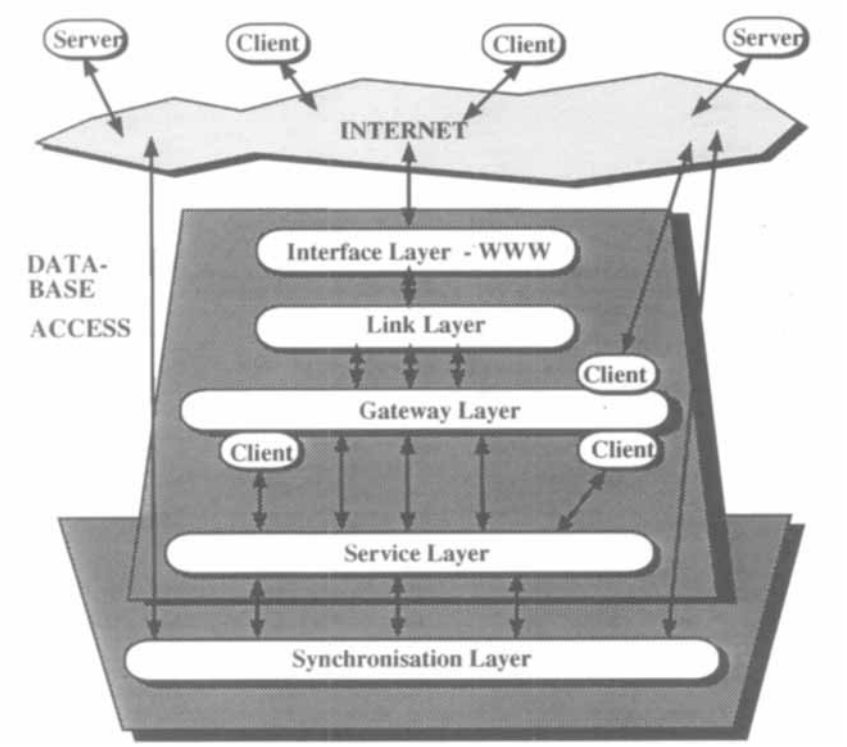

Figure 1 A multiple layered architecture for service integration.

3 *Gateway layer*: The gateway layer must compensate for the constraints imposed by the interface mechanism without modification of the underlying service. It connects the individual services to the interface layer. The gateway renders the specific characteristics of services (temporal behavior, statelessness or state dependency and residence on heterogeneous platforms) transparent to the interface layer. Therefore, services are not restricted by the constraints of the interface layer.

4 *Service layer*: Services may be classified according to the type of gateway they require. We include the resource databases within the service layer. New services may be developed independently and integrated into the system at any time.

5 *Synchronization layer*: This layer supervises integrity of data (and links) by dedicated transaction protocols. This layer can also be viewed as orthogonal to layers 2-4 ensuring mutual synchronization across layers.

Components of the service layer

When integrating a set of heterogeneous services, different applications and their access methods must be standardized. In order to simplify standardization of interaction we classify services according to their lifetime in relation to the time constraints imposed by the interface mechanism:

- **primitive services**: These services perform standard on-line request serving based on database (*Get*) or application (*Generate*) primitives. There is a one-to-one relationship between an user input and the output of the service. The gateway directly associates a command execution with a user request. Primitive services are often used as basic components by more complex services.

- **simple service**: These services perform on-line and stateless requests. They may involve specialized services, possibly including subqueries or subdialogues. Typical cases are services (*Retrieve*) that render precalculated relations intrinsic to the data accessible (e.g. access through indices). Most often, simple services are used to navigate through the data set.

- **complex service**: These services perform state dependent operations maintained throughout a session. Complex services have lifetimes extending beyond single user requests. They need specialized gateways that are able to connect on-line requests to persisting application sessions. Complex services are often entry points to a database search.

- **dynamic service**: These services perform requests that outlast interactive sessions. These event handling services (*Alert*) can use any other service but have a persisting effect. Alert services have a maintenance component that operates on-line to define the persistent request. The operational component of the alert service operates detached and is triggered by the synchronization layer. As an additional feature, differential evaluation of the query results may be supported.

Figure 2 demonstrates the differences in temporal behavior of the service classes within a WWW session.