IMGT/LIGM-DB: A Systematized Approach for ImMunoGeneTics Database Coherence and Data Distribution Improvement

Véronique Giudicelli (a), Denys Chaume (b) and Marie-Paule Lefranc (a)

(a) Laboratoire d'ImmunoGénétique Moléculaire, LIGM, UMR 5535 (CNRS, Université Montpellier II), 1919 route de Mende, 34293 Montpellier Cedex 5, France, giudi@ligm.crbm.cnrs-mop.fr, lefranc@ligm.crbm.cnrs-mop.fr
(b) CNUSC, 950 avenue de Saint Priest, BP 7229, 34184 Montpellier Cedex 4, France, chaume@cnusc.fr

Abstract

IMGT, the international ImMunoGeneTics database (http://imgt.cnusc.fr:8104), created by Marie-Paule Lefranc, Montpellier, France, is an integrated database specializing in antigen receptors and MHC of all vertebrate species. IMGT includes LIGM-DB, developed for Immunoglobulins and T-cell-receptors. LIGM-DB distributes high quality data with an important increment value added by the LIGM expert annotations. LIGM-DB accurate immunogenetics data is based on the standardization of biological knowledge related to keywords, annotation labels and gene identification. The management of such data resulting from biological research requires an high flexible implementation to quickly reflect up-to-date results, and to integrate new knowledge. We developed a systematized approach and defined LIGM-DB systems which manage and realize the major tasks for the database survey. In this paper, we will focus on the coherence system, which became absolutely crucial to maintain data quality as the database is growing up and as the biological knowledge continues to improve, and on the distribution system which makes LIGM-DB data easy to access, download and reuse. Efforts have been done to improve the data distribution procedures and adapt them to the current bioinformatics needs. Recently, we have developed an API which allows Java™ programmers to remotely access and integrate LIGM-DB data in other computer environments.

Keywords

API; Database distribution; Data quality; Immunogenetics; Immunoglobulin; Java; Object-oriented analysis; Relational database; Remote access; Sequence expertise; T cell receptor.

Introduction

The immune system has evolved to protect individuals against pathogenic viruses, micro-organisms and parasites. It is vital therefore that individuals have a normal immune system. Normal immune responses depend on the ability to recognize foreign molecules or antigens on the potential pathogen in order to eliminate the source of the antigen. The molecules involved in the recognition of antigens are encoded by the immunoglobulin superfamily, and this includes immunoglobulins (Ig), T cell receptors (TcR) and Major Histocompatibility Complex (MHC).

Scientists over a number of years have been rapidly sequencing the DNA which encodes the molecules of the immune system. Presently more than 25,000 gene sequences have been determined, and this number will double over the next two years. The molecular synthesis and genetics of the Ig and TcR chains is particularly complex (Lefranc 1990, Honjo and Alt 1995), and the generalist databases cannot adequately label or annotate these sequences. Moreover, there is no way to search exhaustive and efficient links between pathologies and sequence components. This makes the core data held at these generalist databases difficult to use or interpret for researchers and clinicians. Obviously, a specialist database was required to add value, and to link it to other biological databases.

The international ImMunoGeneTics (IMGT) database (Lefranc et al. 1998) was created in 1992 by Marie-Paule Lefranc (CNRS, Université Montpellier II, Montpellier, France, lefranc@ligm.crbm.cnrs-mop.fr). IMGT (**http://imgt.cnusc.fr:8104**) comprises alignment tables and expertly annotated sequences, and consists of three databases: LIGM-DB for Ig and TcR, MHC/HLA-DB and PRIMER-DB (an Ig, TcR and MHC-related primer database), these last two are currently in development.

Data management resulting from biological research requires an high flexible implementation to quickly reflect up-to-date results, integrate new knowledge and restore high quality data. We defined LIMG-DB quality parameters according to information quality general view (see Appendices 1-3), and with respect of the following main objectives:

- to contain all Ig and TcR sequences published in generalist databases,
- to provide specific immunogenetic expertise using IMGT standardized annotation rules ,
- to be updated with publication of new scientific knowledge,
- to be checked for incoherence,
- to be easily accessed by non computer-scientist end-users,
- to be open to the whole scientific community,
- to provide portability of the application to other computer environments.

The first task of LIGM-DB is to be able to cope with the enormous flow of new data. The publication of novel Ig and TcR sequences continues at an ever increasing pace and the development of automated sequencing techniques suggests that this trend will continue for the foreseeable future. The LIGM-DB completeness in Ig and TcR sequences is certified by a collaboration with the EMBL generalist database (Stoesser et al. 1998), which daily sends by e-mail new or updated Ig and TcR submitted entries to LIGM. The administration system has been developed to deal with multiple sources of sequence data reception: core data from EMBL, expertized data from LIGM annotators, and in a near future data from direct submission by the authors (in development).

The second task is the improvement in the automation of the annotation procedure. Sequence annotation, a time consuming and an elaborate step, is the limiting factor for the addition of expert data onto the nucleotide sequence information. To reach that goal, several tools which search for conserved motifs in Ig and TcR sequences, and align nucleotide and protein sequences (Lefranc et al. 1998) are integrated in the annotation system. These tools, currently available for LIGM experts, will be adapted for WWW users.

The third task is to ensure the coherence between the multiple sources of data, core data from EMBL, annotation data from LIGM experts and from the authors, and knowledge data issued from the immunogenetic research. Coherence evaluation of the database comprises many aspects that combine biology and computer sciences. It concerns the application integrity according to the relational database management concept, and data integrity, consistency and accuracy, according to the biological rules. The resulting issues of the coherence evaluation are to verify that data follow the rules defined by the biologist, to identify potential sources of incoherence and to propose solutions in case of errors. The coherence system is being designed to support these concepts and to easily evolve.

The fourth task corresponds to the improvement of data distribution. This includes the development of an easy-to-use interface on the web, which is the best way to make data and related information available for the scientific community, and the distribution of IMGT data in standard format like the commonly used flatfile format (EMBL-like or ASN.1 (Normes ISO8824 and ISO8825, CCITT recommandations X.208 and X.209). However the best way to allow the developers to remotely access and integrate LIGM-DB data in other computer environments is to provide them with an API (Application Programming Interface).

In the conception and initial development phase of IMGT, priority was given to the administration and annotation systems in order to manage LIGM-DB data flow. During that period, rules for coherence control were established step by step and applied by experts and annotators before becoming an independent system ready to be implemented in the IMGT application. In the same way, priority for data distribution was first restricted to the WWW interface and to the EMBL-flatfile production. Significative improvement in data distribution, particularly with API production, led us to develop an independent data distribution system. In this paper, we will focus on the conception and development of the coherence and the distribution systems.

Methods

Material, Languages and General Tools

IMGT/LIGM-DB is a relational database managed by the Sybase RDBMS as many other biological databases. Sybase has been chosen for its robustness and its available tools that allow to check and maintain data consistency. Sybase Transact SQL Server system 11 is running on an IBM RS6000/900 server located on the French national academic computer center (Centre National Universitaire Sud de Calcul). The current main user access is a Web interface developed with functionalities of HTML 3 language to make it available from Web browser as Netscape Navigator or Microsoft Internet Explorer. Apache 2 has been chosen for Web server. CGI programmes are developed using Java™ Development Toolkit 1.1 (see Appendix 4). Database access is performed using jCONNECT for JDBC from Sybase.

Design of the Database: a Systematized Analysis

To reach the objectives mentioned in the "Introduction", we have defined a new analysis method based on a systematized approach.

System Modelling. Our approach is close to an Object Oriented analysis but can be used for any kind of data or

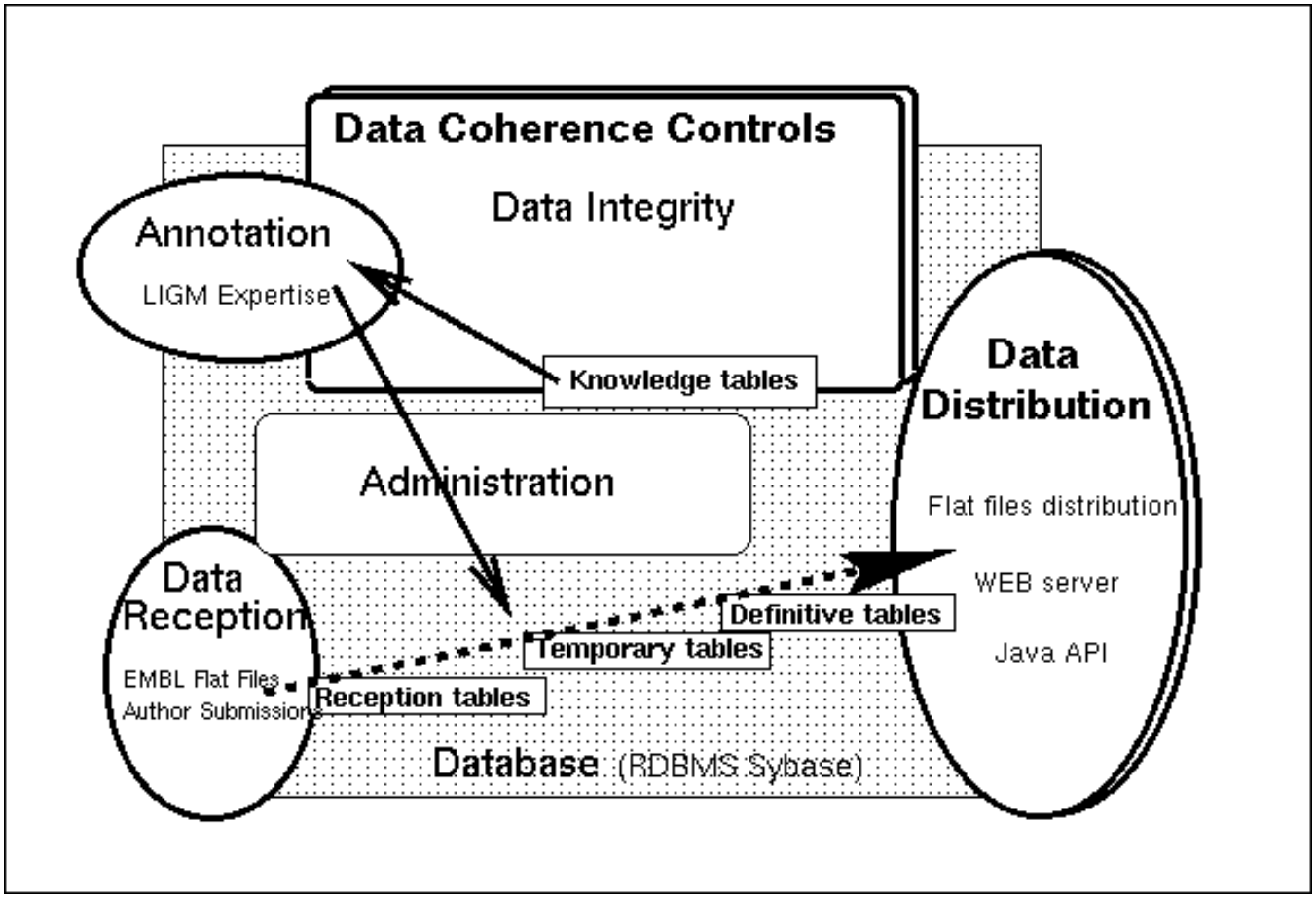

Figure 1: IMGT/LIGM-DB application general scheme. This scheme shows six master systems:
- the "Data Reception" system which receives data from either EMBL databank or from direct author submission and parses them into temporary tables,
- the "Annotation" system which brings the biological expertise in,
- the "Administration" system which follows the status of the entries from reception to distribution,
- the "Data Coherence Controls" system which uses "Knowledge" and "Administration" tables to verify data integrity and biological coherence,
- the "Data Distribution" system which takes in charge the distribution of data in various formats and various interfaces,
- and the "Database" system, i.e. the table set managed by the RDBMS Sybase, which is the main container of data.
Data is stored in the database and migrates (dotted arrow) from "Data Reception" to "Data Distribution" with expertized information added and under coherence controls (solid arrows).

processing and relies on a unique concept: the concept of *system* which can be used as well for data in a Entity/Relation scheme, as for WEB server organization, sequence analysis processing, graphical objects displayed in a Graphic User Interface, Java classes used by data management programmes... Each system is composed of various subsystems which are themself composed of more precise subsystems and so on. This system concept has beenused to represent the functional parts of the application such as the "Data Distribution" system, and to organize the relational "Database". The general scheme of the IMGT/LIGM-DB application shows such master systems (Figure 1).

Development Phases. Systems are defined to manage the application and each of them has its own evolution. Details of the four development phases are described in Figure 2. This dynamic approach allows to review and update one part of the application without having to do a new complete analysis. We only need to take care of the interfaces between the different systems.

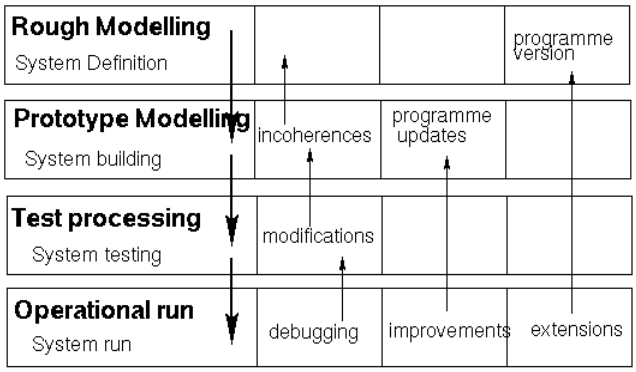

Figure 2: The Systematized Analysis Method. It comprises four development phases for each defined system:
1. During the "**Rough Modelling**" phase the new system is divided into subsystems according to a decreasing analysis.
2. Then, the low level subsystems which do not yet exist are developed, while the existing ones are updated in the "**Prototype Modelling**" phase. In that way the targeted system is built step by step from either existing or new created low level subsystems. During that phase incoherences may appear which will oblige to go back to the "Rough Modelling" phase.
3. Once the programmes have been written, the new system is integrated into the application to be tested. "**Test Processing**" can lead to some modifications of the first defined functionalities and oblige to go back to the "Prototype Modelling" phase. The tests may even discover incoherences which will need a return to the "Rough Modelling" phase.
4. Finally the system comes into the "**Operational run**" phase. Using it, we may find minor errors which require debugging and generally a return to "Test Processing" level. If significative improvements are needed, this will imply new developments at the "Prototype Modelling" level. The addition of functional extensions will need a new analysis at the "Rough Modelling" level.

Examples: LIGM-DB data coherence and data distribution systems. In the IMGT/LIGM-DB application low level subsystems are described according to their content:

1. tables, constraint rules, catalogued procedures, triggers, granting managed in the relational database.
2. HTML documents and CGI programmes dispatched by the WWW server Apache.
3. Java classes used for processing, interfaces and data transmissions,
4. external C programmes with their own data and working files (DNAPLOT or LIGMOTIF (Lefranc et al. 1998)).

For example, the "Data Coherence Controls" system includes Administration and Knowledge tables used by the triggers which control Referential Integrity Constraints, and methods in Java classes used in the incoming data processing. The "Data Distribution" system is composed of definitive tables, HTML forms used for requests by the WWW server, and Java programmes to product flatfiles.

Results and Discussion

Data Coherence Controls System

LIGM-DB Application Integrity. Data stored in LIGM-DB database originate from multiple sources:

- the core data, like nucleic sequence enchainment and composition from EMBL. These are not updated by LIGM experts since identical accession numbers are kept as sequence identifiers in LIGM-DB,
- annotation data from direct author submissions. They are integrated in the database and checked by LIGM (in development),
- annotation data produced by the LIGM annotators who are allowed to alter the corresponding tables by data insertion, update or deletion.

The IMGT/LIGM-DB application manages and coordinates the entry of multiple sources of information in order to avoid data corruption within the database and unauthorized modification data. This is partly performed by the RDBMS Sybase with the management of concurrence and serialization of transactions. However, permission for sequence loading and sequence annotation by LIGM experts is mainly under the control of the administration system (Figure 1). At anytime, the administration sytem knows the status of all entries in the database. When a sequence arrives from EMBL, it is immediately identified either as a new entry for LIGM-DB database, or an updated one, or a wronly assigned Ig or TcR sequence. As the sequence goes into the annotation process, it is locked for an identified annotator, and its related data cannot be altered by anybody else. Security provided by Sybase, combined with the administration system ensure the IMGT/LIGM-DB application integrity: restored data for distribution are extracted from core data and from the annotation data stored in the database without corruption.

LIGM-DB data integrity. LIGM-DB basically follows the primary rules of relational databases like the identification of each entity using a unique primary-key value.

Domain integrity control is applied to field values with simple domain data type: this is the case of the sequence length, which must be a positive integer.

In most cases, field values have biological significance, like keywords, feature labels or gene names. This justifies the construction of knowledge table set which contains the

allowed values with their characterization. Domain constraints have been then translated in referential constraints between sequence related tables and knowledge tables.

LIGM-DB knowledge tables. Knowledge tables have been established to record and standardize theorical and experimental concepts resulting from the immunogenetic research. Their content is under the IMGT coordinator responsability. This knowledge is used to characterize Ig and TcR sequences in LIGM-DB application. The Entity-Relation modelling of LIGM-DB knowledge data is shown in Figure 3. This biological knowledge constitutes an important part of search criteria available from the WWW interface.

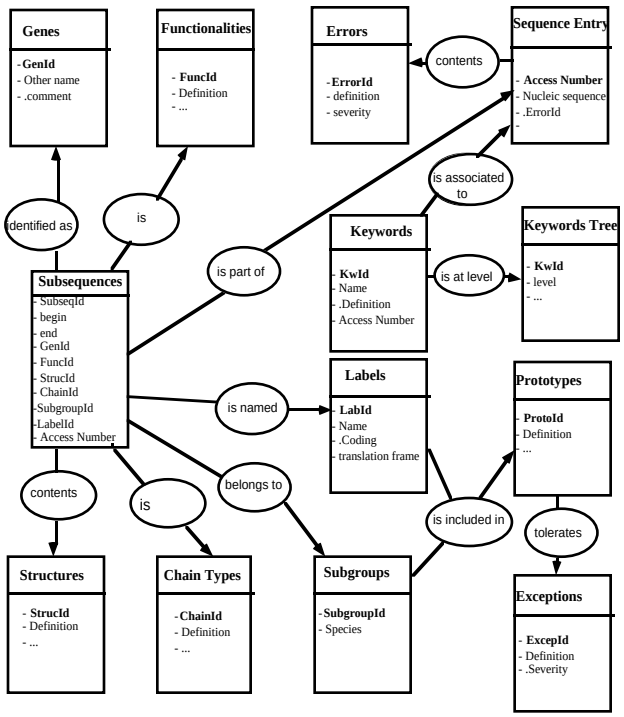

Figure 3: The Entity-Relation modelling of LIGM-DB knowledge data

Lists of allowed values: We designed a first set of knowledge tables which contain the lists of allowed values with their biological characterization. These tables include:

- the standardized keywords for description of Ig and TcR sequences,
- the IMGT gene name nomenclature,
- the sequence chain type (Ig-Heavy, TcR-Beta, ...),
- the variable region subgroups defined for each species,
- the structure of the sequences
- the sequence functionality,
- the feature labels: 177 feature labels are necessary to describe all structural and functional subregions that compose Ig and TcR sequences, whereas only 7 of them are available in EMBL (Stoesser et al. 1998), GenBank (Benson et al. 1998) or DDBJ (Tateno et al. 1998). Annotation of sequences with these labels constitutes the main part of the expertize. Information useful either to check the data consistency or to help annotators is added in these tables. For each subregion, it is indicated if it can be translated, and if so, which is the corresponding translation frame by default. LIGM-DB keywords, label list, label definitions and representation are available at URL http://imgt.cnusc.fr:8104.

Knowledge organization: Because part of knowledge data in lists of allowed values are dependent of each other, a second set of tables was required to record, to organize and to control the relationships existing between knowledge data. Two tables have been defined so far, one for the tree organization of keywords, the second for the prototype organizations of labels. Prototypes represent the organizational relationship between feature labels and give information on the order and the expected length (in nucleotide number) of the labels: for instance, in the prototype which corresponds to a germline V-GENE, it is recorded that the leader peptide(L-PART1) is always in 5' of V-EXON label (Figure 4).

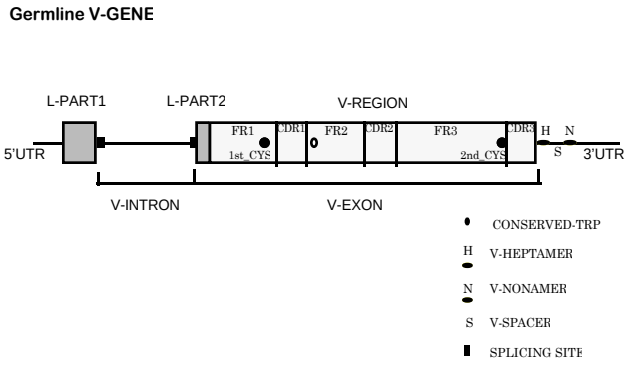

Figure 4: A simplified scheme of a germline V-GENE prototype with the main feature labels. Boxes indicate coding regions, lines indicate untranslated regions and intron.

Prototypes can apply to general configuration of Ig or TcR, independently of the chain type, the species or any other parameters like functionality: of course, the more general the prototype is, the less precise the relations are. However, prototypes may also be established for very precise cases when sequence characteriztics are clearly established. For example, the general V-GENE prototype

tolerates a rather large variation of the FR and CDR length (Lefranc 1997) valid whatever the species and the chain type (Figure 5A), whereas exhaustive studies in human have led to establish very precise prototypes for each Ig or TcR, chain type and subgroup (see Figure 5B).

(A)

	FR1-IMGT	CDR1-IMGT	FR2-IMGT	CDR2-IMGT	FR3-IMGT	CDR3-IMGT
Aminoacid numbering	1 -->26	27 -->38	39 -->55	56 -->65	66-->104	105-->115
Number of aminoacids	25 to 26	5 to 12	16 to 17	0 to 10	36 to 39	2 to 11

(B)

IGHV sub-group	FR1-IMGT 1--->26	CDR1-IMGT 27--->38	FR2-IMGT 39->55	CDR2-IMGT 56--->65	FR3-IMGT 66-->104	CDR3-IMGT 105->115
	25	8 to 10	17	6 to10	38	2, 3
1	-1 (AA 10)	8		8	- 1 (AA 73)	2
2	-1 (AA 10)	10		7	- 1 (AA 73)	3
3	-1 (AA 10)	8		6, 7, 8, 10	- 1 (AA 73)	2, 3
4	-1 (AA 10)	8, 9, 10		7	- 1 (AA 73)	2
5	-1 (AA 10)	8		8	- 1 (AA 73)	2
6	-1 (AA 10)	10		9	- 1 (AA 73)	2
7	-1 (AA 10)	8		8	- 1 (AA 73)	2

Figure 5: FR and CDR length (A) in the general V-GENE prototype, (B) in human V-GENE prototypes for the different subgroups of the Ig heavy chain. As an example, the CDR2-IMGT length varies from 0 to 10 aminoacids in the general V-GENE prototype, whereas it is restricted to 7 aminoacids in the specific human IGHV2 subgroup prototype.

Exception management: because immunogenetics data come from research and from alive subjects, it is necessary to accept the existence of 'natural exceptions' to the designed prototypes. The exceptions, associated with a rule transgression severity evaluation, are recorded in exception tables. According to the degree of severity, actions will be triggered off. This table is very carefully filled in to only enter exceptions which need to be taken into account and to avoid the introduction of uncontrolled inconsistent data.

Error checking (in development): since LIGM-DB contains all Ig and TcR sequences recorded in generalist databases, we need to accept sequences with errors, waiting for the updates from the authors. Checking entry errors is an important aspect of the coherence control which will be developed in the future to make end-users aware of error types we meet and of their incidence in IMGT expertise.

Coherence control. The control of coherence is implemented as Java classes. It is being set up mainly:

- to check annotations before their entry into the database. This will be particularly useful for direct submission from the authors who are not always aware of IMGT annotation rules.
- to check data already entered in the database to trap errors and inconsistencies.
- to identify data that need to be updated when an annotation rule is modified, and to perform automatized consistent changes. As an experimental result, we currently use this tool to update old annotations by taking into account the new IMGT numbering of FR and CDR (Lefranc 1997).

The annotation data coherence control is based partly on the data stored in three sets of knowledge tables: list of labels is used to verify translation parameters (label coding or not, identification of the translation frame). Prototype tables allow to check the organizational description of labels. When a discrepancy is identified, control modules look for allowed exceptions, and the severity that has been notified, before giving the alarm. This detection will lead to a review of the annotations, and if necessary, to an evolution of the biological knowledge.

Benefits of the Data Coherence Controls system are multiple and allow LIGM-DB to reach its goals:

- high quality of data,
- application integrity
- knowledge data modelling,
- update of biological rules,
- improvement of the automation of annotation,
- quality of exchange and interpolarity with other databases.

Data Distribution

The classical way to distribute data to biological scientists is to provide them with text files containing sequence information. This media is used by big generalist databanks (GenBank, EMBL, DDBJ) as well as by specialised databases. These files, designated as "flatfiles" as opposed to structured databases, can be obtained either by CD-ROM distribution, e-mail subscription or internet FTP anonymous accesses. More recently, database providers have opened public WWW servers from where everybody can request sequence information. Some query tools have now an universal usage like SRS (Etzold, Ulyanov, and Argos 1996) or Entrez (see Appendix 5). By these ways biologists can easily retrieve selected flatfile entries.

Since 1995, the IMGT internet server (http://imgt.cnusc.fr:8104) offers accesses to the LIGM-DB database using a pleasant query tool constructed with HTML forms. By connecting to this server with a browser (Netscape Navigator, Microsoft Internet Explorer, or other), biologists can select the exact sequences they want, and can choose the part of information they need (including flatfile entries !).

Now IMGT application proposes two new access media to permit application developers to directly retrieve LIGM-DB data from the database without using flatfiles and without filling interactive forms.The first one is a set of URL to get direct HTTP links to the IMGT server. This kind of access was already possible for various flatfile databanks, including IMGT/LIGM-DB, using SRS, but it is now possible to access directly LIGM-DB database. The second one is for Java developers who want to incorporate LIMG-DB data into their own application. Based on the Java/RMI (Remote Method Invocation) protocol, the IMGT Application Programming Interface is a package of classes useful to retrieve remote IMGT data objects.

Direct links to the IMGT server. Two sets of data can be retrieved with these links (see tables below):

1. knowledge data: keywords, labels, groups, chain types, genes, prototypes...
2. sequence related data: nucleic sequences, references, annotations,...

URL of these links are in the following format:

1. for knowledge data:

```
http://imgt.cnusc.fr:8104/cgi-bin/IMGTlect?query=#info
```

2. for sequence data:

```
http://imgt.cnusc.fr:8104/cgi-bin/
IMGTlect?query=#info+#access
```

where #info is the information number (see tables below) and #access is the accession number of the sequence.

Knowledge data examples:

#info	retrieved data	query parameter
7	label list	query=2+X03960
8	keyword list	query=3+X03960

Sequence data examples:

#info	retrieved data	query parameter
2	Catalog entry and References	query=2+X03960
3	sequence in FASTA format	query=3+X03960
4	sequence with translations	query=4+X03960
5	LIGM flatfile	query=5+X03960
6	LIGM annotation	query=6+X03960
11	sequence in dump format	query=11+X03960
12	coding subregions	query=12+X03960

The complete list of available information can be found from the IMGT home page.

IMGT/LIGM-DB API. The IMGT/LIGM-DB API is a set of two Java packages which can be used by Java programmers: the *IMGTData* package which contains the class definitions for sequences and sequence related data, and the *IMGTSelect* package which contains the class definitions for query and retrieve data. A complete description of these packages is available on the IMGT server.

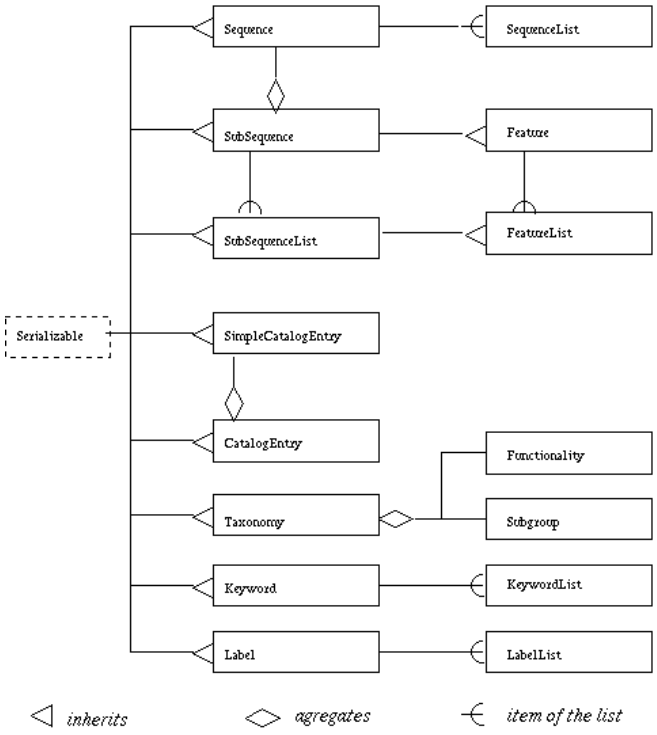

Figure 6: Object Model of the IMGTData package (extract)

IMGTData package: In the IMGTData package, relational data is modelled with object classes for manipulation and processing. We can distinguish three sets of data:

1) Catalogue and Sequence subsystem:

- *CatalogEntry* to introduce accession numbers, mnemonics, dates, ...
- *SimpleCatalogEntry* to introduce accession numbers, mnemonics, and definition,
- *SequenceVersion* for the version number and release date information,
- *Sequence* for the nucleotide sequence,
- *SequenceList* for a list of Sequence objects,
- *SubSequence* for a substring of amino acid extracted from a sequence,
- *SubSequenceList* for a list of SubSequence objects.

2) References and Cross-references subsystem:

- *Reference* which contains literature references,
- *ReferenceList* for a list of Reference objects,
- *DBCrossReference* to introduce references to other databases.

3) Taxonomy and Annotation subsystem:

- *Feature* for labeled subregions defined on a nucleotide sequence,
- *FeatureList* for a list of Feature objects,
- *Functionality* to introduce functionalities,
- *Keyword* to introduce keywords,
- *KeywordList* for a list of keywords,
- *Label* to introduce subsequence labels used in features,
- *LabelList* for a list of label,
- *AnnotationList* for list of LabelList objects,
- *Species* to introduce species,
- *Subgroup* to introduce subgroups,
- *Taxonomy* to introduce species, chain type, group, subgroup, functionality, specificity, ...

Each object contains the methods to retrieve particular information. Figure 6 shows an extract of the Object Model of the IMGTData package.

IMGTSelect package: The IMGTSelect package includes tools to access data using two steps: the construction of the query with *SelectionEditor* objects, and the retrieval of the data with *Provider objects*.

1) Selection Editors

A query to the database is made using an object of Selection class. Once instantiated, a Selection object can be updated by SelectionEditor objects. SelectionEditor objects are able to add selection criteria into Selection objects. There are several selection editors which inherit the SelectionEditor class:

- *SelectionCatalog*: selection using catalog entries: accession numbers, mnemonics, dates, definition, ...
- *SelectionKeywords*: selection using keywords
- *SelectionFeatures*: selection using labeled sub-regions
- *SelectionReferences*: selection using references
- *SelectionTaxonomy*: selection using species, chain type, group, subgroup, funtionality, specificity,...

Once constructed the selection recorded by the Selection object will be translated by its "genereSql()" method to made the "from" and "where" clauses of the SQL query. Provider objects use this to made the SQL query sent to the Sybase database server.

2) Providers

Providers are objects which access the data and return data objects defined by the IMGTData package classes. They are defined by classes which inherit the Provider class. At each set of data corresponds a class of provider:

- *CatalogProvider*: gives catalogue entries information: accession numbers, mnemonics, dates, versions and releases, sequence lengths, ...
- *ReferenceProvider*: gives references, Medline cross references, ...
- *SequenceProvider*: gives the nucleic sequence, protein translations, ...
- *KeywordProvider*: gives keyword information: keyword list, sequence keywords, ...
- *AnnotationProvider*: gives information about LIGM added expertise: subregion characteristics, partiality, coding properties, ...
- *TaxonomyProvider*: gives information on species, chain type, group, subgroup, functionality, specificity,...

The client objects communicate with their homologous server objects to access data. The server objects access data from the database using a JDBC (Java DataBase Connectivity™) interface. Local application programmers just deal with client provider objects. Following is given one example of programme using IMGT/LIGM API.

Example: The following programme retrieves LIGM annotations of the sequence having the accession number given as argument from the IMGT/LIGM-DB database and prints out the feature list. It gives the protein translation of the V-REGION if it exists.

```
// import useful packages
import IMGTSelect.*;
import IMGTData.*;

public class sample1 {

 /** Providers.
      "AnnotationsClient" is the Provider
      used to extract annotations
      information, here the feature list.
  */
  static AnnotationsClient provider;

  /** working objects. */
  static Feature ft;

  /** Constructor. */
  public sample1() {
    provider=new AnnotationsClient();
  }

  /** Main method.
      The accession number is transmited as
args[0].
  */
  public static void main(String[] args) {
    // get the feature list for the sequence
    FeatureList featlist=
      provider.getFeatureList(args[0]);

    // note that FeatureList inherits
    // java.util.Vector
```

```
    Enumeration e=featurelist.elements();
    while (e.hasMoreElements()) {
      ft=(Feature) e.nextElement();
      System.out.println(ft.label() +
        " begins at " + ft.begin() +
        " and ends at " + ft.end());
      if (ft.label().equals("V-REGION")) {
        System.out.println("translated by: "
          + ft.protein());
      }
    }
  }
}
```

Application Developments Using IMGT/LIGM-DB API. Because of security reasons, it is not possible, for a server other than IMGT server, to distribute Java Applets which use IMGT/LIGM-DB API classes. But one can develop a main programme running on his own machine to access LIGM-DB data and make various computing with them. Figure 7 shows a "four tier" architecture used by a remote WWW server which acts as a gateway to access LIGM-DB data and distributes them to WWW clients.

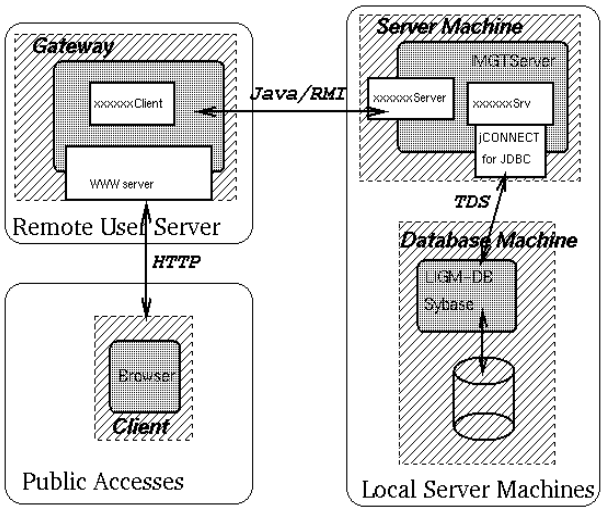

Figure 7: "Four tier" architeture using IMGT/LIGM-DB API

IMGT/LIGM-DB API is available from the IMGT server home page as a Java ARchive file. Users have to be registered to be authorised to download the API. However the registration is free. The currently flatfile format used for sequence data distribution is too rigid to easily include new information especially for specialised databases. Scientists are going to use new encoding to transmit sequence related data. Let us note the ASN.1 format promoted by NCBI and used by Entrez, and the SGML based format proposed by the Weismann Institute GAMBIT (see Appendix 6) project. Both propose a tagged text which structures the information by a hierarchical set of data. IMGT project is looking about these two approaches.

Other tools exist to dispatch data without loosing the structure of the information. The main one is the CORBA architecture. However we have chosen the Java "Remote Method Invocation" which is very similar to CORBA (see Appendix 7) by the architecture, but easier to implement. The use of Java/RMI requires a "RMI server" always listening for a remote request. That is, for us, two "deamons" are necessary, one is the HTTP server Apache, the other is the Java/RMI server. It is possible to distribute objects using the Java serialization technique. After being serialized, objects can be transmitted using HTTP protocol like images or sounds. We are studing this way too.

Conclusion

LIGM-DB is the most developed application of IMGT whose goals are to establish a common data access to all immunogenetics data, including nucleotide and protein sequences, oligonucleotide primers, gene maps and other genetic data of Ig, TcR and MHC molecules, from all species, and to provide a graphical user friendly data access. LIGM-DB experience in data modelling, data annotation, data control and distribution will be used for MHC/HLA-DB and PRIMER-DB, the two other IMGT databases in development. By its high quality data and easy data distribution, IMGT has important implications in medical research (repertoire in autoimmune diseases, AIDS, leukemias, lymphomas), therapeutical approaches (antibody engineering), genome diversity and genome evolution studies.

Acknowledgments

We thank Chantal Busin (LIGM) for the IMGT WWW interface design, Valérie Barbié (EUROGENTEC) for the development of IMGT/PRIMER-DB, Géraldine Folch, Nathalie Pallarès, Manuel Ruiz and Dominique Scaviner for LIGM-DB annotations. IMGT is funded by European Union's BIOMED1 and BIOTECH programmes, the CNRS (Centre National de la Recherche Scientifique) and MENRT (Ministère de l'Education Nationale, de la Recherche et de la Technologie). Subventions have been received from ARC (Association de Recherche sur le Cancer), ARP (Association de Recherche sur la Polyarthrite), FRM (Fondation pour la Recherche Médicale) and the Région Languedoc-Roussillon.

The database is developed by LIGM (Montpellier, France), in collaboration with EMBL-EBI (Hinxton, UK), ICRF (Oxford, UK), IFG (Köln, Germany), BPRC

(Rijswijk, The Netherlands) and EUROGENTEC (Seraing, Belgium).

Appendices: Internet References

1. Firth, C. Data quality in practice: experience from the frontline. IQ'96, URL http://sunflower.singnet.com.sg/~cfirth/dataquality
2. DISA. DoD Guidelines on Data Quality Management, URL http://sunflower.singnet.com.sg/~cfirth/dataquality
3. Segev A. On Information Quality and the WWW Impact A Position Paper, URL http://sunflower.singnet.com.sg/~cfirth/dataquality
4. Java. http://java.sun.com
5. Entrez, URL http://www3.ncbi.nlm.nih.gov/Entrez
6. Glusman G., Prilusky J. and Lancet D., Introducing GAMBIT, the Genome Annotation Markup Language, ISMB'97 Poster, http://bioinformatics.weizmann.ac.il/gambit
7. CORBA. Common Object Request Broker Architecture, URL http://www.acl.lanl.gov/CORBA. See also the EMBL-CORBA Web Poster, URL http://industry.ebi.ac.uk/~slidel/embl-corba

References

Benson, D.A.; Boguski, M.S.; Lipman, D.J.; Ostell, J.; and Ouellette, B.F.F. 1998. GenBank.. *Nucleic Acids Research* 25:1-7

Etzold, T.; Ulyanov, A.;.and Argos, P. 1996. SRS: information retrieval system for molecular biology data banks. *Methods Enzymol* 266:114-128

Honjo, T; and Alt, F.W. eds. 1995 *Immunoglobulin genes* . Academic Press.: 3-443

Lefranc, M.-P. 1990. Organization of the human T-cell receptor genes. *Eur. Cytokine Network* 1: 121-130

Lefranc, M-P. 1997. Unique database numbering system for immunogenetic analysis. *Immunology Today* 18:509

Lefranc, M.-P.; Giudicelli, V.; Busin, C.; Bodmer, J.; Müller, W.; Bontrop, R.; Lemaitre, M.; Malik, A.; and Chaume, D. 1998. IMGT, the international ImMunoGeneTics database. *Nucleic Acids Research* 26:297-303

Stoesser, G.; Moseley, M.A.; Sleep, J.; McGowran, M.; Garcia-Pastor, M.; and Sterk, P. 1998. The EMBL Nucleic Sequence Database. *Nucleic Acids Research* 26:8-15

Tateno, Y.; Fukami-Kobayashi, K.; Miyasaki, S.; Sugawara, H.; and Gojobori, T. 1998. DNA Data Bank of Japan at work on genome sequence data. *Nucleic Acids Research* 25:16-20

Address for correspondence
Professor Marie-Paule Lefranc
Laboratoire d'ImmunoGénétique Moléculaire, LIGM, UMR 5535 (CNRS, Université Montpellier II),
1919 route de Mende,
34293 Montpellier Cedex 5, France,
Telephone: +33 (0)4 67 61 36 34
Fax: +33 (0)4 67 04 02 31
E-mail: lefranc@ligm.crbm.cnrs-mop.fr,