UNIVERSITY OF OSLO
Department of Informatics

International Conference on Intelligent Systems for Molecular Biology(ISMB)
Vienna, Austria:
(July 21-25, 2007) – Poster

Research Report No: 399

Sabry Razick

Ian M Donaldson

ISBN :82-7368-360-5

September 2010

Design and prototype of a system to integrate and visualize biological interaction data.

Sabry Razick,[1,2] and Ian M Donaldson [1,3]
[1]The Biotechnology Centre of Oslo, University of Oslo, P.O. Box 1125 Blindern, 0317 Oslo, Norway
[2] Biomedical Research Group, Department of Informatics, University of Oslo, P.O. Box 1080 Blindern, 0316 Oslo, Norway
[3] Department for Molecular Biosciences, University of Oslo, P.O. Box 1041 Blindern, 0316 Oslo, Norway

Biomolecular interaction data is an increasingly important bioinformatics dataset used to examine biological systems. However, these data are spread across multiple databases and expressed in disparate data structures and formats. A prerequisite for working with these data would be consolidation into a single non-redundant updated repository. An initial design and prototype for consolidating and visualizing interaction data will be presented in this poster with an especial emphasis on providing scalable and reliable web-services as part of the solution.

The central point of operation is a data warehouse with numerous parsers retrieving updated information from existing data sources. We have designed parsers for PSI-MI 1.0, PSI-MI 2.5 XML files and tab delimited text files. The parsers for XML files are built using an event-driven pull-parsing API which gives them the ability to handle very large files.

A local application interface provides access to this data warehouse for local programmers, java servlets and web services. The usage of various operating systems and programming languages by the intended clients were considered when designing the system. Therefore, platform independent protocols were used. Moreover, multiple implementations of hosting the services were considered with respect to their ability to handle large data sets reliably in a stateful manner. Using these web-services, a tool was developed to retrieve and present interaction data visually. The approach taken was to construct modules as plugins to existing molecular visualization software.

This poster was published in the ISMB (International Conference on Intelligent Systems for Molecular Biology) conference 2007
(http://www.iscb.org/cms_addon/conferences/ismbeccb2007/)

Data warehouse for molecular interaction data

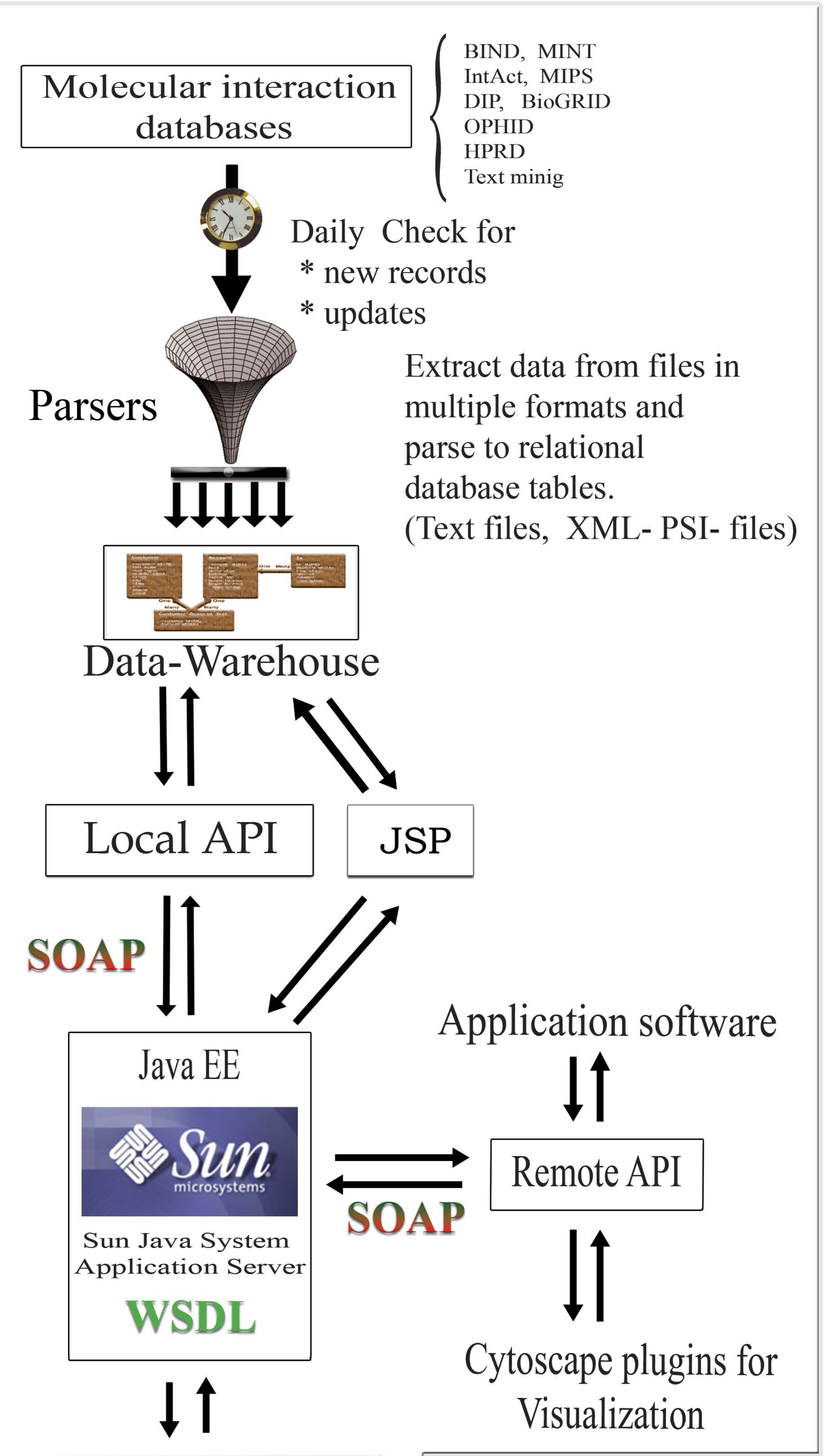
Sabry Razick and Ian Donaldson

Biotechnology Centre of Oslo
University of Oslo, Norway
Correspondence : ian.donaldson@biotek.uio.no

BiO
www.biotek.uio.no

Molecular interaction data is still maintained in multiple data warehouses. A prerequisite for working with these data is consolidation. A single non redundant updated repository can provide access to both local and remote users. We present here a preliminary design for such a warehouse. Redundancy is addressed by using a redundant object group (ROG). The algorithm is explained using IntAct interaction data as the example dataset. The design provides for access to these data in a very flexible manner, including FTP, direct web interface and web services based on the SOAP protocol. This will allow users the freedom of choosing there own operating system and programming language to access these data

Data flow


Molecular interaction databases

BIND, MINT
IntAct, MIPS
DIP, BioGRID
OPHID
HPRD
Text minig

Daily Check for
* new records
* updates

Parsers

Extract data from files in multiple formats and parse to relational database tables.
(Text files, XML- PSI- files)

Data-Warehouse

Local API

JSP

SOAP

Java EE

Sun Java System Application Server

WSDL

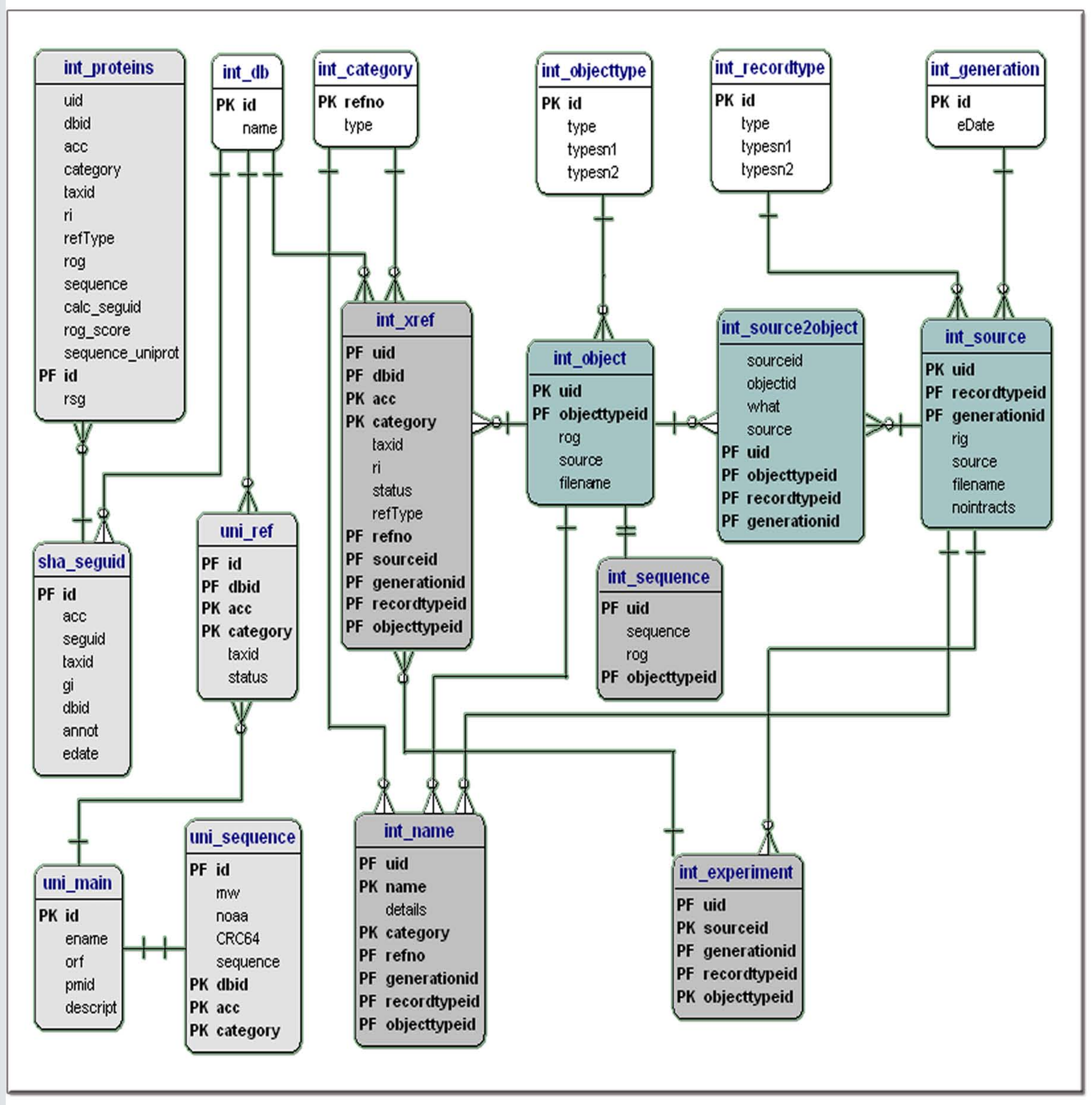
SOAP

Application software

Remote API

Cytoscape plugins for Visualization

Relational database design was used to implement the warehouse model. The database is currently hosted on MySQL (Ver 14.12 Distrib 5.0.37)


int_proteins: uid, dbid, acc, category, taxid, ri, refType, rog, sequence, calc_seguid, rog_score, sequence_uniprot, PF id, rsg

int_db: PK id, name

int_category: PK refno, type

int_objecttype: PK id, type, typesn1, typesn2

int_recordtype: PK id, type, typesn1, typesn2

int_generation: PK id, eDate

int_xref: PF uid, PF dbid, PK acc, PK category, taxid, ri, status, refType, PF refno, PF sourceid, PF generationid, PF recordtypeid, PF objecttypeid

int_object: PK uid, PF objecttypeid, rog, source, filename

int_source2object: sourceid, objectid, what, source, PF uid, PF objecttypeid, PF recordtypeid, PF generationid

int_source: PK uid, PF recordtypeid, PF generationid, rig, source, filename, nointracts

sha_seguid: PF id, acc, seguid, taxid, gi, dbid, annot, edate

uni_ref: PF id, PF dbid, PK acc, PK category, taxid, status

int_sequence: PF uid, sequence, rog, PF objecttypeid

uni_main: PK id, ename, orf, pmid, descript

uni_sequence: PF id, mw, noaa, CRC64, sequence, PK dbid, PK acc, PK category

int_name: PF uid, PK name, details, PK category, PF refno, PF generationid, PF recordtypeid, PF objecttypeid

int_experiment: PF uid, PK sourceid, PF generationid, PF recordtypeid, PK objecttypeid

Assigning interactors to a ROG

ROG (Redundant Object Group): All members of a ROG have an identical sequence and taxon. This logic flow describes how interactors found in interaction records are mapped to a ROG using accession number, NCBI taxonomic identifier and sequence.

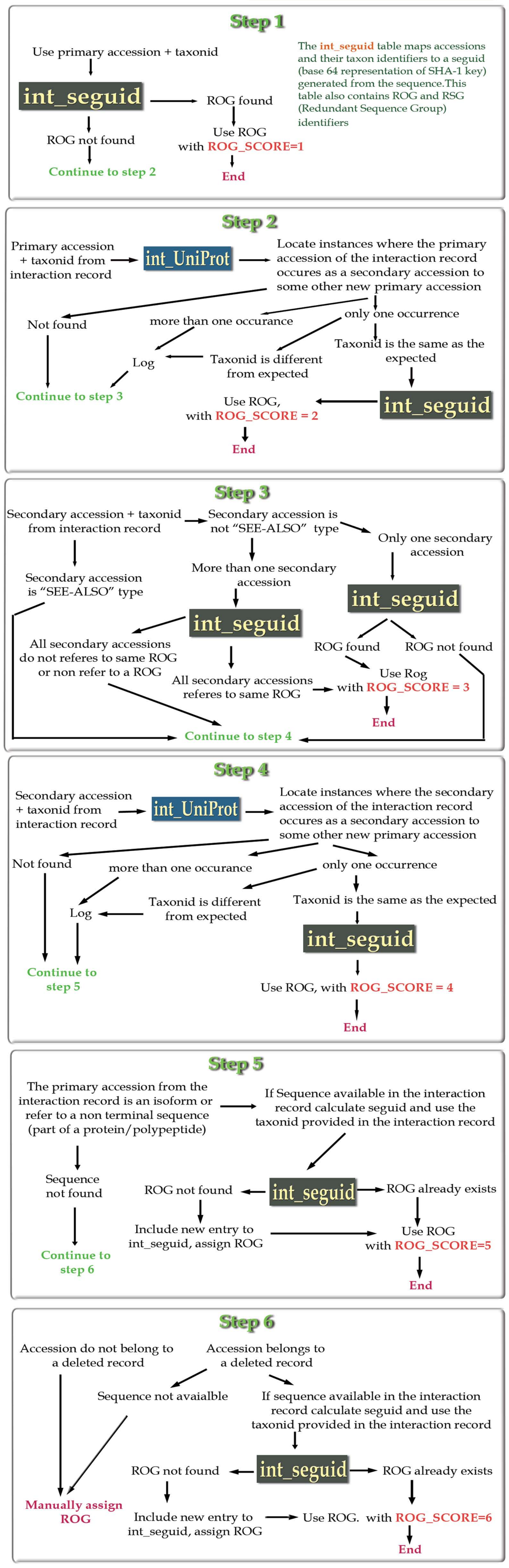
PSI-MI standard XML file.

<interactorlist>

</interactorlist>

→

Interactor details
* Type (e.g protein)
* Primary and secondary accession
* Source of accessions (e.g UniProt)
* Reference type of the accessions (e.g. identity)
* The NCBI taxonid (e.g 9606)
* Sequence


Step 1

Use primary accession + taxonid → int_seguid → ROG found → Use ROG with ROG_SCORE=1 → End

int_seguid → ROG not found → Continue to step 2

The int_seguid table maps accessions and their taxon identifiers to a seguid (base 64 representation of SHA-1 key) generated from the sequence.This table also contains ROG and RSG (Redundant Sequence Group) identifiers

Step 2

Primary accession + taxonid from interaction record → int_UniProt → Locate instances where the primary accession of the interaction record occures as a secondary accession to some other new primary accession

Not found → Continue to step 3

more than one occurance → Log → Continue to step 3

only one occurrence → Taxonid is different from expected → Log

only one occurrence → Taxonid is the same as the expected → int_seguid → Use ROG, with ROG_SCORE = 2 → End

Step 3

Secondary accession + taxonid from interaction record → Secondary accession is not "SEE-ALSO" type

Secondary accession + taxonid from interaction record → Secondary accession is "SEE-ALSO" type → Continue to step 4

Secondary accession is not "SEE-ALSO" type → More than one secondary accession → int_seguid

int_seguid → All secondary accessions do not referes to same ROG or non refer to a ROG → Continue to step 4

int_seguid → All secondary accessions referes to same ROG → Use Rog with ROG_SCORE = 3 → End

Secondary accession is not "SEE-ALSO" type → Only one secondary accession → int_seguid

int_seguid → ROG found → Use Rog with ROG_SCORE = 3 → End

int_seguid → ROG not found → Continue to step 4

Step 4

Secondary accession + taxonid from interaction record → int_UniProt → Locate instances where the secondary accession of the interaction record occures as a secondary accession to some other new primary accession

Not found → Continue to step 5

more than one occurance → Log → Continue to step 5

only one occurrence → Taxonid is different from expected → Log

only one occurrence → Taxonid is the same as the expected → int_seguid → Use ROG, with ROG_SCORE = 4 → End

Step 5

The primary accession from the interaction record is an isoform or refer to a non terminal sequence (part of a protein/polypeptide) → If Sequence available in the interaction record calculate seguid and use the taxonid provided in the interaction record → int_seguid

The primary accession from the interaction record is an isoform or refer to a non terminal sequence (part of a protein/polypeptide) → Sequence not found → Continue to step 6

int_seguid → ROG not found → Include new entry to int_seguid, assign ROG → Use ROG with ROG_SCORE=5 → End

int_seguid → ROG already exists → Use ROG with ROG_SCORE=5 → End

Step 6

Accession do not belong to a deleted record → Manually assign ROG

Accession belongs to a deleted record → Sequence not avaialble → Manually assign ROG

Accession belongs to a deleted record → If sequence available in the interaction record calculate seguid and use the taxonid provided in the interaction record → int_seguid

int_seguid → ROG not found → Include new entry to int_seguid, assign ROG → Use ROG. with ROG_SCORE=6 → End

int_seguid → ROG already exists → Use ROG. with ROG_SCORE=6 → End

Data warehouse

The warehouse model presented here will facilitate the consolidation of interaction data from various sources and provide researchers a portal for all available interactions.
This data warehouse is a collection of data from disparate sources and resolves:

1. Data structure dissimilarities
2. Redundant identifiers
3. Redundant interactions

Problems faced during assigning ROG.

Problems faced during assigning ROG

1. The initiator M problem
2. Deleted records : when the protein referenced in the interaction record has been retired
3. Isoform accessions
4. Chain accessions
5. Accession numbers unique to the interaction record source
6. Retired accessions. Replaced with new version
7. References to non terminal proteins (parts of proteins)
8. Not providing the taxonomic identifier in the interaction record. Synthetic proteins
9. Errors in the record file (e.g. leading and trailing spaces)

Solutions:

1. Use sequence from latest UniProt records
2. Use deleted record list from UniProt
3. 4, 5 and 7 use sequence provided with the interaction record
6. Use Entrez ID_1 fetch client to find latest version of a sequence
8 not solved
9 solved during parsing

The " initiator M" problem

The decision by UniProt to reformat the sequences corresponding to the precursor form of the protein was introduced with Release 52.0 of UniProtKB (06-March-2007). This added an initiator methionine(M) to some sequences. Thus beaking their identity with the sequences available in earlier interaction data files and earlier UniProt/Swiss-Prot records. This anomaly was resolved by using the sequences from the latest UniProt records instead of the sequence available with interaction records for calculating SEGUID and assigning ROG.

Mapping protein interactors from intact to a ROG:

Score	No of proteins	definition
1	69651	Primary accession from a interaction record used
2	66	New accession from UniProt used(retrieved using primary from interaction record)
3	38	Secondary accession from a interaction record used
4	445	New accession from UniProt used(retrieved using secondary from interaction record)
5	3559	Sequences from interaction record was used in cases where the interactor accession pointed to a isoform, PRO form, chain form or deleted UniProt records

Total protein Interators = 74128
Total proteins assigned to ROGs = 74059 (99.91 %)

References

- **IntAct – Open Source Resource for Molecular Interaction Data.**
  S. Kerrien; Y. Alam-Faruque; B. Aranda; I. Bancarz; A. Bridge; C. Derow; E. Dimmer; M. Feuermann; A. Friedrichsen; R. Huntley; C. Kohler; J. Khadake; C. Leroy; A. Liban; C. Lieftink; L. Montecchi-Palazzi; S. Orchard; J. Risse; K. Robbe; B. Roechert; D. Thorneycroft; Y. Zhang; R. Apweiler; H. Hermjakob. Nucleic Acids Research 2006; doi: 10.1093/nar/gkl958
- **A database of unique protein sequence identifiers for proteome studies.**
  Babnigg G, Giometti CS. Proteomics. 2006 Aug;6(16):4514-22.
- **UniProt user manual**
  UniProt Knowledgebase, Swiss-Prot Protein Knowledgebase, TrEMBL Protein Database. Release 11.3 of 10-Jul-2007 . http://au.expasy.org/sprot/userman.html
- **ID1_FETCH - the ID1 and Entrez2 client help files.**
  http://www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=toolkit.section.ch_demo.id1_fetch.html