



The geneticXchange discoveryHub Frequently Asked Questions

What is the DiscoveryHub?

discoveryHub is a software product based on a mature, patented technology more than eight years in the making. The core discoveryHub technology is based on a mathematical principle called 'nested relational calculus'. Nested relational calculus is similar to the theory behind most of the relational database management systems (RDBMS) on the market today with one major difference: nested relational calculus (NRC) can deal inherently with heterogeneous, hierarchical structured data. discoveryHub was built from the ground up utilizing NRC as its underlying principle, so its query processing engine can uniquely allow users to select, join and in some cases update data that resides in complex and often disparate nested structures.

How does discoveryHub Work?

Level 1: QUERY / OPTIMIZATION -----	→	discoveryHub SQL Query Engine
Level 2: MIDDLEWARE / DATA ACCESS -----	→	Wrappers
Level 3: PHYSICAL DATA STORAGE	→	3rd party databases, algorithms, search mechanisms

discoveryHub handles both Level 1 and Level 2 of the above diagram. Level 1 is the query and optimization layer that would be similar to ones typically be found in a RDBMS system. Level 2 is “pre-built” middleware (wrappers in discoveryHub terminology) that allow access to Level 3 - the proprietary data structures found in many scientific data sources, both internal and external.

discoveryHub never needs to store data internally, but will leverage the retrieval/update mechanisms that already exist to 3rd party databases, so it’s completely transparent to the user. This makes discoveryHub uniquely and completely data source independent and able to deal with even the most complex heterogeneous data environments in an automated, scalable fashion.

What Does this Mean?

It means that discoveryHub is like having the 'top' part of a relational database (the query/join bit) without the 'bottom' part (the file storage structures) and instead we use the ability to access external data sources as if they were local to discoveryHub - via our wrapper middleware. This enables discoveryHub to access a multitude of data sources, including internal/external flat files, web sites, and databases, so users can see and query multiple, complex data sources as if they were a single, homogenous “virtual” database.

Why is this Important?

With the ever increasing volume and complexity of scientific data in the drug discovery space, discoveryHub simplifies and automates the access and integration mechanisms in a scalable, efficient way. discoveryHub already provides access to 60+ popular data sources in use by drug discovery companies today (including protein databases, genetic databases, domain search algorithms and



patent searches), and these are only a very small subset of the sources available to discoveryHub. So, in one simple move, discoveryHub enables users to query a protein database, for example, like pdb, for a single protein or a set of proteins, blast the resulting sequences and/or take the results and check them against a relevant patent database(s). This kind of search could be done using a very short SQL query and easily integrated into internal processes.

Who Uses discoveryHub?

The typical user of discoveryHub would be a bioinformatic or IT professional who needs to access scientific data sources without the time or budget to spend months programming costly hand-coded interfaces in Perl or C++, etc.

What Skills are Required?

There are many ways to access discoveryHub, and most of these are completely transparent to the end user - in the same way that an Oracle database is transparent to a majority of its end users. The user would see a front end visualization system or even a web page (like Genbank from NCBI). Underneath this, the access mechanisms in discoveryHub are similar to any other API. discoveryHub API's are compatible with C, C++, Perl, Java, JDBC, ODBC, Enterprise Java Beans (EJB), Simple Object Access Protocol (SOAP) and via the intra/internet using our Web Proxy Server. Each of these mechanisms accepts SQL statements - which can be easily authored using our discoveryBuilder query visualization interface.

What are the Technical Benefits of discoveryHub?

discoveryHub enables users to write standard SQL queries against various, dispersed data sources, and drill into and expand the hierarchies within the data. This enables the user to take data from different sources in completely different structures and create a common view across them – i.e. data from two protein databases (say, Swissprot and Entrez) and create a view that represents a cross-sectional analysis against them. The target for the query could simply be an application, like a visualization interface, or in many cases an internal relational database to store the results of the complex queries. discoveryHub also enables the target database to maintain the nested nature of the original source data – i.e. no flattening (without the need to create a complex N-table data model). geneticXchange has developed a method to quickly populate any relational database with hierarchical data in a single table in a single stroke without losing any vital structural information. Additionally, querying the nested data out of the relational table is as simple as writing a SQL query.

Multiple Cross Database Queries

discoveryHub enables the query writer to create queries and joins across a multitude of underlying databases inherently, for example without the need for heavy gateway technology. discoveryHub gives the ability to do inner joins, outer joins, unions, etc. across heterogeneous databases. Many of geneticXchange's clients wish to take data from files or websites such as LocusLink, compare them to internal databases, such as genetic data stored in (and not inclusively) Oracle and as a result 'join' the information together and submit the join as a whole to a process running remotely on a service providers machine across a network. The result should then be written back to the calling application for visualization.



With the discoveryHub technology, this entire process is automated and scalable and consists of a single (or very few) SQL query to access each data source as it were a local table! This results in simple, standard queries across a multitude of technologies in one easy, maintainable step.

Relational Technology / Normalization

Normalization of Relational Databases is a prominent aspect of relational database theory. It addresses how data ought to be organized within a database in order to make the database as compact and easy to manage as possible and to ensure that it produces consistent results. Normalization rules provide guidelines for defining the schema (design) of a relational database. Simply put, the rules specify how a database should be divided into tables and how the tables should be linked together.

There are two major objectives of normalization:

1. Minimize the duplication of data
2. Minimize the number of attributes that must be updated when changes are made to the database, thereby making maintenance of the data easier and reducing the possibility of error.

Typically, normalized design databases are then de-normalized at the physical design stage to facilitate real world experiences (i.e. performance and practicality). For example, if the relational database does not support non-atomic attributes, and chooses to de-normalize for performance, then the user must create multiple columns (e.g. protein 1, Protein 2, etc.) to represent the data structure.

Use of Nested Relational Calculus to solve Relational Limitations

The above examples can make it very difficult to create simple, meaningful SQL statements to retrieve accurate results from relational databases. In a nested relational database, however, the ability to support nested tables allows the user to create a single protein column, or a group of related columns (protein, feature, annotations) that is nested or multi-valued. The nested relational SQL can then simply and easily retrieve the correct results, in a high-performance way.

How does discoveryHub Enhance Relational Database Technology?

discoveryHub has the unique ability to enable RDBMS technology to handle complex object data types in a single table structure, thereby transforming a typical 2-dimensional database into a highly efficient object store. discoveryHub allows the database designer to store and index nested structures within a simple relational environment by making use of large object columns. This means that a single table can become a cascading object structure and can also be adequately indexed to cope with real life retrieval scenarios.

Take an example:

- Protein (Accession, Sequence, title)
- Feature (...source...)
 - o Annotation (...)
- Lineage (...)



This is a very typical (and also one of the simplest) examples of how a protein structure is built up in a database like Genbank from the NCBI. In a relational environment, the structure would typically consist of a minimum of 42 dimensional tables. However, discoveryHub can store this structure in a single table, as below.

- Protein (Columns: Accession, Sequence, Sequence index, title, source, <complex object: *Feature, <*Annotation>, <*Lineage>>

The top level columns in this example have been exposed to the relational database for retrieval purposes and indexing; the complex object is stored in the relational database, unknown to the relational metadata. discoveryHub has the unique ability to store this complex structure and more importantly to quickly and efficiently explode the complex object back into its original nested data form, enabling the user to select ANY column, level or structure within the object.

geneticXchange believes that discoveryHub is a ground-breaking technology that uniquely enables companies to store and index large, complex objects in standard relational databases, thus transforming the effectiveness of such technologies for drug discovery purposes.

What are the Commercial Benefits of discoveryHub?

The majority of the commercial benefits of discoveryHub come from the increased ability to perform critical tasks quicker. So for example, writing interfaces to bioinformatic data sources can be done in a single command, increasing the productivity of a smaller team of IT professionals. Also, as discoveryHub provides a single point of data interfacing, it helps to mitigate risk, as the research team can all use the same information via a standardized user interface. Our clients have found that the discoveryHub approach to bioinformatic data sourcing can save many months of development time in producing vital results and statistics.

Why Should I Implement discoveryHub?

discoveryHub enables organizations to make significant productivity and performance improvements to their internal processes by automating critical data access and integration tasks. discoveryHub also reduces IT costs by providing a proven, industrial-strength solution that can easily scale as data sources - and business needs - change. In addition, discoveryHub's open standards architecture enables clients to be assured that all applications built on the discoveryHub technology will support future infrastructure changes, as well as the inherent, organic growth of information in the biotechnology field.