JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

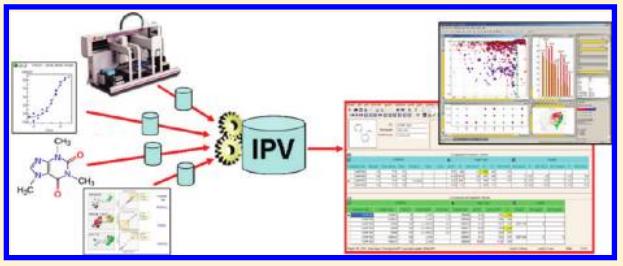
pubs.acs.org/jcim

Integrated Project Views: Decision Support Platform for Drug Discovery Project Teams

Eric J. Baede, Ernest den Bekker, Jan-Willem Boiten,* Deborah Cronin, Rob van Gammeren, and Jacob de Vlieg

Discovery Informatics, Molecular Design and Informatics Department, MSD, Molenstraat 110, 5342 CC Oss, The Netherlands

ABSTRACT: Drug discovery teams continuously have to decide which compounds to progress and which experiments to perform next, but the data required to make informed decisions is often scattered, inaccessible, or inconsistent. In particular, data tend to be stored and represented in a compound-centric or assay-centric manner rather than project-centric as often needed for effective use in drug discovery teams. The Integrated Project Views (IPV) system has been created to fill this gap; it integrates and consolidates data from various sources in a project-oriented manner. Its automatic gathering and updating of project data not only ensures that the information is comprehensive and available on a timely basis, but also improves the data consistency. Due to the lack of suitable off-the-shelf solutions, we were prompted to develop custom functionality and algorithms geared specifically to our drug discovery decision making process. In 10 years of usage, the resulting IPV application has become very well-accepted and appreciated, which is perhaps best evidenced by the observation that standalone Excel spreadsheets are largely eliminated from project team meetings.



INTRODUCTION

Over the last two decades, traditional approaches for drug discovery within large pharmaceutical companies have yielded disappointing success rates: research and development (R&D) costs have been rising steeply whereas output, i.e. the number of new drugs registered, has only declined.[1] Big Pharma has initially responded to this trend by increasing the productivity in early drug discovery, in particular through the introduction of automation in the chemistry (parallel synthesis) and biology laboratories (high-throughput screening). This approach, commonly labeled as "more shots on goal", did not make a notable difference to the fundamental underlying problem, i.e. still too many compounds fail late in the R&D process leading to prohibitively high costs. While there is no silver bullet to reduce this late stage attrition, a clear starting point is the back-translation of knowledge and insights from clinical and other development studies into early drug discovery. As a first step, drug discovery scientists have embraced various tests and techniques (e.g., in the domains of Drug Metabolism and Pharmacokinetics (DMPK) and Toxicology) traditionally only applied in later stage R&D. This trend added further complexity to the multiple-parameter optimization process of developing potent, stable, and safe compounds as schematically depicted in Figure 1.

Within Organon (now part of Merck & Co, Inc.), we advocate that a rigorously data-driven approach throughout drug discovery is required in order to improve the quality of drug candidates with the ultimate aim to reduce late-stage attrition.[2] A critical step in this approach involves the creation of a thorough structure−activity relationship (SAR) analysis in order to make well-informed decisions about what new compounds to design or which tested compounds to progress to the next step in the research workflow. Moreover, these decisions must be taken in a timely fashion under pressure of the deadlines for the synthesis and testing cycles in the chemistry and pharmacology laboratories. Clearly, information technology could play a pivotal role in facilitating the multiparameter optimization.

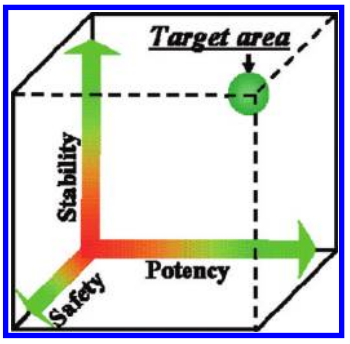


Figure 1. Schematic overview of the multiparameter optimization for drug design. In reality the number of dimensions involved is significantly higher.

Already since the 1980s, various disciplines in Organon's Discovery Research (e.g., chemistry and pharmacology) applied computerized systems to manage their own data processing, storage, and retrieval. However, the lack of tools establishing

Received: June 8, 2011

dx.doi.org/10.1021/ci200253g | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

the highly needed integration[3,4] between the individual data silos forced the multidisciplinary discovery project teams to waste time with the tedious effort of manually consolidating data from the relevant data sources into one repository, usually an Excel[5] spreadsheet or ISIS/Base[6] database: a laborious process, prone to error, regularly resulting in different versions used, making team meetings inefficient and ineffective at the same time. Moreover, this focus of drug discovery project teams on their own data management solutions led to a declining compliance with the policy of data entry into the corporate experimental results databases, as the teams themselves did not directly benefit from the data entry effort.

In the late 1990s, it was clear from the observations outlined above that this silo-based approach to data handling was not sustainable with the growing data volumes in a modern drug discovery organization. Organon and various other companies in the life sciences business have been addressing this issue with dedicated applications, in particular Johnson and Johnson with the ABCD system,[7] Amgen with ADAAPT,[8] ArQule with ArQiologist,[9] and Actelion with OSIRIS.[10] In this publication we will present and discuss the main concepts of Organon's project-centric data integration environment named IPV—the acronym for Integrated Project Views.

At the drawing board we started with identifying the key questions in the drug discovery decision making process which had to be supported by IPV (see also Table 1). Basically, all these queries boil down to two overarching questions:

1. What do I know about compounds previously made?
2. How can we make use of this information to design the next generation of compounds?

Table 1. Typical Drug Discovery Questions Supported by IPV

What are the latest data?
Can I plan a new experiment?
Which compounds match this pharmacological profile?
What are the underlying data for this conclusion?
What can I do to make compound X better?
Why is compound X more active than compound Y?
What are the critical issues in my compound series?

In order to deal with these questions effectively IPV had to become the single data sharing environment for drug discovery teams taking care of all data integration and processing, allowing every project team member to use the same up-to-date information ("to sing from the same hymn sheet"). Moreover, this common data backbone had to be usable for the cheminformatics and molecular modeling experts in the team as well as for any other team member, as opposed to the traditional situation where cheminformatics and modeling groups are insufficiently embedded in the discovery research process and maintain their own data environments. The latter approach would easily result in duplication of efforts and misalignment between the data sets used by the project team.

From these prerequisites, it was clear that IPV had to be useful for scientists with different backgrounds, expertise levels, and scientific questions.

KEY REQUIREMENTS

With the aforementioned high level considerations as a starting point, we identified a series of high-level needs with the underlying key requirements for the application that eventually became IPV:

- *Integration of all data required by drug discovery projects to take decisions on compounds.* Data integration within a project had to be centered around the chemical structures with their related data (purity, origin, amount available, etc.) and the results of experiments performed with those structures: in vitro assay results, in vivo data, results of DMPK and early tox studies, and analytical data (structure identity as well as physicochemical determinations). These core data had to be supplemented with experimental results from earlier studies (e.g., from screening or from other projects) and with in silico predictions from simple physicochemical models.
- *Data presentation enabling systematic data comparison and hypothesis building.* This simple high-level need not only led to specific user interface requirements (a grid view allowing for side-by-side comparison of drug candidate compounds), but also to various complex technical data requirements. Biological results tend to be stored in a generic data model; consequently, data pivoting is needed to arrive at the grid view required for hypothesis building. Other consequential requirements were the ability to convert between different units of measure (e.g., moles per liter and milligrams per liter), and the ability to handle incomplete biological data as commonly indicated by operators (e.g., $EC_{50} > 10\ \mu M$).
- *Support for compound profiling.* First of all, compound profiles had to be supported by a generic and flexible query builder as well as Excel-like features such as sorting and filtering. In addition, IPV had to contain a rich set of conditional formatting options (colors, fonts, etc.) allowing scientists to highlight interesting or deviating data points.
- *A central collaboration environment for the project team.* Scientists should be able to annotate data points, highlight new experimental results, or even knockout data points with questionable quality (e.g., outliers). The usage of the same data environment by a diverse scientific audience also called for different views on the same data: a top level view for aggregated data with the option to drill down to lower level views, if more detail is required. The characteristics of complex data sets, e.g. as found in in vivo experiments, led to the requirement that the same data set may actually be aggregated in different manners depending on the issues considered important by the project team, e.g. aggregation over dose, individual animals, or administration vehicle. This was implemented in IPV using so-called "dividers" (vide infra).
- *Full ownership of the project's data by the project team.* As IPV is not the primary but a derived data source, project teams can really take ownership of "their" IPV project: they are free to make their own choices on which data to include, how to aggregate these into higher level conclusions, how to visualize the data, etc. Data in the corporate repositories remain unchanged and hence available for reloading and reanalysis in IPV. Maintenance of the project definition and data representation should have short turn-around times, either through self-service, or with limited assistance of a Discovery Informatics specialist. A user-friendly administration

dx.doi.org/10.1021/ci200253g | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

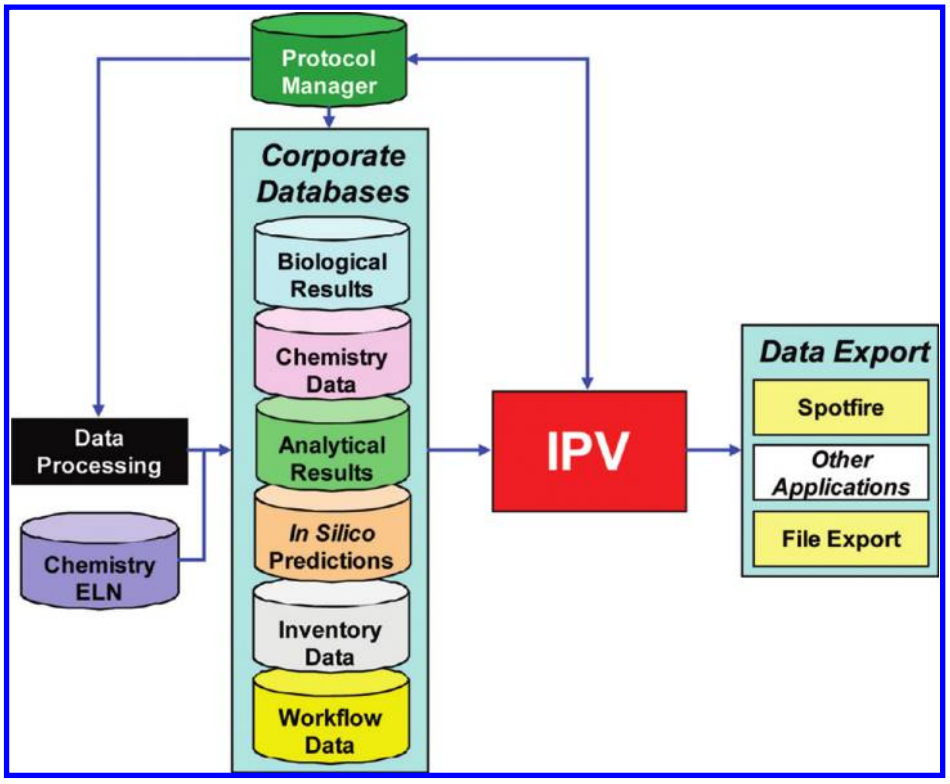


Figure 2. Position of IPV in the overall architecture; the corporate databases are filled by various data processing applications and a Chemistry ELN. IPV sources data from these corporate databases and can export data to several applications and file formats. A Protocol Manager system is used to check the consistent generation, storage, and usage of biological and analytical data.

tool was therefore needed in order to maintain data divisions, additions of new compounds and assays, etc.

- *An interactive environment stimulating the scientists' creativity.* The response times typically expected from an interactive application set high standards for application performance.

IMPLEMENTATION OVERVIEW

Build or Buy? At the time of the design of IPV, several commercially available data warehousing solutions were considered but were felt to be inappropriate to realize the IPV system. The main reasons for this conclusion were the following:

- Data warehouses at the time were heavily focused on integrating data from different technical sources (e.g., different databases, file systems, Excel sheets) and on cleansing the data from those sources. However, the data that were to be handled by IPV were already quite consistent, especially with regard to definition and values of key identifiers. Also, they were all stored in Oracle databases.
- Pivoting, aggregating, and querying of biological data added an extra level of complexity due to the use of mathematical operators (e.g., >) and varying units of measure. These were not dealt with properly by the standard data warehousing solutions.
- A key part of the data warehouse concept is a time-driven component in the way data are to be aggregated and viewed. However, there is usually no time-driven component in the way scientists approach drug discovery data.
- None of the retrieval tools that came with the standard data warehousing packages was capable of handling chemical structures, which was a key requirement for IPV.

Because of these considerations, it was decided to develop IPV in-house. Although no standard data warehouse products were used, over time several data warehousing concepts and techniques, e.g. staging, have been applied, especially to manage system performance.

General Setup. IPV is a retrieval tool integrating data from different sources (see Figure 2). Within IPV, discrete projects are defined, specifically tailored to the needs of the project teams involved. Each project is assigned a dedicated set of project-specific tables containing three levels of data (see also "Data Hierarchy"), which are maintained automatically. The contents of these levels can vary widely between different projects, depending on the nature and number of protocols used in the project. For performance reasons, the three levels are prebuilt and stored in three project-specific tables (the level tables) in a central Oracle database. The specific set up and layouts of the IPV project are driven and owned by the project team.

IPV consists of the following components (see Figure 3):

- IPV Explorer: the end-user application, used for searching and displaying the project data.
- Project Data Repository: the combination of database tables and stored procedures used by IPV Explorer, not only containing the prebuilt level tables with project data, but also providing drill-down access to detailed data from the source systems.

dx.doi.org/10.1021/ci200253g | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

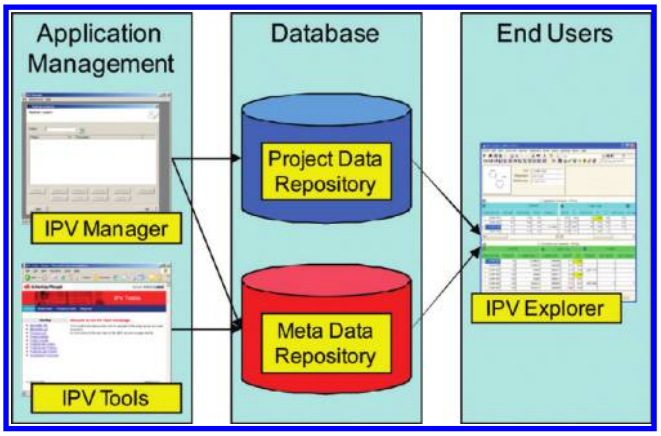


Figure 3. Components of the IPV system.

- IPV Manager: the application used by IPV application managers and power users to define and maintain the projects' definitions and to manage project building.
- Meta Data Repository: the combination of database tables and stored procedures used for storing project definition data, building the specific project level tables, and interacting closely with IPV Manager.
- IPV Tools: the web application used by the IPV application managers to monitor and prioritize background processes.

Data Level Concept. In IPV, the data are organized in three levels, defined per project, where in general:

- Level 3 contains basic results per sample per experiment, e.g. a single %-effect from a dose−response experiment.
- Level 2 contains summary results per sample per experiment, e.g. an EC_{50} calculated over a concentration-effect curve.
- Level 1 contains level 2 data aggregated per compound, e.g. the geometric mean of multiple EC_{50} values. Note: a compound can have multiple samples, and aggregation takes place over all samples and experiments of that compound.

The level concept allows the scientists to look at the data with the following "mind-set":

- Level 1: What's new? What's interesting? Compare activities.
- Level 2: Check the reliability of aggregated data at level 1.
- Level 3: What underlying basic results led to the summary results on level 2?

IPV Explorer. IPV Explorer is the end-user application. As the Excel look-and-feel was (and still is) very popular among scientists, the main component of IPV Explorer is a grid showing the data (see Figure 4).

The main features of IPV Explorer are the following:

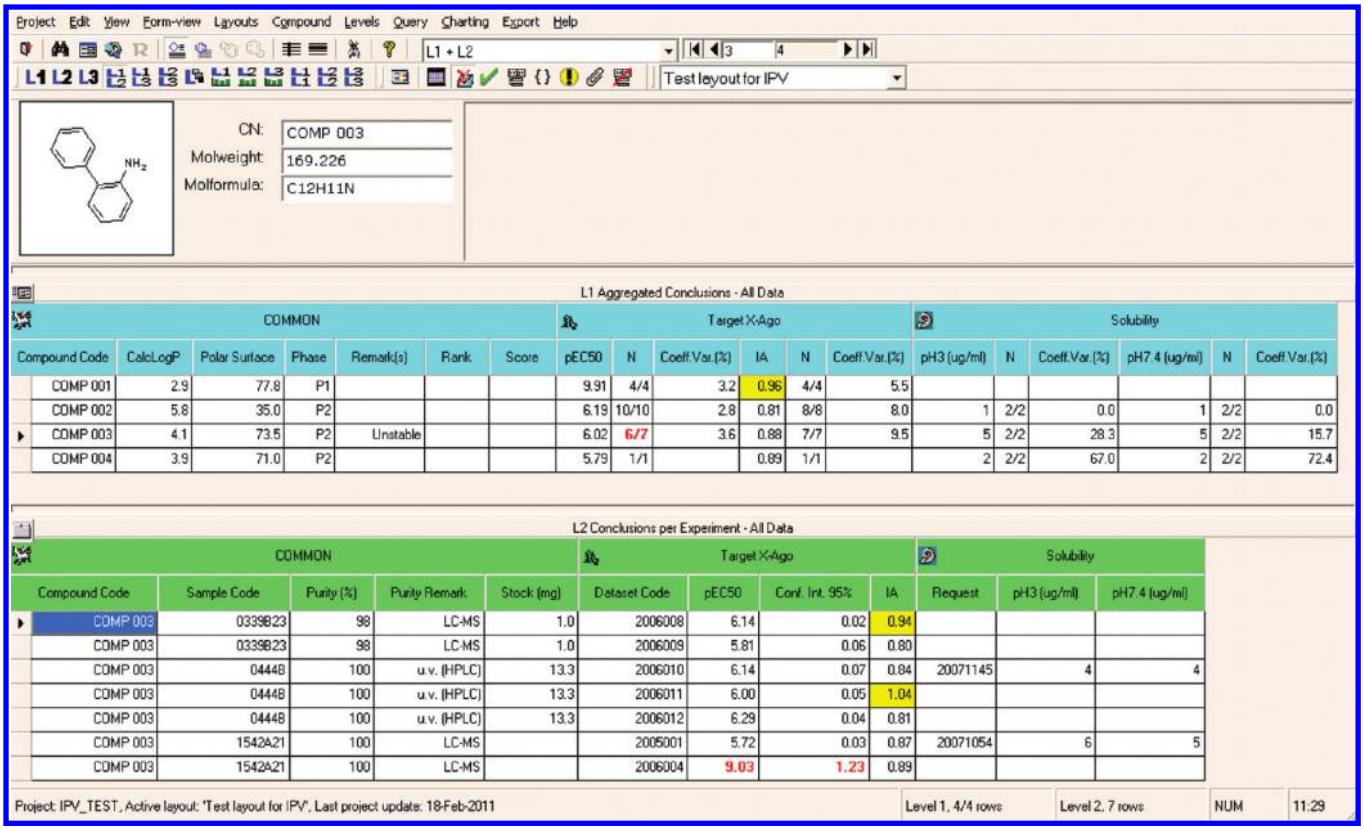


Figure 4. Example of the IPV Explorer main window. The data is shown hierarchically for levels 1 and 2. Each level is displayed in an Excel-like grid where vertically the compounds and samples are listed and horizontally the protocols and results. At level 1, several calculated properties (LogP and Polar Surface) and annotation fields (Phase and Remarks) are shown. The value highlighted in red on level 2 is an excluded value (e.g., an outlier) which translates to $N = 6/7$ on level 1 (also highlighted in red). The values highlighted in yellow on both levels represent an Intrinsic Activity with a value larger than 0.9. In the top section of the main window, the chemical structure and corresponding data are displayed for the compound currently selected at level 1.

dx.doi.org/10.1021/ci200253g | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Data Display. Data are shown for one project at the time in an Excel-like display (grid) for each of the three data levels. This grid enables multiple-column sorting of the project data, splitting the grid into different scroll regions, resizing of rows and columns, drag-and-drop reordering of columns, etc. The combination of these personal preferences is called a layout, which can be saved for reuse and made available to other users. As an alternative to the grid it is possible to show single compound-oriented views as often seen in commercial products like ISIS/Base,[6] Isentris Client,[11] and ChemCart.[12]

Apart from the biological assay data, so-called "related data" are available. These data, linked from other systems at runtime, are compound or sample related at level 1 or levels 2/3, respectively. Typically related data are calculated or predicted properties (e.g., logP, number of hydrogen acceptors/donors, and polar surface) and inventory data. The related data to be shown is defined per IPV project and per data level based on project team requirements.

Data Drill Down. Two grids for different levels of data can be shown simultaneously allowing drill down into the data hierarchy, e.g. to evaluate underlying level 2 data for the aggregated level 1 data (see also Figure 4). Those two levels are synchronized, i.e. when clicking on the grid for one level, the other grid will move to the same compound and protocol. A drill-down for a single data point is also possible, either for a level 1 data point to show the underlying level 2 data (without the need to open the grid for level 2 explicitly) or for a level 2 or 3 data point to view data in the underlying source system (e.g., the full set of experiment data of which the data point is part).

Another drill-down option is a compound-centric view of all chemical, biological, analytical, and inventory data that are available in the corporate research databases, not necessarily part of the IPV project.

Querying/Conditional Formatting. To allow for various levels of complexity IPV Explorer provides three query options:

- a simple form for the most common queries, mainly containing chemical and compound annotation fields;
- a form-based query, which allows querying on all fields in the currently used layout;
- an advanced query builder enabling the definition of compound profiles. It allows querying on any combination of fields in the project (from any level) using the Boolean operators AND, OR, and NOT.

All three query options include various ways of structure searching, such as exact, substructure, and similarity searches. Like layouts, query definitions can be saved for reuse and be made publicly available or kept private.

Furthermore, conditional formatting, i.e. changing font or coloring of values/boxes based on certain conditions, is available as an option to highlight specific data (e.g., all data added or modified since a certain date or since the last use of IPV Explorer).

Compound Annotation. For each compound in every project a set of fields is available to store annotations and status information. Fields available for annotation include compound status, phase, series, and class. These fields can be edited by project team members and are read-only for other users.

Data Exclusion. Team members can exclude level 2 data points for the aggregation to level 1, e.g. in the case of outliers or operators that obstruct averaging (see "level 1 aggregation"). This is visualized on level 1 by n/N data columns, indicating the number of values used for averaging (n) out of the total number available (N).

SYSTEM ARCHITECTURE

Client applications. Both IPV Explorer and IPV Manager are built in Microsoft Visual Basic 6.[13] In IPV Explorer MDL ISIS Desktop[14] is used to implement the chemical structure handling capabilities. Structure searches are handled by the MDL Structure Cartridge[15] within the database; for basic charting functionality ComponentOne Chart[16] was incopo-rated. The IPV Tools application was built in Oracle APEX (v3).[17]

IPV Explorer performs a great deal of client-side processing for data display and visualization. However, for data access and manipulation it relies heavily on the IPV database, e.g. on stored procedures for drill-down.

The main component of IPV Explorer is a grid showing the data. Because of the importance of this grid, a proof of concept was built with four popular commercially available grids[18−21] to select the one that would not only allow implementation of the most important features, but would also offer a better perceived performance by supporting asynchronous data fetching. As a result of this analysis, Infragistics Ultragrid[18] was chosen to be incorporated into IPV Explorer.

Database/server. The IPV database, containing the Project Data and Meta Data repositories, is implemented in Oracle[22] (starting with version 8i and over time has been stepwise upgraded to 10g currently). All server-side functionality is implemented inside the Oracle database using PL/SQL; there is no separate middle tier server used.

An important database task is the automatic creation, rebuild, and update of the project level tables. By default all IPV projects are automatically updated overnight. However this can be overruled:

- The automatic update can be switched off per project. This is particularly useful for large IPV projects where only occasionally new data are generated.
- Project users can request an ad hoc update, which is useful when important new project data are generated and needed immediately, e.g. right before a project meeting.

IMPLEMENTATION DETAILS

Data Hierarchy. In IPV, the data per project are organized in three levels. The data on levels 2 and 3 are extracted from the source systems, i.e. the corporate databases. The aggregation on level 1 is performed and stored by the IPV system, based on settings entered in IPV Manager for the specific project.

The project consists of various protocols, which in turn may be split into divisions. Each division contains one or more result types, and finally, a result type can have one or more columns.

Condition Handling. Conditions are independent variables in an experiment. These can be actual experimental variables (e.g., dose or route of administration) or variables of an administrative or technical nature (e.g., experiment code or replication number). Conditions drive the pivoting of data (see Figure 5) and can be used in IPV in two distinct ways (see Figure 6):

- As an extra column next to the corresponding results to add context to these data. This can be done as either a "protocol generic field" (condition is common for all

dx.doi.org/10.1021/ci200253g | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

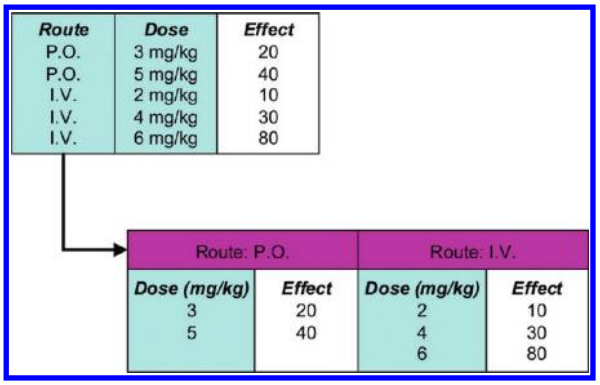


Figure 5. Example of data being pivoted. The data consist of one result type (% effect) with two conditions (route and dose). The condition route is used as a divider, and the condition dose as a division generic field.

divisions in the protocol) or a "division generic field" (condition is repeated for each division).

- To split a protocol into several divisions as a so-called divider. Dividers facilitate the side by side comparison of results of different treatment groups within one protocol/experiment. They are also used to prevent the aggregation of data that should not be averaged.

The way conditions are used and which divisions are created is not fixed for a protocol. Each project team can choose how a protocol should be pivoted and/or divided for their particular purpose. This can be changed over time and it is even possible to define the same protocol multiple times within the same IPV. Projects with different use of conditions lead to alternative views on the same data. This allows scientists to focus on different aspects of the data in a flexible way.

Per protocol zero or more dividers are chosen, e.g., route (of administration) or species. Per divider one or more divider values are chosen (e.g., "P.O." or "I.V." for route). The divisions are derived from this information; the combination of dividers and values is "cross-multiplied". The result is a separate division for every divider combination (see also Figure 6).

Calculations. *Unit Conversion.* At levels 2 and 3, a standard unit of measure is defined per column. Data copied from the source systems are converted to that unit, which facilitates comparing, sorting, aggregating, and querying data (see Figure 7).

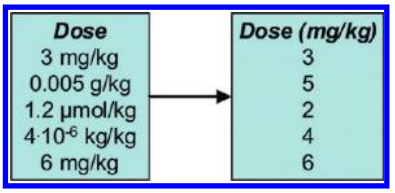


Figure 7. Example of unit conversion for a dose column; from miscellaneous units of measure to milligrams per kilogram.

Derived (Calculated) Columns. Single column calculations may be used at levels 2 and 3 to show extra information. This information is usually inherent to the data, but not stored explicitly in the source system. These calculations are applied to every row in the table and the resulting values are stored as a separate column. The use of this functionality is quite diverse, but some commonly used calculations are the p-value calculation ($Y = -^{10}\log X$) and its inverse ($Y = 10^{-X}$) to convert an EC_{50} to a pEC_{50} (e.g., 1×10^{-5} into 5) or vice versa.

For compound profiling, another calculation type is available at level 1 establishing an arithmetical operation on two columns of values (e.g., the ratio between two EC_{50}s from different protocols, which gives information on the selectivity of a compound; see Figure 8). The operation is performed for each row in the table; i.e. per compound. The results are stored in so-called "Math Columns".

Compound	Target X EC50	Target Y EC50	Y/X EC50 ratio
Compound A	1.00E-08	2.50E-06	2.50E+02
Compound B	1.12E-09	1.00E-05	8.93E+03
Compound C	3.50E-07	3.30E-07	9.43E-01
Compound D	2.78E-10	2.56E-09	9.21E+00

Figure 8. Example of a math column giving the ratio between the EC_{50} values for protocols target *Y* and target *X*. A higher number means a compound is more selective for target *X*.

In general math columns are defined in the project definition and stored in the level 1 table of the project. Alternatively, scientists can create math columns in IPV Explorer, which are not stored physically but are rendered for ad hoc use, without the need to rebuild the project.

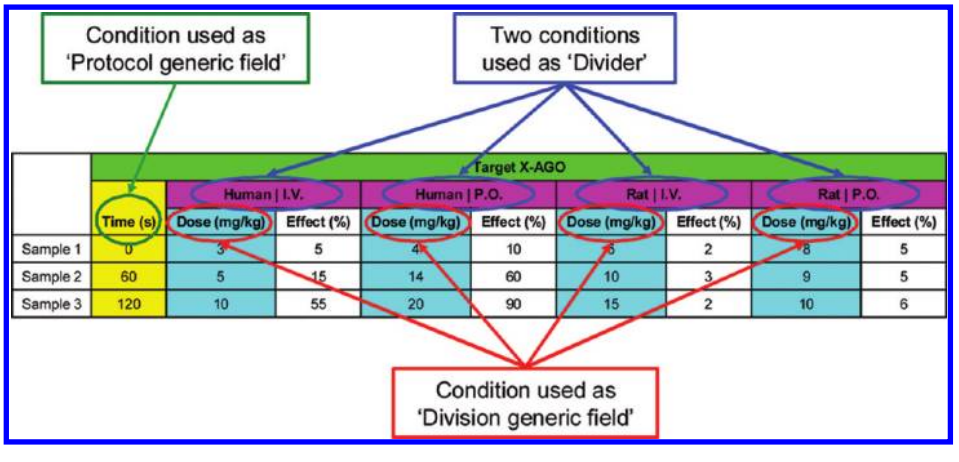


Figure 6. Example of the various ways conditions are used in an in vivo experiment. The conditions used are time (protocol generic field), species and route (dividers), and dose (division generic field).

dx.doi.org/10.1021/ci200253g | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Level 1 Aggregation. At level 1 all underlying level 2 data is aggregated into a single row per compound (see Figure 9). For

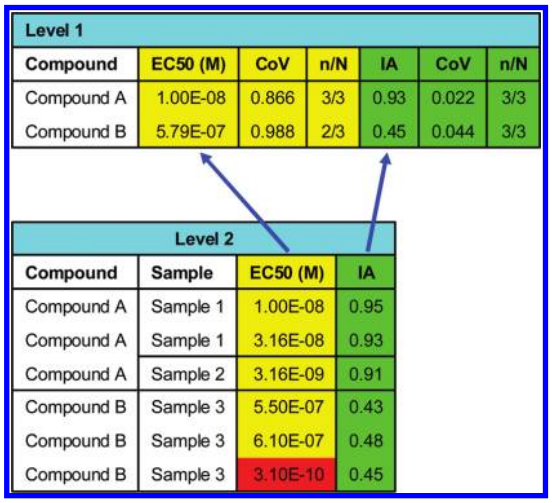


Figure 9. Example of level 1 aggregation. EC_{50} is aggregated with a logarithmic mean, and intrinsic activity is aggregated with a linear mean. Note that one of the EC_{50} values at level 2 is excluded (in red), because it is considered an outlier, giving rise to $n/N = 2/3$ at level 1.

aggregation, empty and excluded values are skipped. There are several ways a value on level 2 can be excluded from aggregation:

- Manual—a value is marked for exclusion by a team member in IPV Explorer, for instance in case of a clear outlier.
- Automatic by setting—determined by project-specific settings, values can be excluded based on their properties (e.g., to exclude unauthorized values).
- Automatic by operator handling—values may be excluded automatically in case of a conflict based on their operators (see "Operator Handling").

Each aggregated value is accompanied by two statistical values that will help the scientists to appraise it:

- Value count in the format "n/N", where "N" is the total number of nonempty values available and "n" is the number of values actually used in the aggregation.
- Coefficient of variation (= standard deviation divided by the mean).

Different aggregation methods can be set per column. Most notable are the linear mean (e.g., for pEC_{50} or IA) and the logarithmic mean (e.g., for EC_{50}).

Operator Handling. One of the complicating factors when handling biological data is the use of operators. Operators are used to indicate incomplete data; e.g. $EC_{50} > 1 \times 10^{-5}$ M is used when the highest applied concentration in a dose–response experiment is 1×10^{-5} M, but the curve is incomplete (i.e., a curve could be plotted through the data, but did not reach 50% of the curve height at the highest concentration).

Operators influence the way a value is handled in calculations, sorting, and querying. Depending on the type of data handling, operators are handled in different ways. As an illustration the operator handling in level 1 aggregation is elaborated in the next section.

Operator Handling in L1-Aggregation. The data aggregation from level 2 to level 1 involves a complex way to handle operators. Several operator handling rules can be chosen, depending on the type of data and the (perceived) quality of the data (see Table 2). For each individual column to be aggregated the operator handling rule can be defined.

Table 2. Operator Handling Rules Available for Level 1 Aggregation

rule	description
strict operators	Strict operator handling—values may not be in conflict with each other; e.g. the largest value with (>) may not exceed the smallest value with (<). In the case of conflict, all values are excluded from aggregation and no mean is calculated.
exclude operators	Only values with (=, ~)[a] are used in the aggregation. All values with (>, ≥, <, ≤) are excluded.
flexible operators	Values with (=, ~)[a] take precedence over values with (>, ≥, <, ≤).
	When there are values with (=, ~),[a] all values with (>, ≥, <, ≤) are excluded from aggregation.
	In case there are only values with operators, values may not be in conflict with each other. In the case of conflict, all values are excluded from aggregation and no mean is calculated.

[a]Note that an equals sign (=) is used to indicate a value without operator.

The operator handling rule chosen determines which values are used in the aggregation and determines the resulting operator for the aggregated value (see Table 3).

Table 3. Examples of Operator Handling Results for Level 2 to Level 1 Aggregation with Linear Mean Calculation and Strict Operator Handling

level 2 data set	agg. value	remark
>3, 4, 5, <10	4.5	mean of 4 and 5
>3, >4	>4	
>5, 6, ~8	~7	mean of 6 and ~8
>8, <3	none	conflict in values

APPLICATION INTEGRATION WITH IPV

Once the IPV project tables are populated or updated with all the key project data, an excellent starting point for data analysis, querying, and decision making has been reached. This is made even more effective by integration with other applications and systems.

Protocol Manager. Protocol Manager, another in-house application, can be launched from within IPV to provide context to the assay data shown (description of experiment, analysis, and meta data). The meta data stored in Protocol Manager are used for IPV project definition.

Spotfire DecisionSite. When Spotfire became available within Organon, it was integrated with IPV for advanced visualization capabilities. This integration was implemented via a direct (OLE) link from IPV Explorer to the Spotfire DecisionSite client,[23] enabling data transfer from an open IPV view or query result into Spotfire, and providing a bidirectional integration: highlighting of data points within one application will be mirrored in the other (see Figure 10). In addition Spotfire can be opened independently, where Spotfire sources current project data directly from the IPV tables. The IPV build and update procedures automatically update the Spotfire meta data repository, using the available webservice from the Spotfire Server API. As a result, all the required Spotfire domains, elements, and joins for each project are created and maintained without the need for manual intervention.

dx.doi.org/10.1021/ci200253g | J. Chem. Inf. Model. XXXX, XXX, XXX–XXX

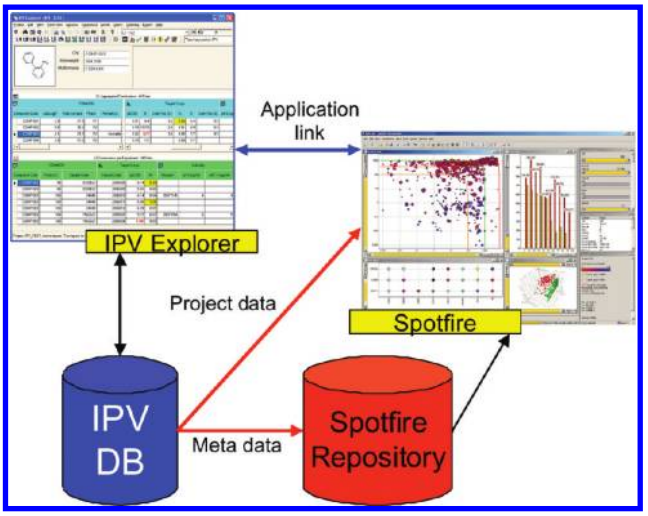


Figure 10. The IPV system links to Spotfire DecisionSite. There are two options: either a direct bidirectional link between the IPV Explorer and the Spotfire client (blue arrow) or access from Spotfire to the project data in the IPV database (red arrows).

The end users gain access to their project data by opening an automatically created information link or through a so-called Spotfire Guide. These Guides, set up by a Spotfire specialist, allow the project teams to view live IPV data directly in predefined visualizations tuned to the specific challenges faced by the drug discovery team; in addition to IPV Explorer these Spotfire views are now used on a routine basis in project meetings and reviews.

Other Integrations. IPV Explorer also allows exports of data to different file formats (e.g., text, SD-file) and to MS Office applications (Word, Excel). Moreover, direct links to more generic in-house chemical and biological data browsers are available, as well as to the compound ordering system and several cheminformatics web tools.

PERFORMANCE

Initial Implementation. While performance of IPV was one of the main design considerations from the beginning, we still had our share of issues in this domain over the years. We therefore feel it could be instructive to provide a semihistoric overview outlining why performance issues emerged, despite all the precautions at design time, and to describe the approaches followed to tackle these problems. The original performance specifications for IPV were rather modest: it needed to be able to handle approximately six concurrent projects, each consisting of 200−300 compounds and a few tens of protocols. At that time this was the typical data volume for a Lead Optimization project at Organon. Early testing showed that this order of magnitude could be handled easily and even tests up to a thousand compounds per project showed no performance bottlenecks at all.

Reactive Tuning—Better Coding. During the first few years after its introduction, IPV encountered a gradual performance degradation of the nightly build and update process. This was not considered to be alarming, however, as, due to IPV's success, both the number and size of the projects had outgrown the original expectations by far. Moreover, after its initial release, IPV had been upgraded to support more data sources and to provide a richer functionality.

At some point though, the time needed for the nightly jobs rapidly increased from hours to days, requiring system modifications to meet the key performance requirements as before, i.e. to be finished before working hours. At that time, the system consisted of approximately fifty active projects containing up to 20 000 compounds each. Using Oracle Statspack[24] as well as the standard Oracle tracing option, we identified bottlenecks (mainly the most time-consuming queries), discovered the proper indexes were not always used, and found some unexpected behavior in the Oracle 9i optimizer. By adapting queries and program code accordingly, we were able to restore acceptable performance in the short run. The main lesson learned from this experience was that, although performance degradation may start as a linear process, it can increase exponentially when the amount of data exceeds certain limits. For IPV specifically, we found that the procedural way of the initial PL/SQL programming was contributing significantly to this problem. A rewrite was required (and performed) to replace the most time-consuming PL/SQL loops by bulk SQL statements.

Proactive Tuning—Smarter Coding. As a planned change in the company research workflow would significantly increase the amount of source data and compounds to be handled by IPV, we felt the need to proactively adapt IPV to avoid previously experienced performance problems. This time we focused on making the system "smarter" by reducing the amount of data IPV needed to handle as well as reducing the amount of processing required.

This was accomplished by implementation of the following:

- *Staging*. Instead of running its queries against the full source systems, the IPV server process would first make a local copy of the subset of the source data needed for building or updating a certain project. As a result, the often complex queries for the builds and updates run against a significantly smaller data set, reducing query time and memory usage. This concept was adopted from data warehouse building techniques.
- *Smart building*. Originally, every time a project definition was changed, the new project tables were built from scratch. However, often these changes only involved the addition, removal or modification of a few columns out of the tens or hundreds of columns that existed in the project. Here, the amount of work for the building algorithm was reduced by identifying which columns were left unchanged with respect to the existing project definition. These were left untouched and copied over from the existing to the new project tables. Now only data for the new or modified columns had to be handled.
- *Smart updating*. Originally, when the update process detected a change in the source data for a certain sample it would remove each row entirely (i.e., the data for all protocols) for that sample and rebuild them all. This mechanism was upgraded to drop and rebuild only the data for the protocol(s) for which the change in data for that sample had occurred.

The effect of these enhancements was spectacular: in some areas the data volume had increased almost thousand fold while the system was in many aspects faster than it was originally.

- **Other Enhancements Added Later.** Partitioning of the tables that contained data for all projects using the standard Oracle partitioning mechanism,[25] i.e. the data in these tables, including indexes, are automatically split-up

dx.doi.org/10.1021/ci200253g | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

and stored per project. This allows for a faster response, as data in IPV is usually retrieved per project.

- Partitioning of the structure tables by creating a separate structure table for each project. This had to be custom programmed, as the domain indexes for the MDL Structure Cartridge cannot be partitioned via standard Oracle partitioning. The structure tables are built and maintained automatically in the overall build and update process. Structure query performance in the IPV Explorer client application benefited significantly: depending on the size of the project and the type of structure query, the reduction in query time measured was up to 95%.
- Making the build and update procedures re-entrant, which allows for multiple simultaneous jobs to handle builds and updates in parallel. Jobs can be assigned specific tasks, e.g. a job that only handles updates, making sure that the updating of the data in the existing projects continues even when large projects are being built.

FINDINGS: BENEFITS, RESULTS, AND OBSERVATIONS

In the early days—especially within parts of discovery research where central storage of data was yet to become general practice—IPV has truly stimulated the increase in the number of experimental data sets that were loaded into the corporate databases as illustrated in Figure 11.

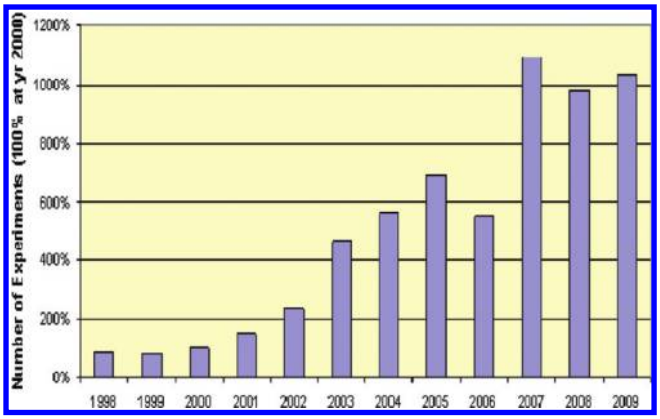


Figure 11. Increased central storage of data sets in the Newhouse (Scotland) Research site after introduction of IPV in 2002. The dip in 2006 and subsequent spike in 2007 are caused by the introduction of new assay automation systems. The subsequent plateauing is due to saturation (i.e., almost all users were uploading all their data sets).

After the launch of IPV, results generated by different research groups at different locations, using similar methods, could all of a sudden be shown side by side in IPV project overviews. As a spin-off from this functionality—to avoid comparing apples and oranges—much effort has been put since into harmonization of test methods across laboratories and sites in all areas of discovery research, e.g., chemistry, pharmacology, and bio/cheminformatics. As a result, data quality and consistency went up significantly over time.

We have been asked repeatedly to provide a convincing return-on-investment (ROI) analysis justifying retrospectively the in-house creation of IPV. Despite all the qualitative and anecdotal evidence available (as outlined above), we have never been able to produce an entirely clear-cut quantitative analysis which is independent of assumptions. The real value of a platform like IPV should lie in the improved design of the right drug compounds, and the early rejection of compounds likely to fail. Assuming these benefits materialize sufficiently in practice (as we believe), it is easy to translate this to an overwhelming ROI, since IPV was initially created by a small team totaling an effort of approximately 5 man-years. Since then, the effort for application management and maintenance has been 2 FTE per year.

Because of the automatic data processing at the server-side (i.e., building and updating of the IPV projects) and the ease of use of IPV Explorer for data sharing, the use of Excel as the main tool for data integration was strongly reduced in project team meetings. In fact IPV is now often used live during team meetings in our multimedia "War Room" (see also Figure 12)

Figure 12. "War Room" in use as a collaboration environment for data-driven drug discovery meetings.

and often shared with off-site team members using web conferencing tools. This way of working not only ensures that all meeting participants use the same, up-to-date data set for analysis and decision making, but also allows drilling down into specific areas of the data set for live review and scientific discussion. This powerful symbiosis of IT tools (i.e., IPV and Spotfire) and a multimedia collaboration environment tends to make team meetings more effective and the resulting conclusions more data-driven.

Following the introduction in 2002, various IPV upgrades were released to satisfy newly emerging requirements. One of the most challenging aspects over the system's lifetime was to preserve performance in order to cope with the ever growing data volumes to be processed. By all means growth expectations have been superseded both from a data volume and scientists' buy-in perspective. Another interesting observation is the distribution of the IPV users over the various scientific disciplines. IPV turns out to be primarily a tool for "data consumers", i.e. medicinal chemists and in silico design chemists, while "data producers" (in particular pharmacologists) tend to use IPV less intensively. Chemists have become the primary user community accounting for over 60% of the IPV sessions (as illustrated by Figure 13), where the annual total is running well into the 10 000s of sessions.

Initially, IPV was designed for and used by lead optimization teams only, but the potential benefits were quickly recognized by other groups as well:

dx.doi.org/10.1021/ci200253g | J. Chem. Inf. Model. XXXX, XXX, XXX–XXX

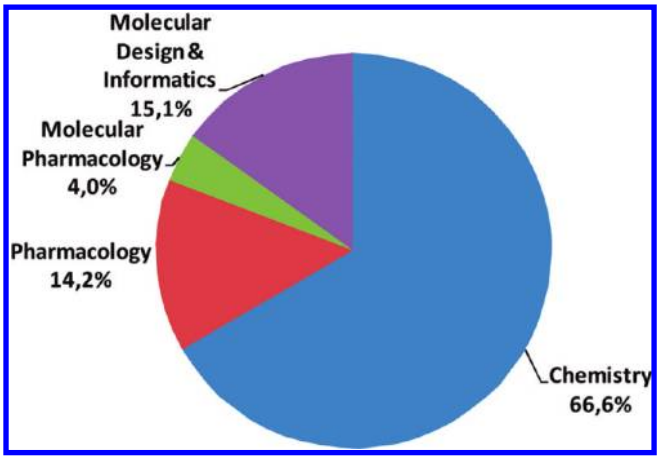


Figure 13. IPV usage over various disciplines within discovery research. If corrected for relative size of the scientific communities, the percentage usage would tilt even more toward chemistry and molecular design and informatics, since these are the smaller departments.

- Over the past decade IPV has also been introduced within teams operating in the earlier phases of the drug discovery process, in particular during lead finding.
- So called "functional groups", such as DMPK and kinase cross screening teams, also embraced the use of IPV for comparing and sharing test results.

The latter example demonstrates that the IPV project concept is applicable to many logical subsets of the corporate data repositories that can be gathered into a data mart. Basically, IPV can be adopted in any research area where decisions about compounds need to be made on the basis of a multiparameter data set.

Despite all successes mentioned, IPV usage has also reached some of its limits. Data volumes have grown well beyond the volumes anticipated at the time of system design. Interestingly, one of the contributing factors is the strong sense of ownership by the project teams, i.e. the tendency to collect not only the key project data in IPV, but also data only remotely connected to the project. In the perception of the project teams all these data need to be collected in "their IPV" overlooking the fact they could leave less interesting data in the underlying data sources for the time being, until they actually become relevant for their project. As a result, these projects may experience performance hits during fetching/retrieving the data from the database on the server into the IPV Explorer on the client PC. Occasionally, also the number of result columns requested by the project teams would exceed the fixed Oracle limit of 1000 columns.

A similar effect is seen when closing projects. The project IPV remains to be perceived by the scientists as a valuable asset even after project closure, whereas the discovery informatics team tends to position IPV merely as a temporary view on underlying primary data sources. In the latter perspective, the IPV project could be disposed and recreated whenever needed.

DISCUSSION/OUTLOOK

Although the IPV ideas were highly supported by the scientific community, one of the key hurdles to be taken during the introduction was the acceptance of the data sharing concept. IPV is founded on corporate databases that are being shared openly within the entire discovery research organization. The debate centered around two specific concerns with this high level of data sharing:

1. *Data sharing could lead to leaking of confidential results potentially damaging the company's interests.*

   This debate has been fierce over the years cumulating at the conclusion that the value created through open sharing of data within the company's scientific community outweighed the potential risks. Senior R&D management support was crucial for overcoming the resistance in this area. Of course, we also implemented the company-standard technical and procedural system security measures in order to mitigate the potential risks as far as possible.
2. *Sharing complex scientific results could lead to misinterpretation of results by scientists not being experts in a particular field.*

   This concern in particular surfaces when dealing with late-stage pharmacological experiments where it may be hard to grasp the full scientific complexity as a nonexpert. The company took a clear position in this debate: on the one hand, it was felt that instead of jumping to conclusions it is every scientist's obligation to consult experts in the field to arrive at scientifically sound conclusions about data outside their own area of expertise. On the other hand, data in corporate databases need to be registered in such a manner that other experts in the same discipline do have sufficient level of detail and the right context to interpret the results correctly. The Protocol Manager application, describing assays in great detail with all corresponding meta data, is one of the important measures taken to enforce scientists to provide this context consistently, prior to data entry.

It would be interesting to see how IPV compares to similar platforms in other companies.[7−10] Unfortunately, such a comparison is hardly possible in practice due to lack of access to the real-life versions of each of the systems. It seems that some of the other systems, most notably ABCD, are more feature-rich and broader in scope. However, we consider it IPV's strength to be very project-centric focusing entirely on the needs of a drug discovery team and dealing effectively with the peculiarities of the data typically produced in this process.

Another question we can ask ourselves is how we would build a system like IPV today. Such a modern solution is likely to consist of a hybrid of generic IT development platforms such as supplied by Microsoft.NET[26] and Oracle combined with specific life science oriented solutions such as visualization tools,[23,27] data integration environments,[11,28,29] and pipelining tools.[30,31] Also modern data warehousing techniques (e.g., data vault[32]) and business intelligence concepts would have to be considered.

More advanced data integration techniques may become relevant when the data integration ambition extends beyond the discovery research phase of Pharma R&D. Effectively translating (late stage) development results back into the early discovery research phases ("Bridging R&D") is one of the holy grails for drug discovery in an attempt to reduce late stage attrition rates in development.[33] Theoretically, generic data integration concepts supplied by companies like IBM[34] could facilitate this cross-R&D data integration. However, actually making a decision support impact with such an implementation may very well turn out to be too challenging at this time: the question still remains how to integrate "tall and skinny"

dx.doi.org/10.1021/ci200253g | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

research results (many compounds with a limited number of data points per compound) with "short and fat" development outcomes (many results for a few compounds) in a meaningful manner.

Another area to be considered when returning to the drawing board today would be the handling of in silico data sources. Ideally, SAR models validated by the cheminformatics specialist in the project team would be built into the team's IPV and automatically recalculated when new data arrive, an implementation certainly tangible with the current generation of pipelining tools. By the same token several other informative data presentations could be automated, such as R-Group analyses and mapping of assay results onto protein family trees (e.g., as found in the screening of a kinases panel). Some interesting initiatives in this domain have recently been reported.[35,36] Another idea would be the inclusion of virtual compounds (and their calculated properties) in IPV, i.e., compounds under consideration for future synthesis but still to be prioritized.

Another challenging area to be considered for future decision support systems is the introduction of "unstructured" data types, such as images, graphs (e.g., dose−response curves), metabolic pathways, and documents. Typically, project teams tend to collect these data into dedicated Microsoft Sharepoint sites[37] used next to IPV. Technical developments around Sharepoint by Microsoft itself, as well as by informatics vendors dedicated to the Life Sciences domain (such as Accelrys) may very well enable a solution where IPV-like data integration for structured data is combined with more traditional unstructured data storage in Sharepoint. An interesting first step in this domain is the utilization of OneNote[38] as the user interface on top of Sharepoint as recently published.[39]

CONCLUSIONS

In 10 years of use, IPV has become such a strong brand that users have gradually perceived IPV as their invaluable database and started thinking they load their data directly into IPV rather than in the corporate databases. The greatest success of IPV may have been the virtual elimination of Excel spreadsheets from project team meetings. This strong uptake of IPV and its underlying concepts reinforced our belief in the data-driven approach.[2] Nevertheless, many enhancements are currently under consideration for the next generation of IPV with flexibility, maintainability, and the introduction of new scientific concepts as the focal points. Finally, it would be interesting to speculate about applying the fundamental IPV concept to entirely new domains, e.g. gene sets to be prioritized as potential biomarkers.

AUTHOR INFORMATION

Corresponding Author

*E-mail: jw.boiten@gmail.com.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We thank Dike van Beijnen, Rich Bell, Ann Brown, Maurizio Camporelli, Joop Cramer, Philip van Galen, Niels de Greef, Daniel Janse, Joost van Kempen, Tiny van Lanen, Chris Melgert, Sylvia Odusanya, Matthijs Rademakers, Brian Reilly, Wendy Saywood, Marten Schoonman, Renate Tjee, Gerrit Tol, Harjeet Virdi, Erik de Vocht, Jan Wertenbroek, and last but not least the IPV key-users for their contributions to the development and growth of IPV.

REFERENCES

(1) Munos, B. Lessons from 60 years of pharmaceutical innovation. *Nat. Rev. Drug Discovery* **2009**, *8*, 959−968.
(2) Lusher, S. J.; McGuire, R.; Azevedo, R.; Boiten, J. W.; van Schaik, R. C.; de Vlieg, J. A Molecular Informatics View on Best Practices in Multiple-Parameter Compound Optimization. *Drug Discovery Today* **2011**, *16*, 555−568.
(3) Searls, D. B. Data integration: Challenges for Drug Discovery. *Nat. Rev. Drug Discovery* **2005**, *4*, 45−58.
(4) Waller, C. L.; Shah, A.; Nolte, M. Strategies to support drug discovery through integration of systems and data. *Drug Discov. Today* **2007**, *12*, 634−639.
(5) *Excel*, version 2002; Microsoft: Redmond, WA, 2001.
(6) ISIS Overview. http://accelrys.com/products/informatics/decision-support/isis.html (accessed December 11, 2011)
(7) Agrafiotis, D. K.; et al. Advanced Biological and Chemical Discovery (ABCD): Centralizing Discovery Knowledge in an Inherently Decentralized World. *J. Chem. Inf. Model.* **2007**, *47*, 1999−2014.
(8) Cho, S. J.; Sun, Y.; Harte, W. ADAAPT: Amgen's data access, analysis, and prediction tools. *J. Comput.-Aided Mol. Des.* **2006**, *20*, 249−261.
(9) Rojnuckarin, A.; Gschwend, D. A.; Rotstein, S. H.; Hartsough, D. S. ArQiologist: An Integrated Decision Support Tool for Lead Optimization. *J. Chem. Inf. Model.* **2005**, *45*, 2−9.
(10) Sander, T.; Freyss, J.; Korff, M.; von; Reich, J. R.; Rufener, C. OSIRIS, an Entirely in-House Developed Drug Discovery Informatics System. *J. Chem. Inf. Model.* **2009**, *49*, 232−246.
(11) Isentris Overview. http://accelrys.com/products/informatics/decision-support/isentris.html (accessed December 11, 2011)
(12) ChemCart a web-based search and update tool for scientific information. http://www.deltasoftinc.com/docs/chemcart.pdf html (accessed December 11, 2011)
(13) *Visual Basic*, version 6; Microsoft: Redmond, WA, 1998.
(14) *MDL ISIS Desktop*, version 2.5; MDL Information Systems Inc.: San Leandro, CA, 2003.
(15) *Symyx Direct*, version 6.3; Symyx Technologies Inc.: Sunnyvale, CA, 2009.
(16) *ComponentOne Chart*, version 7.0; ComponentOne: Pittsburgh, PA, 2001.
(17) *Apex*, version 3.0; Oracle: Redwood Shores, CA, 2007.
(18) *Ultragrid*, version 2.0; Infragistics: Cranbury, NJ, 2003.
(19) *VSFlexGrid Pro*, version 7; ComponentOne: Pittsburgh, PA, 1999.
(20) *TrueDBGrid*, version 7; ComponentOne: Pittsburgh, PA, 2000.
(21) *Spread*, version 3.0; Farpoint: Kirkland, WA, 2000.
(22) *Oracle 10g*, version 10.2; Oracle: Redwood Shores, CA, 2005.
(23) *Spotfire Decision Site*, version 9.1; TIBCO Software Inc.: Palo Alto, CA, 2007.
(24) Using Statspack. http://docs.oracle.com/cd/B10501_01/server.920/a96533/statspac.htm (accessed April 15, 2011).
(25) Partitioned Tables and Indexes. http://download.oracle.com/docs/cd/B19306_01/server.102/b14220/partconc.htm (accessed April 15, 2011).
(26) NET Downloads, Developer Resources & Case Studies - Microsoft .NET Framework. http://www.microsoft.com/net (accessed December 11, 2011).
(27) *Vortex*, version v2012.04.1.4098; Dotmatics: Bishops Stortford, U.K., April 2012.
(28) *D360*, version 5.5; Tripos: St. Louis, MO, April 13, 2011.
(29) *ChemBioViz*, version 12.0; PerkinElmer: Morrisville, NC, July 2009.
(30) Pipeline Pilot Overview. http://accelrys.com/products/pipeline-pilot/ (accessed December 11, 2011).
(31) KNIME User Documentation. http://tech.knime.org/knime (accessed December 11, 2011).

(32) Linstedt, D. E. Data Vault Overview. http://www.tdan.com/view-articles/5054/ (accessed April 15, 2011).
(33) Wehling, M. Assessing the translatability of drug projects: what needs to be scored to predict success? *Nat. Rev. Drug Discovery* **2009**, *8*, 541−546.
(34) IBM Life Sciences Solutions: Turning Data into Discovery with DiscoveryLink. http://www.redbooks.ibm.com/abstracts/sg246290.html (accessed April 15, 2011).
(35) Brodney, M. D.; Brosius, A. D.; Gregory, T.; Heck, S. D.; Klug-McLeod, J. L.; Poss, C. S. Project-Focused Activity and Knowledge Tracker: A Unified Data Analysis, Collaboration, and Workflow Tool for Medicinal Chemistry Project Teams. *J. Chem. Inf. Model.* **2009**, *49*, 2639−2649.
(36) Agrafiotis, D. K.; Wiener, J. J. M. Scaffold Explorer: An Interactive Tool for Organizing and Mining Structure − Activity Data Spanning Multiple Chemotypes. *J. Med. Chem.* **2010**, *53*, 5002−5011.
(37) *SharePoint*, version 2010; Collaboration Software for the Enterprise; Microsoft: Redmond, WA, June 15, 2010.
(38) *OneNote*, version 2010; Planner and note taking software; Microsoft: Redmond, WA, May 12, 2010.
(39) Barber, C. G.; Haque, N.; Gardner, B. "One point" − combining OneNote and SharePoint to facilitate knowledge transfer. *Drug Discovery Today* **2009**, *14*, 845−850.