IV

Informatics and Bioinformatics

20

Overview of Informatics for High Content Screening

R. Terry Dunlay, Wallace J. Czekalski, and Mark A. Collins

Summary

With the growing use of high content screening (HCS) and analysis in drug discovery and systems biology, informatics has come to the forefront as a critical technology to effectively utilize the massive volumes of high content data and images being generated. Informatics technologies are required to transform HCS data and images into useful information and then into knowledge to drive decision making in an efficient and cost effective manner. In this chapter, we provide an overview of informatics tools and technologies for HCS, discuss some of the challenges of harnessing the huge and growing volumes of HCS data, and provide insight to help toward implementing or selecting, and utilizing a high content informatics solution to meet your organization's needs.

Key Words: Data integration; data management; data mining; databases; high content screening; image management; informatics; N-tier architecture; visualization.

1. Introduction

High content screening (HCS) systems generate enormous amounts of data and images that are pushing the limits of conventional information technologies. The massive volumes of feature-rich data and images being generated by these systems and the effective management and use of information from the data have created a number of challenges. These challenges lie not only in the capabilities of the software and hardware technologies, but also in educating users in the optimal use of informatics tools. In addition, partnerships between researchers and their counterparts in information technology (IT) are critical to effectively manage HCS data, share it, and integrate it, so that it can be used in meaningful ways. To fully exploit the potential of data and images from modern high content systems, it is therefore crucial to understand the key factors in determining a suitable high content informatics solution to fit your organization's needs.

HCS systems typically scan a multiwell plate with cells or cellular components in each well, acquire multiple images of cells, and extract multiple features (or measurements) relevant to the biological application, resulting in a large quantity of data and images. The amount of data and images generated from a single microtiter plate can range from hundreds of megabytes (MB) to multiple gigabytes (GB). Large numbers of plates are typically analyzed in screening operations and large-scale system biology experiments, often resulting in billions of features and millions of images with a need for multiple terabytes (TB) of storage in a short period of time.

High content informatics tools are needed to manage the large volume of HCS data and images generated for collection, storage, retrieval, analysis, and display to enable understanding of the samples under investigation. The importance of informatics for HCS is briefly discussed in **refs.** *1* and *2*.

From: *Methods in Molecular Biology, vol. 356:*
High Content Screening: A Powerful Approach to Systems Cell Biology and Drug Discovery
Edited by: D. L. Taylor, J. R. Haskins, and K. Giuliano © Humana Press, Inc., Totowa, NJ

Our goal in this chapter is to provide an overview of the key aspects of informatics tools and technologies needed for HCS, including characteristics of HCS data; data models/structures for storing HCS data; HCS informatics system architectures, data management approaches, hardware and network considerations, visualization, data mining technologies, and integrating HCS data with other data and systems.

2. Characteristics of HCS Data

HCS data is characterized as having large numbers of parameters, massive data sets, and large numbers of high resolution images that require significant amounts of storage, especially in drug discovery, and systems biology applications. In order to better understand these characteristics, we will provide some background. We should note that when we refer to typical values here and throughout the rest of the chapter, we are basing these on our experience and they by no means cover the full range of possibilities.

HCS systems typically scan and analyze multiwell microtiter plates. These "plates" typically have 96, 384, or 1536 wells. Each "well" is a container in the plate that typically contains an individual sample of cells. Each well is divided into multiple fields. Each "field" is a region of a well that represents an area to image (this is also sometimes referred to as a "field-of-view," "frame," or "scene"). Each field typically consists of multiple images, one for each individual wavelength of light (referred to as a "channel" or "color"), corresponding to the fluorescent markers/probes used for the biology/dye of interest (e.g., Hoechst). There are typically between one and four channels per field (e.g., one channel may show the nuclei, another the cytoplasm, another the cell membrane, and so on). In each field, a certain number of cells are selected to be analyzed by the HCS system. The number of cells per field varies depending on the experiment, but typically ranges between 10 and 500 cells. For each cell, multiple cell features (or measurements) are calculated by the HCS system's image analysis algorithms. The cell features include measurements such as size, shape, intensity, and so on. The number of cell features calculated varies depending on the assay, but typically ranges between 5 and 50. In addition, cell features are often aggregated to the well level to provide well level statistics familiar to discovery scientists. The well features include measurements such as average size, standard deviation of size, average shape, total intensity, and so on. The number of well features varies depending on the assay, but typically ranges between 5 and 50. In kinetics assays, the above measurements are taken at multiple points in time, from a few seconds to minutes or hours, and additional features (e.g., rate changes, min, max, and so on) are also calculated. The number of time-points again varies, but typically ranges between 1 and 10. Thus, a large amount of data is collected for just one well of a single plate. In addition, other associated information about the assay or experiment, such as protocol information, is also typically recorded.

We define three categories of HCS data:

1. *Image data*—these are the images acquired at each channel for each field within a well.
2. *Derived data*—these are the measurements that result from performing an analysis on an image with image analysis algorithms (e.g., well features, cell features, and so on).
3. *Metadata*—these are the associated data that provide context for the other two categories of data (i.e., metadata is data that describes other data). For example, assay type, plate information, protocols, operators, calculated data such as dose–response values, as well as annotations imported from other systems (e.g., sample identifiers and properties).

From a data volume perspective, the data to be saved per plate is primarily based on the image data and the derived data. The size of the Metadata in comparison is negligible. For each well of a plate, the data is estimated by the number of feature records needed to store the derived data and the number of images acquired. The number of images acquired can be estimated by: [number of wells × number of fields × images per field (i.e., the number of channels × number of time-points)]. The typical size of an image ranges between 262 kb (for a 512 × 512 × 1 byte image)

Table 1
Example Data Volumes for Different HCS Application Scenarios.

HCS application scenarios (assuming 100 cells per field and an image size of 0.5 MB ([= 0.05 GB] for all examples)	Image data, number of images (storage GB)	Derived data (GB), number of records (storage GB)
One hundred 96-well plates, one field/well, two channel/field, 20 feature/well, 50 features/cell, one time-point	192 images (9.6 GB)	48.2 million records (1.5 GB)
One hundred 96-well plates, two field/well, three channel/field, 50 feature/well, 25 features/cell, one time-point	576 images (28.8 GB)	48.5 million records (1.6 GB)
One hundred 96-well plates, four field/well, four channel/field, 20 feature/well, 50 features/cell, one time-point	1536 images (76.8 GB)	192.2 million records (6.2 GB)
One hundred 96-well plates, 10 fields/well, two channels/field, 50 feature/well, 50 features/cell, one time-point	1920 images (96 GB)	480.5 million records (15.4 GB)
One hundred 384-well plate, two fields/well, two channels/field, 20 feature/well, 100 features/cell, one time-point	1536 images (76.8 GB)	769.9 million records (24.6 GB)
One hundred 384-well plate, four fields/well, two channels/ field, 50 feature/well, 50 features/cell, one time-point	6144 images (307.2 GB)	769.9 million records (24.6 GB)
One hundred 96-well plates, one field/well, two channel/field, 20 feature/well, 50 features/cell, 10 time-point	1920 images (96 GB)	480.2 million records (15.4 GB)
One hundred 96-well plates, four field/well, three channel/field, 50 feature/well, 50 features/cell, 20 time-point	11,520 images (576 GB)	960.5 million records (30.7 GB)

Shown are data volumes for Image Data and Derived Data together with the "number of images" and associated storage requirements and the "number of records" for cell and well features stored in the database and estimated storage requirements.

and 2 MB (for a $1024 \times 1024 \times 2$ byte image). Images are often compressed using some form of lossless compression, which usually results in a 25–50% storage reduction. For derived data, the number of cell feature records can be estimated by (number of wells × number of fields × number of cells × number of features per cell) and the number of well features can be estimated by (number of wells × number of features per well).

The amount of data generated in a period varies depending on a number of factors including the biological assay, the types of experiments or tests to be run, the throughput of the instrument or analysis application, the number of instruments, and so on. **Table 1** shows ranges of possibilities for different types of example assays (*see* **Note 1** for detailed example calculations). This data could be generated in days or weeks leading to tens of TB of storage requirements in a few months.

Although similar to other informatics modalities in some aspects, high content informatics has some unique characteristics. The requirements for management of high content data and images are different than the requirements for purely managing images with simple annotated data. In high content informatics, the data is supported by the images as opposed to the images supporting annotated data. As we can see from the **Table 1**, high content data is far more complex and voluminous than simple image annotations. Any high content informatics solution therefore needs to be able to efficiently handle the relationships between the various levels of feature data and the associated images.

3. Data Model/Structure for HCS Data

To enable effective decision making in HCS, data and images and associated information must be stored with high integrity in a retrievable form. HCS data should be stored in a manner that takes advantage of the characteristics of this type of data to enable full access and exploitation of the data. The underlying data model (or database structure or database schema) should be flexible to handle the various HCS data types (i.e., image data, derived data, and metadata) and a wide range of changes in the data (e.g., different numbers of wells, cells, features, images, different image sizes and formats, different number of time-points [in kinetic assays], and so on).

The structure of the metadata is also important. The metadata provides a means of describing data and the relationships within the data, enabling data to be better organized, cataloged, and searched effectively. Metadata enables joining of related data to allow meaningful visualization, analysis, and data mining. The metadata is also important for integration with other systems and data sources and defined vocabularies should be used for metadata whenever possible. For example, using defined lists and consistent words for describing assays, samples cell lines, and so on, rather than free comments. This is an area where standards across the HCS field would be helpful, but should at least be consistent within an organization.

4. System Architecture

Managing the collection, storage, retrieval, analysis, and display of huge volumes of HCS data demands a system architecture that utilizes best practices from the world of IT. The system architecture defines the fundamental organization of the system, the underlying structure of the various components and their interrelationships, and the principles governing the overall design.

A key component of any high content informatics solution is the data management component and this is best handled by some form of database technology, because managing HCS data via file based systems does not provide a scalable solution. In contrast, databases (e.g., relational, object oriented, or object-relational) are designed to provide efficient access to large amounts of data. Relational databases are the most commonly used databases for HCS data. Relational databases are available from many vendors (e.g., Oracle, www.oracle.com; Microsoft SQL Server, www.microsoft.com; and so on) and high quality open source databases also exist (e.g., MySQL, www.mysql.com and PostgreSQL, www.postgresql.org). Relational databases address two of the three categories of HCS data, derived data, and metadata. The remaining HCS data category (i.e., image data), are the images usually stored outside the relational database with only pointers to the images being stored in the database, because the database simply grows too large to be efficiently managed with traditional tools if the images are stored directly in the database. In contrast, there are a wide variety of options available when storing the images outside the database. All that is really necessary is a large amount of disk space. However, as the needs of the system grow, more complex technologies may be employed such as Network Attached Storage, Storage Area Networks, Content Addressed Storage, and Hierarchical Storage Management, which are available from various vendors including IBM (www.ibm.com), EMC (www.emc.com), and Network Appliance (www.netapp.com). The key to scaling image storage is that a pointer in the database to the externally stored image must exist in the relational database in order to retrieve the image at a later time.

For the relational database to be useful, the HCS data must be entered into the database via some automated collection, transfer and integration processes. This is an essential and complex task for all but the smallest usage scenarios. This requirement is best handled by a Utility Service *(3)*. Utility Services provide features like processing results in an unattended manner, running even when the computer is not logged on, and sending notifications of important events or status.

Retrieving data from the relational database is equally important as getting it in. Once again, a Utility Service offers the best approach to accomplish this requirement. Such a service will

allow the configuration of permissions to create, view, update, and delete data to be consolidated within the service. This is a key feature that allows the architecture to scale within an enterprise.

Collectively, moving the data in and out of the system comprises the underlying business rules/logic or middleware. Often this logic is exposed to both the given system and external systems by means of a dedicated application server. An application server may expose some or all of the underlying business rules via a Web Service ***(4)***. The Web Service is the key integration point that allows the HCS data to be integrated within a customer's own enterprise data repository.

Client applications are the software applications that are used to visualize, analyze and mine information from HCS data. For users, the client applications are usually the most important component, as these tools are what they interact with on a daily basis. In follow-on sections of this chapter, we review two client applications, visualization tools and data mining tools in more detail.

Combining the relational database, application server, HCS instruments, and client application components together form the basis of a traditional "N-Tier" architecture ***(5,6)***. This can be seen in **Fig. 1** with the associated HCS system components.

An N-Tier architecture refers to a system that has at least three tiers (or "layers") that are separate and each tier interacts only with the tier below (or above) and has a specific function that it is responsible for:

- *Presentation tier*—The presentation tier is for displaying the user interface and driving that interface. Essentially these interfaces are the user facing parts of HCS instrument software and the client applications. This is also sometimes referred to as the "user tier."
- *Middle tier*—The middle tier provides the automated transfer of data from the instruments to the data tier and moves data back from the data tier back to the presentation tier. This tier is also responsible for processing the data retrieved and sent. The middle tier is also sometimes referred to as the "application tier" or the "business tier."
- *Data tier*—The database for the HCS data and the repository for the images reside in the data tier. This is where the three categories of HCS data (image data, derived data, and metadata) are stored.

These tiers can be physically together but conceptually separate. They can also be located on physically different servers even if the servers are in different geographical locations. Separating the logic and processing contributes to the major benefits of N-tier, which are robustness, maintainability, and scalability. The scalability part is especially important, allowing improvements to be applied where needed (e.g., additional or more powerful database servers can be used as data volumes grow).

Because each tier can be located on one computer or physically different computers, each can be scaled to the needs of an organization (i.e., number of users, number of instruments, amount of data, and so on). This approach is provided by Cellomics, Inc. (Pittsburgh, PA). in their HCi™ informatics platform (www.cellomics.com). In limited usage scenarios, all three of these tiers may be physically installed onto a single computer. Usually though, at least the application server and relational database are installed on different computers than the client applications. At the upper end of the scale, multiple computers may be used at each tier. This will generally be the case at sites with multiple HCS instruments and/or client applications.

5. Hardware and Network Considerations

There are a wide variety of ever evolving options for server hardware, storage hardware, and networking capabilities for an organization's informatics solution. The number of HCS instruments, number of users, the number of sites, and the network bandwidth within a site (i.e., Local Area Network) and between sites (i.e., Wide Area Network), are a few of the key factors impacting the hardware requirements for an informatics solution.

Sizing and scoping the optimal hardware for an informatics solution is an area where having professional IT support is critical. Each organization is unique in their HCS usage scenarios, which

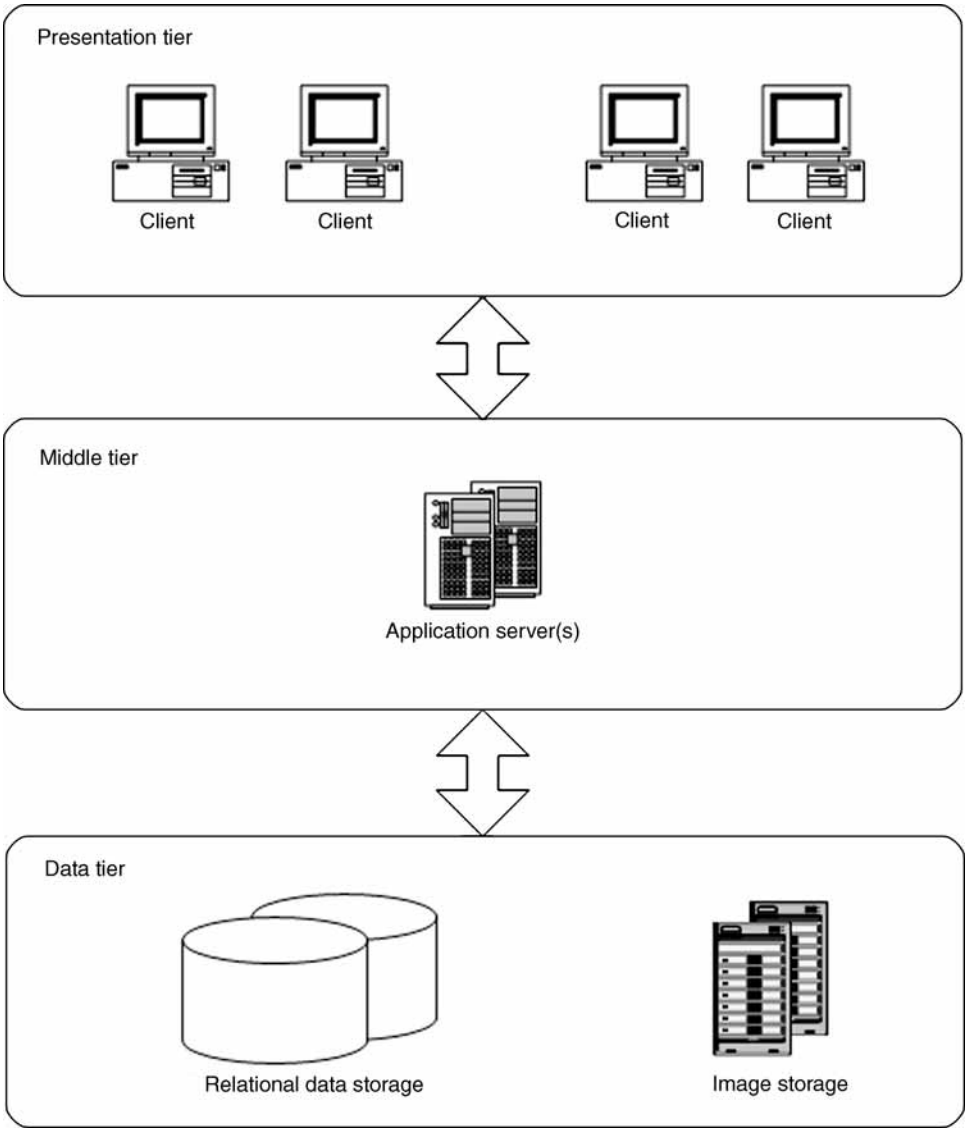


Fig. 1. "N-Tier" Informatics system architecture.

directly impacts the requirements put on an informatics solution. In general, it is best to identify an informatics solution with a system architecture that can scale as the organization's HCS needs evolve over time. For example, a user might start with one instrument, one combined database and image storage server, one application server, and a couple of client applications. Then it may grow to multiple instruments at multiple sites with multiple database servers, multiple image storage systems, multiple applications servers, and multiple client applications. In general, a vendor's informatics solution should be flexible and scalable to fit a variety of hardware configurations and usage scenarios.

A key factor impacting the configuration of the system architecture across multiple sites is the bandwidth of the network that connects the various computers together. HCS instruments typically generate data and images at rate of 1–5 GB (or more) per hour and there are limits to current network and server technology that can support instruments writing this amount of data

across networks at multiple sites. Tradeoffs between network bandwidth, server, and storage system configuration, and each organization's unique use cases for how information will be accessed and shared, all need to be taken into account in order to optimize overall system performance.

6. Data Management

Over a period of time, a tremendous amount of HCS data will be collected. An effective workflow must be developed to manage the data. This workflow will include such things as who is allowed to view and manage the data, how the data will be backed up, and when to archive or delete data.

Policies or procedures for storing HCS data need to be determined by each organization. Specifically what HCS data (Image Data, Derived Data, and Metadata) will be stored for how long (e.g., 5, 10, or 15 yr or more) directly impacts the workflow as well as the overall data volume, scope or data management, and cost. This is currently an area where policies are still being formulated by organizations, and in many cases they just decide to play it safe and store everything, further fueling the need to store, and manage even more data.

Regulatory compliance (e.g., FDA 21 CFR part 11 [www.fda.gov/ora/compliance_ref/part11]) is another issue to consider, in particular, in pharmaceutical, and other highly regulated industries. A high content informatics product that can help an organization establish and maintain regulatory compliance could be critical. Although no product itself can make an organization compliant without the proper policies, a properly designed, developed, and supported product can make complying with regulations significantly easier.

Regarding who can view or manage the data, some forethought must occur just to get the system up and running. Simply assigning everyone full control of the data may be problematic, therefore having access to professional IT personnel who have the experience to assign and manage permissions effectively is extremely important.

Managing permissions is also a key point that reveals why having an application server in an "N-Tier" architecture is so important. Without this type of architecture, all users must be assigned permissions to the file storage, and relational databases. With an "N-Tier" architecture, access to these resources may use a proxy account from the application server. This greatly simplifies deploying and managing the system, especially when trying to share data across multiple sites or different domains.

Backing up the data is another area where having professional IT support is very valuable. As the data volume grows, creating, and maintaining complete backups is a difficult task. The key feature that a successful HCS backup strategy has is preventing the volume of data that needs to be backed up from growing beyond the manageable range of the backup solution.

One of the best approaches to achieve this is to store the HCS data in different locations based on time. A location's time may then be used to determine whether the data has already been backed up. Once a particular location is no longer having data added, a final backup of this location may be completed. This location may then be removed from the periodic backup regimen.

IT professionals can also help with the archiving of data. This is especially true if the data may be archived based on metadata criteria such as creation date, storage location, or creating user. However, if biological metadata like projects, compounds, or hits drive the archive process, then scientists will need the ability to archive data. Regardless of who actually performs the archiving, coordination among users, and IT staff is vitally important to effectively manage HCS data (*see also* Chapter 21).

7. Visualization

Visualization tools are one type of client application mentioned earlier that provide a quick and effective means to interrogate HCS data and images stored in a secure repository. Users want to view the data, share it with colleagues, and compare results. Visualization software

should provide powerful search and navigation tools to rapidly locate plate, well, cell, and image data. Rich search functions should be available to find data based on various metadata and derived data parameters (e.g., user name, dates/times, assay type, features, and so on) (*see also* Chapter 22).

The most basic form of any HCS data visualization tool should provide interactive tools for reviewing data with drill-down capabilities from the plate, well, and cell level together with links to images, and any graphical image overlays. Various forms of viewing the data should be provided including tables/spreadsheets and graphs (bar charts, scatter plots, and so on, *see* **Fig. 2**). Various views should also be provided for different types of users (e.g., managers, scientists, operators, IT personnel, and so on).

Capabilities should be provided for comparing data within a plate, across plates, and so on. Additional capabilities should also be provided for generating statistics on groups of data (e.g., groups of wells, cells, and so on). The data should be displayed in ways that allow the user to explore patterns and recognize patterns and outliers. Users want to be able to save their analyses and visualizations as well as build reports and save these. Making annotations on the data is also very important.

Common uses for visualization in HCS include assessing the quality of the dataset (e.g., identifying outliers and false positives), and identifying hits. There are many possibilities for visualization of HCS data using commercially available tools (e.g., Spotfire (www.spotfire.com), OmniViz (www.omniviz.com), and so on) (*see also* Chapters 13 and 23).

8. Data Mining

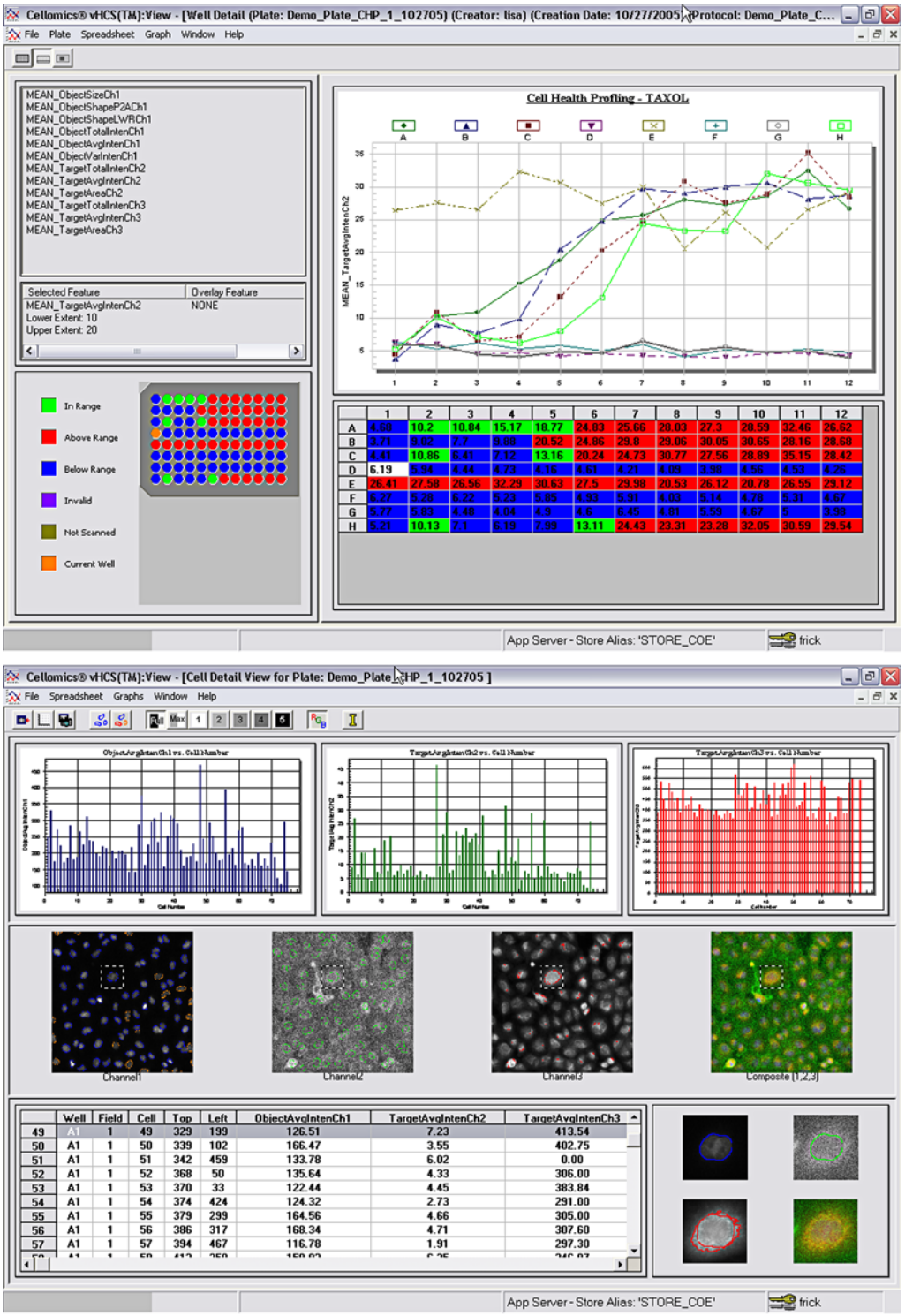
The large amount of multiparameter data inherent in HCS provides opportunities to reveal patterns or trends in the data using data mining tools *(7,8)*. Data mining tools are another type of client application mentioned earlier. These tools can include pattern recognition techniques, self-organizing maps, fuzzy logic, statistical methods, and machine learning methods. In addition to identifying patterns and trends from the data, data mining technologies can be used in making predictions and simulations of future events.

Used together with visualization tools, data mining can be used to discover knowledge in HCS data sets in a form that is more easily understood. The goal is to reduce complexity and extract relevant and useful information from large HCS data sets in an intuitive and efficient manner so that better decisions can be made (*see also* Chapter 23).

Although data mining tools can be a very powerful aid to making important decisions, they are not self-sufficient. To be successful, data mining requires skilled technical and domain specialists who can structure the analysis and interpret the output that is created. For example, data mining can help identify patterns and relationships, but it does not tell the user the value or significance of these patterns. These types of determinations need to be made by the user. Similarly, the validity of the patterns discovered is dependent on how they compare to real world circumstances. Nevertheless, data mining holds great promise as a critical tool for HCS analysis and we expect that data mining will therefore have a significant impact, much as it has had in other industries that have large quantities of data.

9. Integrating HCS Data With Other Data and Systems

With the widespread adoption of HCS throughout the drug discovery and academic research domain, the need to integrate HCS data with other discovery data and external systems has arisen. Indeed, integration has become a key issue as HCS data is used to make decisions that require multiple data sources, from target validation data through to ADME/Tox and preclinical domains. HCS data cannot be a critical part of the drug discovery decision process unless it is effectively integrated. Integration can take many forms, but can be categorized as data-level integration, database integration/federation, and application/software integration.



	1	2	3	4	5	6	7	8	9	10	11	12
A	4.68	10.2	10.84	15.17	18.77	24.83	25.66	28.03	27.3	28.59	32.46	26.62
B	3.71	4.02	7.7	9.88	20.52	24.86	29.8	29.06	30.05	30.65	28.16	28.68
C	4.41	10.86	6.41	7.12	13.16	20.24	24.73	30.77	27.56	28.89	35.15	28.42
D	6.19	2.94	4.44	4.73	4.16	4.61	4.21	4.09	3.98	4.56	4.53	4.26
E	26.41	27.58	26.56	32.29	30.63	27.5	29.98	20.53	26.12	20.78	26.55	29.12
F	6.27	5.28	6.22	5.23	5.85	4.93	5.91	4.03	5.14	4.78	5.31	4.67
G	5.77	5.83	4.48	4.04	4.9	4.6	6.45	4.81	5.59	4.67	5	3.98
H	5.21	10.13	7.1	6.19	7.99	13.11	24.43	23.31	23.28	32.05	30.59	29.54

	Well	Field	Cell	Top	Left	ObjectAvgIntenCh1	TargetAvgIntenCh2	TargetAvgIntenCh3
49	A1	1	49	329	199	126.51	7.23	413.54
50	A1	1	50	339	102	166.47	3.55	402.75
51	A1	1	51	342	459	133.78	6.02	0.00
52	A1	1	52	368	50	135.64	4.33	306.00
53	A1	1	53	370	33	122.44	4.45	383.84
54	A1	1	54	374	424	124.32	2.73	291.00
55	A1	1	55	379	299	164.56	4.66	305.00
56	A1	1	56	386	317	168.34	4.71	307.60
57	A1	1	57	394	467	116.78	1.91	297.30

Fig. 2. Example HCS data visualization with interactive tools for reviewing data with drill-down capabilities from the plate and well level (top display) to cell level together with links to images and graphical image overlays (bottom display) (Courtesy of Cellomics, Inc.).

9.1. Data-Level Integration

In data-level integration, HCS data is usually exported to third party systems, for either archive or "warehousing" purposes. Often data is imported from the same third party system so that HCS data can be annotated appropriately so that links may be made. An example of this is linking HCS data to sample information (e.g., chemical compounds or biological test samples). Often the user has centralized systems for collating all instrument or discovery data together, such as IDBS ActivityBase (www.idbs.com) or other laboratory information management system software. Linkage at the data level via an export is a simple means to deliver HCS data into the enterprise as well as integrate HCS data into laboratory workflows. The informatics architecture therefore needs to support both the necessary relational data structures to permit annotation, such as sample identifiers for compounds. In order to push data into the enterprise and link it in, requires flexible, format neutral export tools. Over the past 2–3 yr XML (eXtensible Markup Language) *(9)* has arisen as the format of choice for data export, as it is self-describing text (i.e., not only does it contain the data to export but a description of the data in the same file [metadata]). Virtually any software can interpret XML and it can be translated into other formats, if necessary. Data-level integration has certain advantages in that it is relatively straightforward to implement, almost any data can be integrated, and few changes, if any, are required by either the source or target applications. Disadvantages are that an additional copy of the data is made and there may not be a way to actively link content (e.g., see an interesting data point and wish to see the associated image without further programming).

9.2. Database Integration/Federation

In this integration type, HCS data is either (1) directly integrated or published into a data warehouse with other discovery data sources, loader scripts or database views are used, and data is often cleansed or (2) some middleware software is used as an abstraction layer to more loosely "federate" for example HCS databases with genomics, and cheminformatics databases. Middleware layers, often called metalayers, provide consumers of data with a single "view" on the data, independent of the native data format or schema, so that a user application can query and work with data across perhaps dozens of data sources, be they relational databases or unstructured data such as text files and images. The integrated data warehouse approach to database integration does have some advantages in that it is relatively simple to implement and there are now sophisticated data warehousing tools for carrying this out, however, as the desire to integrate more data sources grows, the system has to scale and this requires hands on effort. The volume and complexity of HCS data is also a consideration when building a data warehouse/integrated data integration. Using a middleware or metalayer approach to federating databases became popular particularly during the late 1990s in the bioinformatics revolution, as sophisticated data analysis tools needed to look across many data sources. Several life science and informatics vendors use this kind of technology. Such approaches have more of merit as no data gets copied anywhere and the data sources stay intact. It is also much easier to make links between data. In addition, the metalayer can be "smart," being able to semantically interpret data queries, for example. HCS data can certainly be federated using this approach, providing the advantage of best in class management of the large volumes of complex data with the ability to more actively link this data with other key discovery data sources. However, this kind of integration comes at a price, adapters have to be written for every data source, which demands an intimate knowledge of the database schema and business logic of the source data. Performance of the metalayer when querying across dozens of disparate data sources can also be an issue. If the schema of the source changes then the adapter has to be updated.

9.3. Application/Software Integration

The third category of integration focuses on more of a programmatic integration (i.e., an application programming interface [API] rather than a pure data integration). Data might well be

a result of the integration, but the primary point is that some third party application requests either data or a function to be performed by the source software. For example, a third party application might ask for an image to be analyzed or for an HCS experiment to be statistically evaluated, sending the data to a visualization application such as Spotfire. From a user perspective, the user is working with an application that perhaps spans several functions (e.g., gene sequencing, proteomics with an HCS analysis being just another choice). From an IT perspective, applications, and workflows involving HCS data can be built as need dictates. No special database is needed, no metalayer adapter, no knowledge of the underlying schema is required and no copy of the data needs to be made.

From an informatics architecture perspective and in terms of integrating HCS data, workflow, and business logic into the life science enterprise, the API integration has considerable merit. However, traditional APIs are often compile time code and so changes by the API vendor force a change to the calling application, in addition the API might only work with a limited set of programming languages or tools. Recent advances in web services ***(4)*** overcome many of the disadvantages of using traditional APIs. Web services are part of a more distributed, federated approach to data/application integration that does not require programmatic integration in the traditional sense. External applications are seen as services (irrespective of location or hardware), which are "consumed" by other applications. System architects can then build very powerful systems based on a loose coupling of web services in a so called "service oriented architecture" ***(3)***. Given the data volumes and the emerging business, and workflow of HCS, exposing this functionality through a web service is considered as the best practice for integrating HCS into life science workflow. Furthermore, this approach fits very well with "workflow" software such as that provided by Scitegic's PipelinePilot (www.scitegic.com). Workflow software allows large scale integration of business functions (rather than data) to achieve an end point, one could envisage such an application, looking at genes of interest for a particular group of targets, analyzing the literature, finding the appropriate RNAi, interpreting the subsequent RNAi experiments using HCS, evaluating the proteins of pathways knocked out and suggesting compounds likely to have an effect. All this could be achieved if all the functions were available as services that can be coupled as needed.

10. Summary

Advances in various HCS technologies, including new cell-based assays, imaging algorithms, and higher throughput instrumentation, have created an explosive growth of available HCS data. The massive volumes of information-rich data being generated by HCS systems and the effective management and use of this data has evolved into one of the most pressing issues organizations face today. The success of HCS is now more heavily linked to informatics capabilities than ever before. In the same way that the impact of genomics was greatly enhanced by bioinformatics, so HCS requires its own unique informatics infrastructure and tools. Getting access to the huge volumes of information rich HCS data, managing it, sharing it, analyzing it, and, in general, effectively using it, is critical to the overall success of the field of HCS as a whole. Informatics is and will continue to be a key critical technology for the ongoing success and widespread acceptance of HCS.

11. Notes

1. *Example data volume calculations:* in a typical drug discovery screening scenario, the image data generated for a plate with 96 wells, with four fields per well, three images per field, and one timepoint, and an image size of 0.5 MB (512 × 512 × 2 bytes) would be about 1152 (96 wells × four fields/well × 3 channels/field) images requiring about 576 MB (1152 × 0.5 MB) of storage (uncompressed). If 100 cells per field are selected with 10 features per cell calculated, then 384,000 (96 wells × four fields/well × 100 cells/field × 10 cell features/cell) cell feature records would be required. If 50 well features are calculated per well, then 4800 (96 × 50) well feature records would be required.

Depending on the speed of the HCS reader, such a plate could typically be analyzed in 10–30 min. If we use 20 min as an example scan time, and 48 plates being analyzed in a 16-h period per day, this results in a need for 55,276 (48 × 1152) images requiring about 27.64 GB (48 × 0.576 GB) of storage and about 18,662,400 [(48 plates × 384,000 cell features/plate) + (48 plates × 4800 well features/plate)] cell and well feature records requiring about 600 MB (18,662,400 × 32 bytes per record) per day. (The record size for storing features in a database depends on the database technology used and the specific implementation—for this example we use 32 bytes per feature record). In a 5-d period, this results in 276,380 images requiring about 138 GB of space and 93 million features requiring about 3 GB of storage. In 1 yr this translates into over 13 million images and over four billion cell and well feature records with combined storage requirements of over 7 TB. In a production screening operation, where multiple plates could be scanned in parallel on three or four HCS instruments operating continuously, the storage requirements could easily exceed 25 TB per year.

References

1. Giuliano, K. A., Haskins, J. R., and Taylor, D. L. (2003) Advances in high content screening for drug discovery. *Assay Drug Dev. Technol.* **1,** 565–577.
2. Comley, J. (2005) High content screening—emerging importance of novel reagents/probes and pathway analysis. *Drug Discov. World*, **6,** 31–53.
3. Erl, T. (2004) *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*, Prentice Hall, Upper Saddle River, NJ.
4. Newcomer, E. (2002) *Understanding Web Services: XML, WSDL, SOAP, and UDDI, First ed.,* Addison-Wesley Professional, Boston, MA.
5. McConnell, S. (2004) *Code Complete,* Second ed., Microsoft Press, Redmond, Washington.
6. Bass, L., Clements, P., and Kazman, R. (2003) *Software Architecture in Practice, Second ed.,* Addison-Wesley Professional, Boston, MA.
7. Han, J. and Kamber, M. (2001) *Data Mining: Concepts and Techniques,* Morgan Kaufmann Publishers, New York.
8. Adriaans, P. and Zantinge, D. (1996) *Data Mining,* Addison Wesley, New York.
9. Harold, E. and Means, W. S. (2004) *XML in a Nutshell, Third ed.,* O'Reilly, Sebastopol, CA.