## ONLINE METHODS

### Software Availability

Software, user-manuals and periodic updates are available for online download. Materials regarding SDCubes can be found at: http://www.semanticbiology.com/software/sdcube and ImageRail at: http://www.semanticbiology.com/software/imagerail.

**Details of SDCubes**

SDCubes store experimental data from various bioassays in HDF5 file format with an XML file describing how each sample was treated experimentally. Details of the HDF5 structure are described in the Results section and additional information can be found in the accompanying SDCube Programming Library manual. Four types of information are recorded in the XML component of an SDCube, conforming to an XML schema specification. The first consists of standard information such as date, experimenter, research group, etc. The second describes the experimental protocol, including information on cell lines, reagents, drugs and environmental factors. Existing minimum information standards such as MIAME (Minimum Information About a Microarray Experiment) and MIACA (components of MIBBI) are used where possible[12], and integration of other relevant XML formats such as CML (Chemical Markup Language)[33] will be straightforward. The third type of information relates to experimental design; that is, how variables in the protocol such as time, perturbation or assay are applied to each sample in the experiment. Design is the primary determinant of data cube dimensionality but relatively little effort has been devoted thus far to making experimental design computer-readable. All SDCubes currently conform to our MIDAS standard for experimental design[27], but we anticipate development of other machine-readable experimental designs in the future. The fourth block of XML-encoded metadata specifies the identities of all algorithms and free parameters used during conversion of raw data into useful experimental measurements; these include algorithms for cell tracking, background subtraction, intensity normalization and descriptor calculation. Given the potential complexity and heterogeneity of such algorithms, we intend to index them in XML using URIs or persistent unique digital identifiers (DOIs, as applied to journal articles), but it is also possible simply to embed equations or software code in HDF5 files.

**ImageRail Software**

Many image analysis software packages are already available in both the commercial and academic domains, and we have designed ImageRail to be interoperable with key open-source applications (**Supplementary Table 1**). ImageRail follows an "overlapping modular design" to create an application with novel capabilities but sufficient functionality in common tasks, such as metadata entry, image segmentation, feature extraction and image and data visualization, to function as a stand-alone application. Additional functionality is acquired through the use of existing software (**Supplementary Fig. 8**). For example, ImageRail imports standard TIFF images from microscope control software, uses Java-encoded analysis algorithms and exports slices of HDF5-XML data as CSV files for analysis by software such as Excel or Spotfire. We also expect to build a link to CellProfiler[29] to enable use of its image-processing engine and storage of the resulting data in SDCubes. ImageRail can also export data slices in CSV-MIDAS format to be interoperable with DataRail[27], a software package we previously developed to manipulate multi-dimensional biochemical data and construct regression models. DataRail currently does not use SDCubes, but we are writing a new version that will. Integration of ImageRail with other software packages requires Java programming through the use of the provided API. The overall goal of the integration effort is to leverage a rich set of existing software and to allow ImageRail to fit into existing data workflows. Conversely, interested developers can use the provided SDCube Programming Library to create new software for flow cytometry[34], protein arrays[35,36] and multiplex immunoassays[37], or even for non-biological data. Refer to the ImageRail user manual (**Supplementary Note 2**) for specific software instructions.

**Cell treatment and immunofluorescence staining**

Cells were plated at 7500 cells/well in 96-well microscopy plates (Corning) in recommended media for 24 hours, and then starved in media lacking serum for 16 hours. Cells were pre-treated for 10 minutes with 10-fold stock solutions of gefitinib (LC Laboratories) or MEK inhibitor PD0325901

(Selleck Chemicals) and treated with 10-fold stock solutions of EGF (PeproTech) for the indicated amounts of time. Cells were fixed in 2% paraformaldehyde for 10 min at room temperature and washed with PBS-T (Phosphate Buffered Saline, 0.1% Tween 20). Cells were permeabilized in methanol for 10 min at room temperature, washed with PBS-T, and blocked in Odyssey Blocking Buffer (LI-COR Biosciences) for 1 hour at room temperature. Cells were incubated overnight at 4 ºC with antibodies specific for pT202/pY204 ERK1/2, pS473 Akt, or pS73 cJUN (Cell Signaling Technology) diluted 1:400 in Odyssey Blocking Buffer. Cells were washed three times in PBS-T and incubated with rabbit-specific secondary antibody labeled with Alexa Fluor 647 (Invitrogen) diluted 1:2000 in Odyssey Blocking Buffer. Cells were washed once in PBS-T, once in PBS and incubated in 250 ng/ml Hoechst 33342 (Invitrogen) and 1:1000 Whole Cell Stain (blue; Thermo Scientific) solution. Cells were washed two times with PBS and imaged in an imageWoRx high-throughput microscope (Applied Precision). The microscope had a 10x objective and 12-bit camera sensor under 2x2 binning giving 1024x1024 pixels per image with final spatial resolution of 1.48 μm per pixel. Microscopy exposure times were 1 second for the 647 nm fluorophore (far red) and 0.012 second for the Hoechst (blue) channel. ImageJ software was used to compute the linear color scaling of [10,100] in the red channel and [10,1500] in the blue channel an enable creation of the multi-channel pseudo-colored images shown figure 5b. Bioplex assays were performed as previously described[38]. Data for Supplementary Fig. 5 was plotted using DataPflex[39].

| Supplementary File | Title |
|---|---|
| **Supplementary Figure 1** | Details of calculations for the data demands of full genome RNAi image-based screens |
| **Supplementary Figure 2** | HDF5 structures for SDCubes for various biological assays |
| **Supplementary Figure 3** | Example of artifact filtering using dynamically-linked scatter plots |
| **Supplementary Figure 4** | Details of segmentation procedure implemented in ImageRail |
| **Supplementary Figure 5** | Additional high-throughput pharmacology data stored in SDCubes |

| | |
|---|---|
| **Supplementary Figure 6** | Agreement between drug dose-response curves computed using microscopy data and ImageRail or using conventional biochemical assays |
| **Supplementary Figure 7** | Details of curve fitting and the extraction of IC10, IC50 and IC90 values |
| **Supplementary Figure 8** | Schematic of the interface between ImageRail and SDCubes software with other programs |
| **Supplementary Table 1** | References to image analysis tools |
| **Supplementary Note 1** | SDCube programming library manual |
| **Supplementary Note 2** | ImageRail software user manual |
| **Supplementary Software 1** | SDCube Programming Library 1.0: Java-based programming library to read and write data in the SDCube format. |
| **Supplementary Software 2** | ImageRail 1.0: Image analysis software for high-throughput microscopy using SDCubes for single-cell and experimental design data management |