Recent research has shown the need for tools that allow for the presentation of microarray data in the context of biological pathways. We present such a tool, VitaPad, which uses a sophisticated user interface and high-resolution graphics to create visually appealing pathway diagrams and incorporate microarray data with them. VitaPad has the advantages of using force-directed graph-layout algorithms to automatically render pathways and using an open XML format to store visual and conceptual pathway data. Most significantly, VitaPad differs from other pathway visualization tools in its flexibility and extensibility; biological knowledge and rendering information are loosely coupled, allowing new classes of pathway-related data to be readily incorporated with existing models.

The user interface for VitaPad is primarily mouse-based and relies on a point-and-click visual editor and a variety of dialogs and menus for visual editing. Entities within a pathway, such as compounds, reactions and enzymes, can be retrieved from the thousands of entities contained in our iSpar database or can be added by the user. Visual editing dialogs allow the appearance of these elements to be customized by changing attributes such as color, shape and font. A thumbnail viewer within the dialogs gives a preview of how the element will appear. User-defined visual types can be saved for future use. Experimental data can also be gathered from hundreds of datasets in the iSpar database or imported directly. Data values are displayed using a customizable gradient. The VitaPad graph viewer takes advantage of its awareness of connections between elements to make editing easier and faster. For example, dragging one vertex automatically repositions all edges associated with the vertex in real time. The viewer also allows panning and zooming of pathway graphs.

To automatically generate diagrams for conceptually defined pathways, our goal is to use an algorithm that optimizes the positions of vertices such that vertices that are conceptually adjacent are rendered accordingly, and such that the number of edges that cross each other in the plane is minimized. The most effective layout algorithm in this regard, we feel, is a force-directed or spring-embedded layout. This layout uses an optimization process that, in principle, treats the edges as springs whose tension is proportional to their length in the diagram and configures the vertices in a way that the overall tension of the diagram is minimized. Specific parameters for this algorithm can be configured from within the program.

VitaPad uses an open XML format as a means of saving and transferring pathway graphs. This format, tentatively called VPML, stores both biological and graphical details from the pathway graph. It has the advantages over the binary formats used by most other pathway rendering programs that it can be easily interpreted on multiple systems and readily integrated with other research tools.

VitaPad deals with two main types of data for a given pathway: conceptual data, such as genes, compounds and expression values and visual data, such as shapes and colors. It is an important feature of our program design that these types of data are coupled as loosely as possible. Scientific concepts are not hard-coded into the rendering algorithm and visual details are not stored in the database of pathway entities. Because of this, VitaPad can be made aware of new classes of data without forcing changes in the application code. All that is required is to define a rendering specification for the new class of data. Data is persisted using the EAV/CR schema, a database design that relies on developer-defined metadata to indicate how classes of data and their attributes are presented (Nadkarni et al. 1999). This schema is particularly well suited to data definitions, which, like pathways, are rather subjectively defined and are constantly exposed to new information and discoveries.

VitaPad is written in Java and is deployable on all major operating systems, including Windows, Macintosh and all flavors of Linux/Unix. VitaPad is licensed under the GPL and requires the presence of Java Runtime Environment (JRE) 1.4 or greater. Source and binaries are available for download at http://bioinformatics.med.yale.edu/vitapad.

References
