Getting the Most out of Laboratory Automation: Instantaneous Cherry Picking

By Alan H. Katz, Chief Scientific Officer
Hudson Control Group, Springfield, New Jersey

Increasing competition and economic pressure require drug companies to maintain a strong, evenly distributed stream of new drug applications. Given high attrition rates throughout the research and clinical cycles, companies must simultaneously process a huge number of exploratory and discovery projects. This increased load is being handled by a fixed number of people trying to meet similar time pressures; most often with little, or no, budget increases.

Although automation of high throughput screens has vastly increased the speed with which leads can be identified, there are still processes in the sequence that are inefficient. The biggest time losses are associated with follow-up screens, such as confirmation tests, or selectivity assays and other downstream tests. For example, most screens are run as single-dose, so potential hits need to be re-screened at multiple concentrations to generate dose-response curves. Often potential hits need to be submitted to a number of additional screens to check selectivity and avoid false positives.

Here we describe how modifications to laboratory automation protocols can be applied to increase the efficiency of the screening process. By adding programming logic into an automation protocol, it is possible for the software to analyze results as they are generated, and then prepare any given sample for an immediate second screen, based on these results. In addition to faster turnaround, this method benefits by using the same reagents and samples thereby avoiding variances that could result upon retesting.

Workcells to Support Automatic Cherry Picking

First, select and setup the all of equipment from which the scientist is likely to choose. This makes it possible to add steps to a protocol as part of a secondary screening strategy. To maximize flexibility, one should follow two principles: 1) include all of the equipment that might be needed, and 2) make sure a microplate can be readily transported between any two pieces of equipment. In designing systems at Hudson Control Group, we also like to make sure each instrument is easily accessible to the user, so that it may be used in one-off, manual experiments.

Biological Assay Screening System

The system in Figure 1 was designed to carryout a large number of screens normally encountered in drug discovery research. A multimode reader supports the vast number of protocols involving fluorescence intensity, polarization, and resonance energy transfer (FRET), as well as AlphaScreen™, ELISA, luminescence and UV/VIS absorption-based assays.

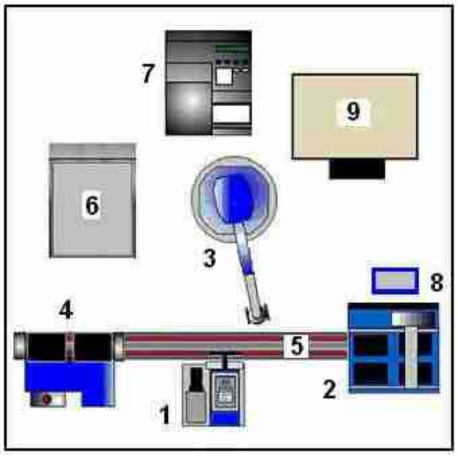

Figure 1. System Configuration for Biological Assay (Hudson VaryScreen™ I) includes: 1. Micro10™ 12-Channel Dispenser; 2. SOLO Single Channel Pipettor; 3. PlateCrane EX Robotic Arm; 4. StackLink™; 5. TrackLink™; 6. Multi-Mode Reader; 7. ELx405 Washer; 8. TeleShake Microplate Agitator; 9. STX40 Incubator

All drug screen protocols involve liquid handling steps to add reagents and test compounds. Typically, the potential lead will be added to only one or two wells, a process better suited for a single channel pipettor. Most other reagents are added to each well of a plate in the same quantity, so a multi-channel dispenser is most appropriate. The system illustrated includes a multi-channel dispenser and an automatic pipettor.

Microplates are transported among the instruments by robotic arms, and moving tracks. Instruments that are frequently used in sequence benefit from contiguous placement along a track. Other components, and those that can’t be adapted to tracks, must be within reach of the robotic arm. Microplates are stored in a searchable stacking system connected to the track. The system also contains an automatic incubator a microplate shaker nest, and a microplate washer accessible to the PlateCrane robot arm.

Software Integration and Protocol Design

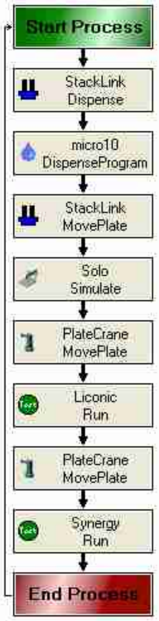

Figure 2. Simple protocol represented as a flow chart.

Several manufacturers of automation equipment have developed software to schedule the sequence of activities within a workcell, like the ones described above. The software examples described here were developed using SoftLinx™ for Systems, which was developed at Hudson Control Group. For each piece of equipment contained in the workcell, there is a corresponding software interface that SoftLinx automatically loads. In some cases, the equipment is controlled exclusively by its own software, while other equipment support additional levels of configuration, such the precise location of microplate nests and the type of plates and pipette tips being used. Once complete, SoftLinx is sufficiently aware of the system to make it ready for protocol design.

Software protocols follow the same logic as in the laboratory. In the simplest case, the entire protocol is represented by a single process, such as the BCA UV protein assay shown in Figure 2. A microplate is first delivered to a liquid dispenser where reagents are added, then moves to the automatic pipettor where test samples are added. The robot arm then carries the microplate to the reader.

More commonly, a protocol consists of a series of parallel processes, each focusing on a particular portion of the method. Each process can be set to start as soon as the protocol is run; or it can be set to wait until any number of conditions is met, such as:

- The presence of a plate in a particular instrument nest,
- The identification of an active compound
- After a number of identified hits is reached
- After another process is complete.

In a typical ELISA assay, there are a number of processes that begin when a plate reaches the incubator, each following different paths based on the stage of the protocol.

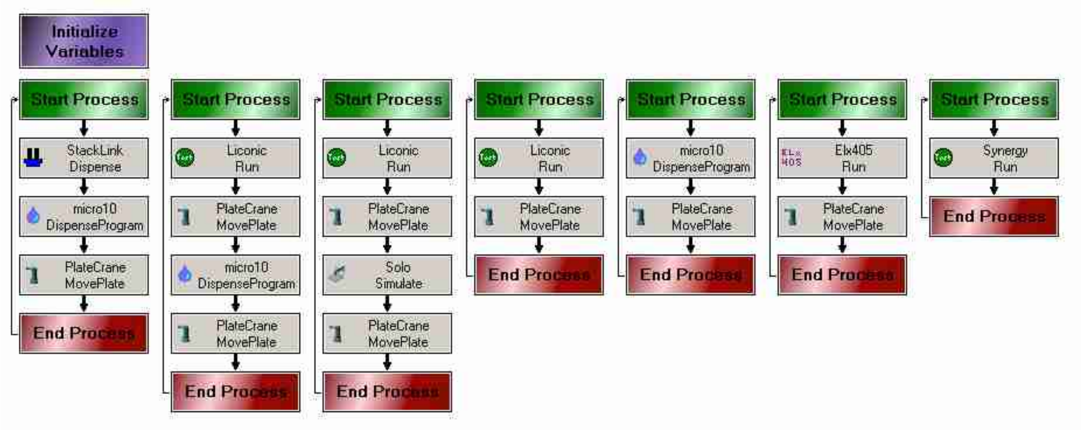

Figure 3: A typical ELISA assay showing a number of parallel processes.

Programming Logic and Smarter Protocols

Many frequently used protocols can be automated with the set of tools just described. However, one can introduce programming logic into a process and produce more powerful and flexible protocols. Global variables can be declared so that every process can be aware of the current state of the entire system. A Boolean-like variable can set as false until a condition is met, or a desired result is obtained. A numeric variable can be used to keep track of how many times a particular plate has been washed, or the number active compounds have been identified. As with conventional structured programming, "If...", "Then...", "Else…" conditional statements allow one to control how a protocol proceeds based on the value of any available variable. Similarly, the "While…" statement allows portions of a process to continue (or stall) until the value of a variable changes.

These constructs add a great deal of flexibility in designing complicated flow charts, but the individual steps are limited to methods and settings of the various instruments in the workcell. In order to carry out "on-the-fly" data analysis and make choices based on incoming results, one needs to add a scripting feature. In the case of SoftLinx, user scripting is supported with a full-featured interface to Microsoft's Visual Basic for Applications (VBA).

Introducing Cherry Picking into Protocols

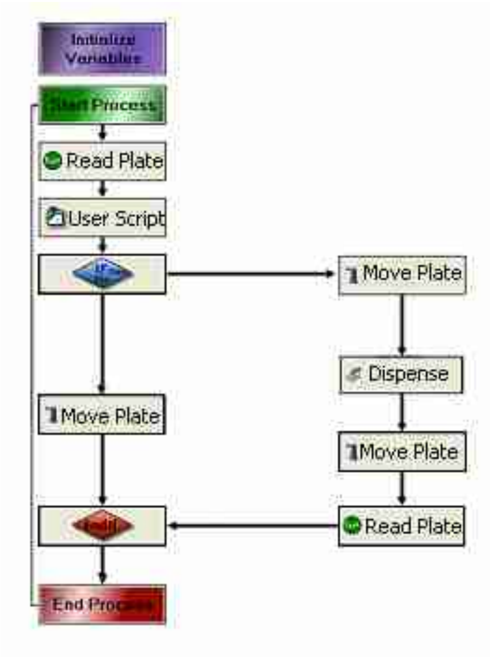

Figure 4. Conditional logic in a lab automation process.

In the code snippet in figure 4, we introduce a user script after submitting a microplate to a fluorescence reader. The user script contains the code necessary to import the text file generated by the reader and checks each result to see if any sample on the plate shows significant intensity to warrant further attention. A variable is then set to true or false depending on the outcome. The If…Then…Else… statement that follows will continue straight down the flowchart if no hits were found, and the robot arm will remove the plate from the workcell. However, if a positive result is found, the plate is taken in a different direction for further analysis. In this example, the robot arm moves the plate to the dispenser, where an additional reagent is added. The plate is then moved back to the reader and an additional reading is taken.

If the density of hits is high enough, it is likely that a hit will be found in every plate, and the above procedure would require *every* plate to undergo a second screening, when, in fact, only 5% of the samples need to be considered. We can combine the implementation of user scripts and conditional logic with multiple processes to arrive at a much more efficient process. In this case, the user script inside the If…Then…Else… statement simply keeps track of the exact location, and number of each hit. A second process is added, which is set to start when the total hit count reaches the number needed to fill a new plate. The process then searches the stacks for the plates containing the hits, and systematically builds a new plate, along with the appropriate amount of serial dilution for computing dose-response curves.

Using this approach, we are preparing cherry-picked plates for secondary screening even while the primary screen is in progress. Thus, the secondary screen can begin immediately upon completion of the primary screen, or, if SoftLinx is employed, both screens can be run simultaneously.

In addition to dose-response curves, this procedure can be used to submit hits of interest to second-generation screens that are too slow to be considered for full library screening, such as flow cytometry, FLIPR or automated patch clamping. For example, it would be impractical to use flow cytometry to screen hundreds of thousands of compounds for the ability to block the interaction between a cell surface protein and a fluorescent-tagged antibody. However, it would be more plausible to wash off unbound antibodies and measure fluorescence with a standard reader. Only samples that show reduced fluorescence would need to be submitted to the flow cytometer.

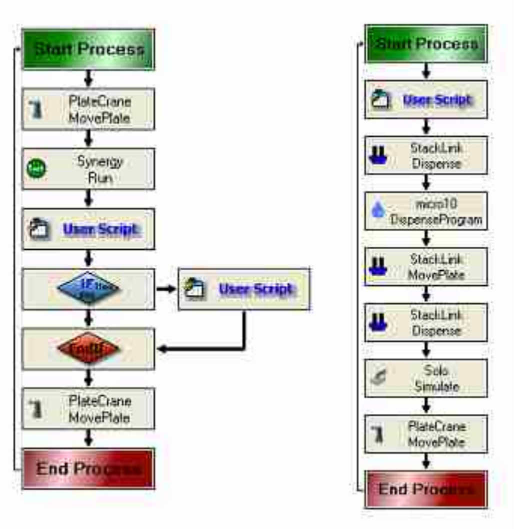

Figure 5. Cherry picking process 2 begins when called by the user script of process 1

Cherry Picking Assay Conditions

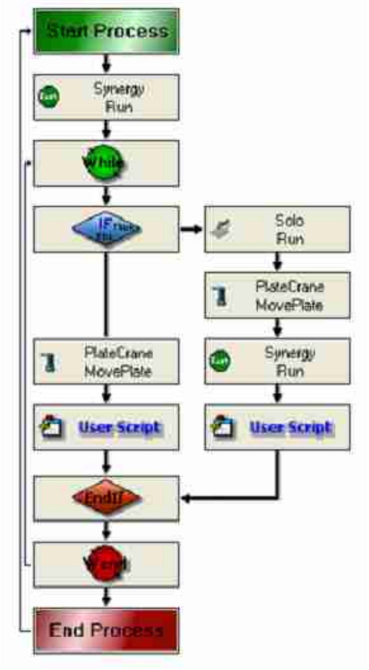

Figure 6. Sample assay optimization process.

This logic can also be applied to the optimization of assay conditions. In this approach, we select a threshold value for a standard sample. The process begins with a Boolean variable set to False. We then enter a “While…While-End” loop set to run until the Boolean is True. A user script is then employed as described above to extract and analyze the results from the fluorescence reader. The conditional statement that follows checks the fluorescence intensity, but in this case, an unsatisfactory result leads to a change in the amount of additional reagent and the sample is submitted again to the reader. This process continues until a satisfactory result is obtained and the Boolean is changed to True. The user script can keep track of the number of passes through this loop, and calculate the effect of added reagent on the reading – and stop the process if there is no improvement. One can optimize a variety of assay conditions by introducing a similar loop for each condition of interest.

Conclusion

The role of software has long been overlooked as tool for optimizing automation and efficiency of highly automated high throughput laboratories. Here we described a few applications of SoftLinx for Systems and demonstrate the incorporation of simple logic enabling testing to proceed to the next logical step automatically and immediately.

Author Bio

Alan H. Katz is the Chief Scientific Officer at Hudson Control Group in Springfield, New Jersey. Previously a Principal Research Scientist in Medicinal and Computational Chemistry at Wyeth Research, Alan’s interest now lies in the development of new lab automation applications to help pharmaceutical and research organizations efficiently conduct their high throughput screening as well as genomics and proteomics research. Alan graduated from State University of New York at Buffalo with a BA in chemistry and Princeton University with a PhD in organic synthetic chemistry.