

Leaf LIMS: A Flexible Laboratory Information Management System with a Synthetic Biology Focus

Thomas Craig,^{*,†} Richard Holland,[‡] Rosalinda D'Amore,[§] James R. Johnson,[†] Hannah V. McCue,[†] Anthony West,^{||} Valentin Zulkower,[⊥] Hille Tekotte,[⊥] Yizhi Cai,[⊥] Daniel Swan,^{||} Robert P Davey,^{||} Christiane Hertz-Fowler,[§] Anthony Hall,^{||} and Mark Caddick[†]

[†]GeneMill, University of Liverpool, Liverpool L69 7ZB, U.K.

[‡]New Forest Ventures Ltd., Fordingbridge SP6 1QX, U.K.

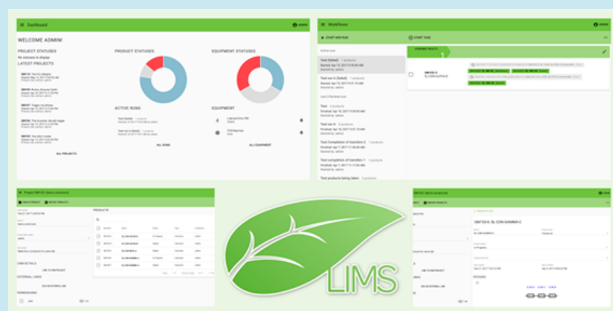
[§]Centre for Genomic Research, University of Liverpool, Liverpool L69 7ZB, U.K.

^{||}Earlham Institute, Norwich Research Park Innovation Centre, Colney Lane, Norwich NR4 7UH, U.K.

[⊥]Edinburgh Genome Foundry, Michael Swann Building, Max Born Crescent, Edinburgh EH9 3BF, U.K.

ABSTRACT: This paper presents Leaf LIMS, a flexible laboratory information management system (LIMS) designed to address the complexity of synthetic biology workflows. At the project's inception there was a lack of a LIMS designed specifically to address synthetic biology processes, with most systems focused on either next generation sequencing or biobanks and clinical sample handling. Leaf LIMS implements integrated project, item, and laboratory stock tracking, offering complete sample and construct genealogy, materials and lot tracking, and modular assay data capture. Hence, it enables highly configurable task-based workflows and supports data capture from project inception to completion. As such, in addition to it supporting synthetic biology it is ideal for many laboratory environments with multiple projects and users. The system is deployed as a web application through Docker and is provided under a permissive MIT license. It is freely available for download at <https://leaflims.github.io>.

KEYWORDS: lab information management system, synthetic biology, data management, LIMS, automation



Synthetic biology takes an engineering focused approach to the manipulation of genetic information within an organism. Functional DNA parts are identified, characterized and converted into a format that allows them to be combined into multiple combinations as functional constructs with predictable properties.^{1,2} Using Golden Gate, Gibson and similar assembly methods³ these parts can be used to efficiently build variations of a construct allowing for rapid assembly, prototyping and optimization of constructs, including complex gene clusters.⁴ In addition, such parts can be distributed, either as a toolkit or individual parts, and thus can be integrated easily into other projects. A significant amount of data is produced during these processes, which can then be used to inform decisions in the design–build–test cycle, for instance through automated validation of constructs.

Combining the part-based assembly with the increasing use of robotic equipment unlocks the potential for a high-throughput construct creation, transfection into, and analytical measurement of, the recipient chassis organism. Automated laboratory equipment can quickly and repeatedly perform the tasks required to assemble a construct, often using smaller volumes, therefore making it both cost and time efficient.⁵

This increase in scale also leads to an increased level of complexity with respect to parts, processes and data, all of which need to be recorded and tracked within the laboratory. With a single construct it is relatively easy to keep track of what is required, something that swiftly becomes difficult as variations expand to encompass multiple combinations of components to formulate and optimize, for example, entire gene clusters (Figure 1). A key issue is how to achieve the necessary management of this information; tracking the usage of the various parts and transitory elements through the process. The answer to this is the development and use of a laboratory information management system (LIMS) to manage the production of the constructs and maximize efficiency.

Information management systems already exist for various other disciplines, such as MISO⁶ which has been designed for next generation sequencing (NGS) workflows and PIMS⁷ for managing protein production. While it should be possible to adapt an existing LIMS to handle this complexity, it is more effective to design a system from the ground up that can be used as a single package rather than following instructions to

Received: June 15, 2017

Published: September 6, 2017

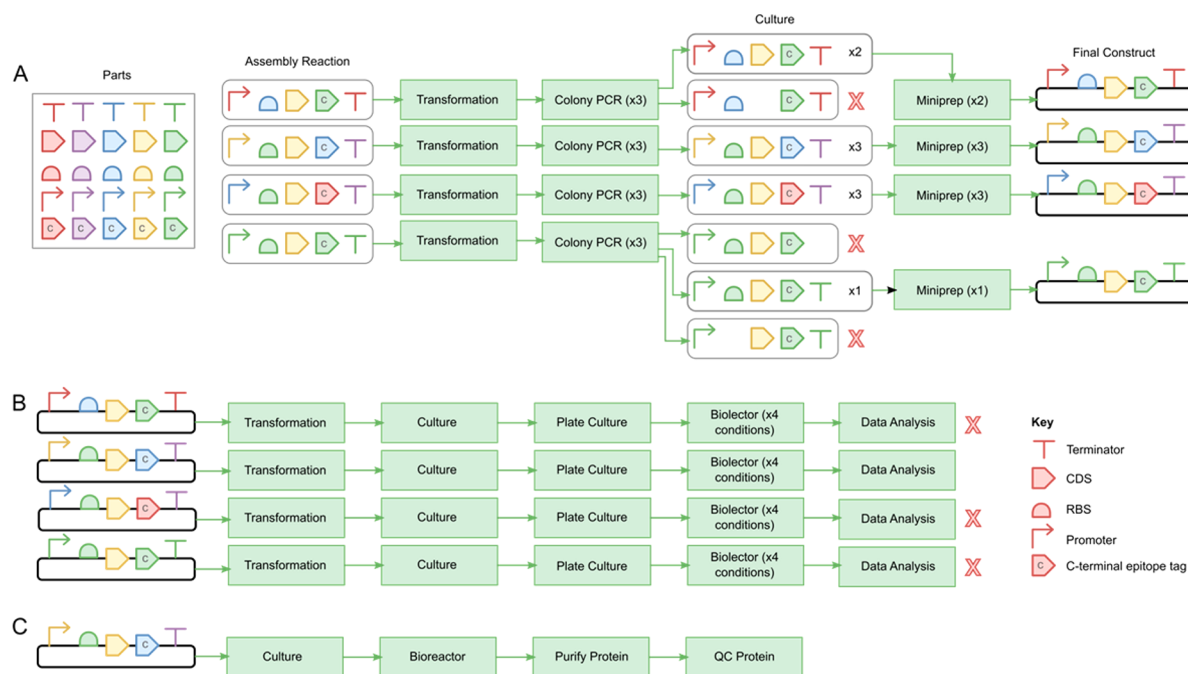


Figure 1. A simplified and condensed version of a synthetic biology workflow for optimizing protein expression. In part A multiple constructs are assembled, each a variation centered on expressing a specific coding sequence (CDS) using differing promoter and ribosome binding site (RBS), C-terminal epitope tag and terminator parts. After transformation, resulting colonies are checked by colony PCR. At the culture phase, three incorrect constructs are discarded and the remainder are used for DNA minipreps. In part B the construct DNAs are transformed into a cell line suitable for protein expression. Expression of the protein is then analyzed under a variety of conditions, in this case using a Biolector microreactor (m2p-laboratories). A significant number of conditions could be tested for each construct and may include culture medium, temperature, bacterial chassis and oxygen. The most highly expressing construct is then picked for large scale expression and purification in C. If in part B no construct had yielded the required output the process would then go back to A with new designs based on the best performing variants. This is representative of the design–build–test cycle that forms an integral part of synthetic biology.⁸ In part C the optimal construct and conditions identified in part B are taken and expressed to generate larger quantities of the required protein. The symbols used correspond to standard SBOL visual with the addition of “c” to a CDS to indicate a C-terminal epitope tag.

replicate an existing configuration (e.g., installation of plugins) in another piece of software. We therefore took this approach and created a LIMS system that is freely available for the community to use.

RESULTS AND DISCUSSION

Identifying the Need for a New LIMS. By looking at the various processes taking place in a synthetic biology laboratory setting, the following specifications were drawn up for the laboratory management system: First, multiple forms of data need to be stored (e.g., project metadata, experimental designs) and any material received, tracked (e.g., DNA Parts, design data, primers). Second, outputs produced during each process (e.g., colonies, PCR products) and the data produced by the workflows themselves require tracking as these might be utilized in later steps. Third, the LIMS needed to record information on the above, for instance validation such as sequencing or data describing functionality. Lastly, the system needed to be flexible and scalable to handle both manual and automated data entry. The system also needed to be flexibly designed such that new information, components and methods within the rapidly changing field of synthetic biology can be accommodated.

With the requirements set, existing systems were evaluated to assess if they offered the necessary features. Existing commercial LIMS such as GenoLogics Clarity⁹ are geared toward the NGS market and lacked the flexibility required to support synthetic biology workflows. This was also the case

with the open source MISO system,⁶ designed to efficiently handle NGS projects and hence would require a complete rewrite. Various other open source projects such as openBIS,¹⁰ PIMS and BikaLIMS¹¹ provide some of the feature set, such as sample and project tracking, but were again designed to suit either a more clinical setting or toward a specific purpose (e.g., protein structure analysis). JIRA,¹² an issue management tool from Atlassian, was also evaluated. While more flexible than most of the identified tools, it required too many workarounds to be used as a viable long-term solution. There are other technologies that currently show strong promise, such as Antha¹³ from Synthace. This provides an open source biology focused language for developing protocols but it requires the use of the closed-source “Antha OS” to provide a user interface and enable its use for a LIMS like system. The closed-source nature of this project therefore limits input from relevant entities such as academic laboratories. For synthetic biology, some resources have been developed such as ICE¹⁴ for storing biological sequences and BioPartsDB¹⁵ for workflow processes. These provide the individual components but lack the integration and depth required for project management. We therefore decided to develop an open source system that would fulfill the criteria demanded for synthetic biology.

Understanding How Leaf LIMS Operates. Leaf LIMS was designed first and foremost to handle information generated through laboratory operations. It takes a task-based approach, breaking down protocols and processes in the laboratory to their minimal components (e.g., steps such as an

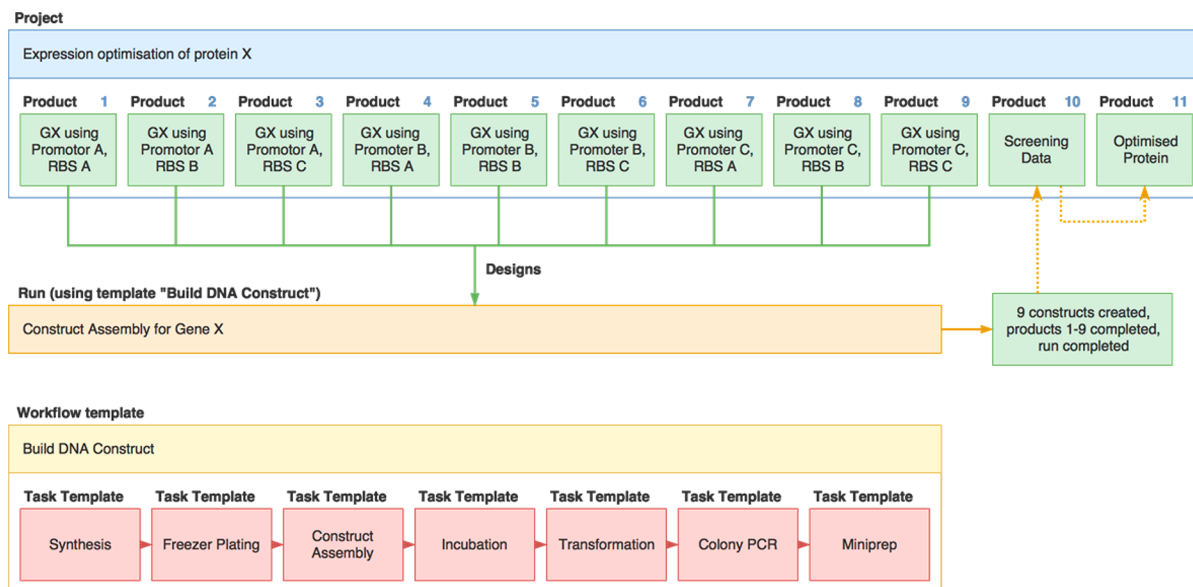


Figure 2. An example of how Leaf LIMS operates. The project, in this case to optimize the expression of protein X, has 11 products. Nine products (labeled 1 to 9) are DNA constructs; once the nine constructs are created they will then be assayed to produce the data for the screening data product (labeled 10). The optimized protein product (labeled 11) will apply the results from the screening data to select the construct which produces the most protein for optimization. For each product to be created, a run (Construct Assembly for Gene X) will take the selected products (in this case 1 to 9) and use a workflow template (Build DNA Construct) to complete each of the tasks by which it is defined. The initial information on what is required to make each product is specified using a design in one of the supported files types (e.g., csv or SBOL v2). The design files contain details of each part used (concentration, plate location and/or barcode, etc.), their order in the final construct, its unique ID and other parameters as chosen by the user, e.g., the DNA sequence. This results in the creation of nine constructs at the end of the workflow, which can now be used in the screening step.

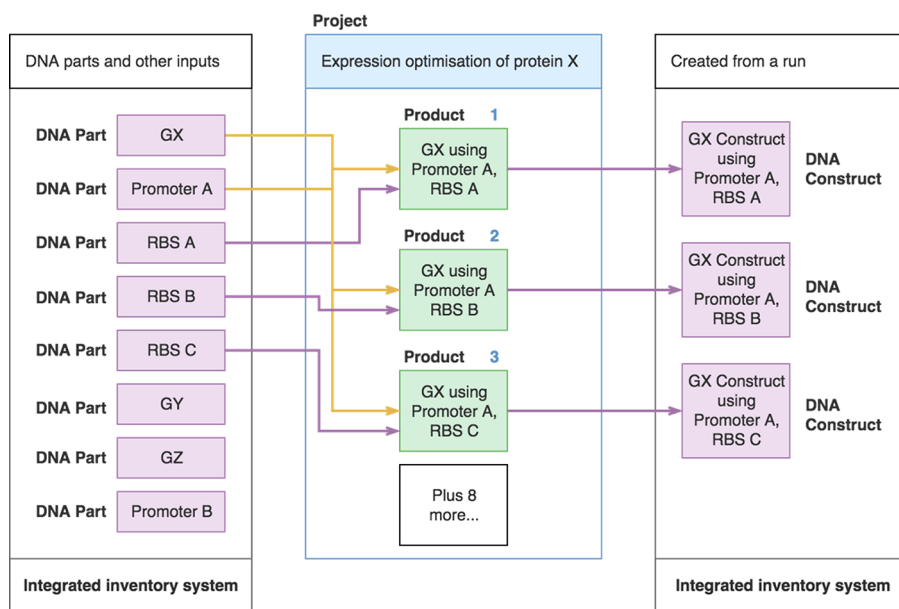


Figure 3. Linking products to the inventory. The three products shown contain designs to make different constructs expressing gene X (labeled GX), each with a different RBS. Yellow lines indicate a part used in all constructs, purple lines indicate a part used only once. All the relevant DNA parts to make these constructs are entered into the integrated inventory system as individual items. On the left are the required inputs (in this case the DNA parts) that are then linked to the products that require them. These can then be used in a task, where the respective workflow picks what is needed to make the construct. Once a construct has been produced it is automatically added to the inventory as an item of the type “DNA Construct” and linked to the product that was used to create it. All the information on the created product is therefore stored in one place.

incubation or PCR conditions) and then arranging these to build workflow templates, much like a protocol. To run a workflow requires that a product, the final output of one or more workflows, needs to be defined: for instance the DNA parts required and their arrangement or the type of data

required (e.g., screening data). These products are then grouped together into a project, a representation of related processes. A demonstration of this can be seen in Figure 2, where a project has been defined to optimize expression of protein X. There are 11 products, nine that represent constructs

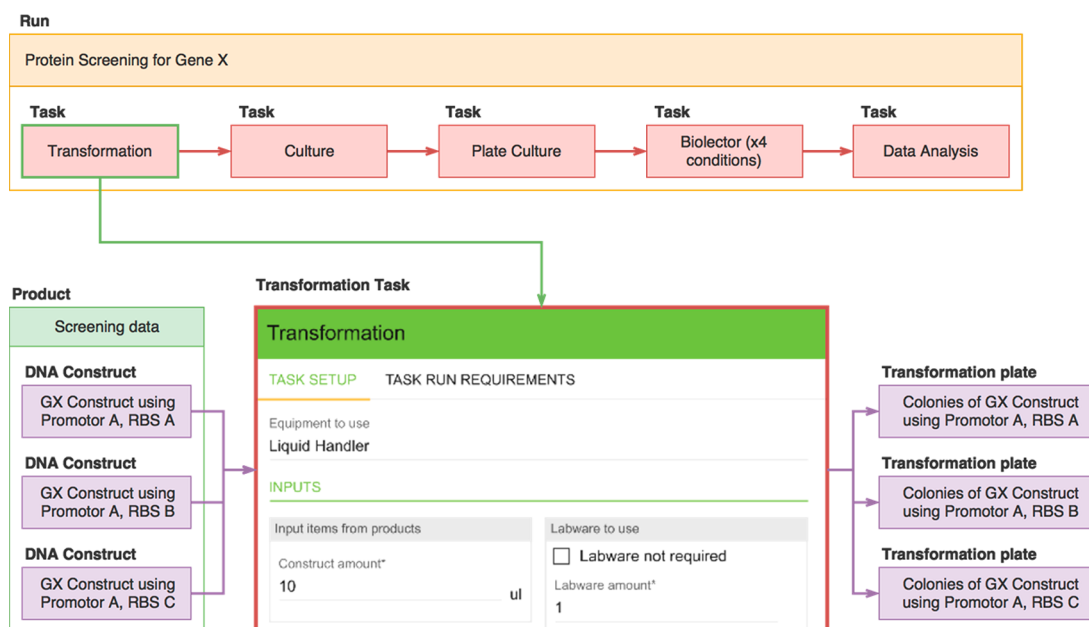


Figure 4. Starting and completing a task during a run. Three constructs, the products of an earlier workflow have been linked to the workflow for the screening data product. The constructs are automatically picked up as the run has been set up to use these products. Three items are selected from the inventory: transformation plates, SOC media and BL21 cells. These items are required to complete the task and the inventory will be automatically updated to reflect the amounts used. At this point it is also possible to make alterations to the amounts to be utilized before starting (for instance if you wanted to halve the volume of cells used). Once data has been entered and items selected, one or more files can be exported to a CSV file and then imported by the equipment to enable set up. Once the task has been completed it will generate three outputs (reflected by having three items passed to the task) of the “transformation plate” type. The plates will at this point physically contain the transformation mix ready for use in the laboratory. These are added to the inventory and reflect physical items created during the transformation step. To make it easy to find these they are also then linked to the product that created them so they can be used in the next step. Any information entered into the task is also stored (e.g., input types and amounts) and results are then copied over and linked for later use.

in *E. coli*, one set of screening data for all nine constructs, while the final product is the purified protein derived from the optimized construct. A run is started using the “Build DNA Construct” workflow template which provides the process designs for the constructs, the first nine products of this project. Each task is run until all nine constructs are produced based on the given designs. It is then possible to use these products, the nine constructs, in a different workflow template. In this example it is a protein screen to produce the output for the “screening data” product.

Managing Projects and Their Outputs. Any significant undertaking in a laboratory requires access to various sources of information, e.g., protocols and records of the progress of various constituent parts through the laboratory environment. Leaf LIMS organizes logically related work into a project (e.g., Expression optimization of protein X in Figure 2). It contains data on what is to be produced and for whom, along with any relevant information on how this is to be accomplished. A project can link to external resources (e.g., protocols and laboratory notebooks) and integrates with Salesforce CRM¹⁶ for projects that require management of financial/customer information and communications.

To provide flexibility to the system, Leaf LIMS uses products to indicate the outputs of the project. By organizing the project in this way it is possible to track every component that makes up the product in a single location in the user interface, thus ensuring that items are accounted for through the entire process. These items are tracked through an integrated inventory system and include both the items required to produce the final product (e.g., DNA parts) as well as items generated during the creation process (e.g., constructs) (Figure

3). Storing items in this manner allows for them to be shared between projects, a key requirement when using a parts-based toolkit.¹ A product does not have to be a physical item and can represent everything from a construct to protein expression data. It is possible to have an entirely data based workflow that does not produce a physical item at any point. This flexibility means that products can be tracked through the entire design–build–test cycle and ensures data is captured at every point. We also designed the facility to handle large numbers of products within each project, which is especially important with the increase in use of automation and the ease with which large numbers of differing constructs can be created.

Leaf LIMS was designed to integrate with third party tools to allow flexibility in how constructs are designed as well as allowing it to fit into existing design workflows. To this end, support is in place for multiple file formats including Synthetic Biology Open Language (SBOL v2),¹⁷ GenBank¹⁸ and CSV. The imported files are parsed to extract the relevant information required to build a product, for example a construct. These designs can be reparsed at any point to extract further information. From these data, links are created to items in the inventory system, providing information required for use in tasks and for display in an SBOL visual formatted diagram. Linking back to the inventory this way allows for combinatorial designs to utilize the same parts in different contexts. Tasks can produce outputs (e.g., a transcriptional unit) which are then included in the inventory and linked to the respective product. A design file can also be uploaded that contains components not available at the time. Subsequently, when available the system will identify them and automatically include them. For example a design could

reference multiple individual transcriptional units, each of which have been assembled as separate products with their own design. An entire project can therefore be planned in advance with all designs finalised before any laboratory work begins.

Managing Laboratory Processes through Workflows.

The intention of synthetic biology is to bring an engineering-focused approach to biology. To this end, Leaf LIMS uses task-based workflows to provide a modular system for building a consistent process (Figure 4). Each task provides a set of fields that are used to record information on inputs, values and outputs. These data are linked to the products produced by the task and cannot be modified afterward, even if the task or workflow is changed. This mode of function allows the LIMS to operate with both automated and manual data entry, thus being adaptable to differences in how laboratories operate. These tasks and workflows can ultimately be shared between laboratories, thus allowing for easy adoption of new or improved processes without requiring specific equipment.

Tasks are intended to represent the smallest possible unit of a process; these can then be combined in different orders to create workflows that can run a given process. A task consists of a number of fields that define its structure; input fields record the components of a task (*e.g.*, a buffer) while output fields produce an item in the inventory system corresponding to the output generated by the process (*e.g.*, colonies on a plate). There are also field types designed to simplify the setup of a task for the user. Calculations can be defined to compute values that can be used in other fields (for example to derive the amount needed for an input field) and in any files to be generated by the task. These calculations can refer to other fields by their names and in conjunction with variable fields can be used to easily modify common parameters for a process (*e.g.*, the annealing temperature in a PCR). Not all are fully automated and equipment often requires at least some level of human intervention. Step fields are provided for this purpose, allowing for written descriptions with the capacity to define up to four parameters (*e.g.*, for defining times or temperatures in the step). All the fields in a task are assigned a default value that can be overridden either for all items or for individual items by supplying a file containing the necessary information. Using a combination of these fields, an entire protocol can be constructed in such a way that a user, without prior knowledge of the protocol, can accurately run the task with the pertinent values.

A workflow acts as a template for the order in which tasks are to be executed. However, experiments may need to be modified and additional tasks may be required to keep pace with developments. Leaf LIMS uses the concept of a “run” to solve this issue, taking the order of tasks defined in a workflow yet allowing for new tasks to be dropped in at latter points to effect changes to an established workflow. The next task will not become available until the current task has been completed successfully. If there are failures at the completion of the current task a new run will be generated, initiating either at the current or an earlier task. Successes will result in the next task of the workflow being initiated until the workflow is complete. These decisions are currently made manually based on the results obtained but future work will incorporate automated decision making, allowing tasks to be initiated based on criteria such as responses from equipment and decision trees. Data produced from equipment (*e.g.*, sequencing results for QC) available in a network accessible location can be copied across and linked to the relevant products.

Tasks and workflows are fully configurable in Leaf LIMS with no limit to the number of tasks or workflows that can be created. They are designed to be reused or easily modified to create variations around commonly used procedures. Existing tasks can be readily adapted to new situations without impacting on any of the data that has already been gathered as this is permanently associated with the product.

Sample and Consumable Tracking. One of the issues when working in a busy, multiperson laboratory is that it can be difficult to know the availability of consumables at any one point. Leaf LIMS addresses this by providing an inventory system to track items including DNA parts and reagents. This system is flexible enough to record amounts of both labware (*e.g.*, boxes of plates) and contents of tubes (*e.g.*, enzymes, plasmids, primers). Each has an associated amount, for instance an SI unit such as microlitres. Using SI units allows for partial amounts to be used (*e.g.*, 50 μ L out of 5 mL) and remain accurate. Custom units can also be defined although these are limited to being defined as an integer. To handle other information that is not one of the core inventory fields, such as manufacturer, item properties can be defined. Each property on an item represents a user defined key with a corresponding written value and no limit to the number of properties. Any inventory item created by the system automatically provides a link to the items that created it, allowing for the iterative building of new items (*e.g.*, parts) yet still maintaining the connection to their sources. The inventory is also coupled into an alert framework, which can be used to set up conditions to monitor and provide notifications when they are reached. With this it is possible to set up thresholds at which the relevant users are notified so that the item is restocked in a timely way. The inventory system is designed so that it can work in parallel with existing dedicated inventory systems such as Mosaic¹⁹ and Open Freezer²⁰ but as yet there is no integration beyond allowing import of items already in the previously mentioned systems.

Controlling Access. Each project, product, workflow and inventory item is assigned a set of permissions that indicate who and at what level groups of users can access it. A user can be assigned to multiple groups, each of which can grant either read or write access to a project. This allows a simple but granular access to items in the system thereby allowing individuals to work within the same system without revealing access controlled information.

Using Equipment to Automate Processes. While automation is becoming more prevalent in the laboratory setting there are still limitations and almost all laboratories still require the handling of various steps by laboratory personnel. Leaf LIMS aims to strike a balance between providing a human-focused approach yet still offering the capability to simplify interactions with automated equipment. At its most simple, Leaf LIMS allows for tasks to generate files that can be used as inputs for equipment by using information defined in the task (*e.g.*, volumes, component location). Files for use as inputs to equipment are easily configurable, allowing integration with any equipment that supports text files. Eventually this will be expanded and will allow for true communications between laboratory equipment and tasks, driving the system toward minimal manual intervention. We aim to take advantage of a plugin-based system for integration with equipment, for example producing optimized task lists and directly interfacing with equipment.

Leaf LIMS also provides a machine-agnostic approach to handling the files generated by equipment. The files that are required to be copied over can be defined directly or through a template system that allows variables (such as the date) or regular expressions to be used to create the name of the file. They are then copied and linked to the relevant products when a task using that equipment is run. These can then be queried later to extract further information.

Expanding the Scope beyond Synthetic Biology. Leaf LIMS is designed primarily with synthetic biology in mind but this does not restrict it from being used for other purposes. While synthetic biology requires the handling of a variety of complex tasks other areas are as demanding in their requirements and would benefit from its flexibility and adaptability. The highly configurable nature of the system can be adapted for use in any circumstance where repeated, consistent workflows are required or where items are produced and need to be tracked; examples of this could include use in generalized laboratory settings where protocols are converted to tasks and workflows or as a tracking system for managing academic projects and their outputs and even for use in purely data driven settings such as bioinformatic analyses where projects and data need to be managed.

Future Direction. The future roadmap covers a wide range of features that will enhance the current system's utility. In the immediate future these include the creation of plugins to allow integration of available tools and extensions to the system (e.g., integration with parts repositories such as ICE¹⁴). Further work would include the integration of more third-party tools, such as the design of constructs, equipment control software to allow greater levels of automation and machine integration and a greater range of inventory management tools to boost the capabilities of the integrated system. There are also plans to increase the planning capabilities of the LIMS to allow for the estimation of costs and time involved in a given workflow.

Leaf LIMS is currently being used in three laboratories (GeneMill, EGF and Earlham), with their feedback used to prioritise the roadmap for development. This level of codevelopment ensures that Leaf LIMS is constantly on the forefront of best practices for synthetic biology and receives features that benefit a broad range of users rather than just a single laboratory. The benefits of Leaf LIMS are already being seen in GeneMill where it has reduced the amount of time taken to plan and document laboratory work. For example, when running multiple projects simultaneously, it is desirable to run as many assemblies or PCR reactions in one plate as possible. However, it can be very time-consuming to plan this manually in order to maximize the productivity of laboratory staff and/or automation. The introduction of Leaf LIMS has greatly streamlined these processes, decreasing time spent per project, improving turnaround time and reducing the cost to the user.

CONCLUSION

Leaf LIMS provides a system that is tailor-made for handling the complexity of synthetic biology workflows. It can track the myriad components required for a specific product and handles the various required outputs from the processes that create it through an inventory system. It is centered on a task-based system that is easy to use and configure, allowing efficient development and utilization of common workflows, storing the relevant data and finally ensuring that project and product information is easily accessible.

The code has been released as open source to ensure that the community has access and can collaborate, ensuring the continued development of Leaf LIMS and the ability to adapt to new situations and technologies as they evolve.

METHODS

Leaf LIMS is composed of two components that communicate to provide a working system: a back-end API (Application Programming Interface) written in Python and a front-end written in HTML utilizing the AngularJS framework. These components can be deployed separately or together under one package using Docker (Figure 5).

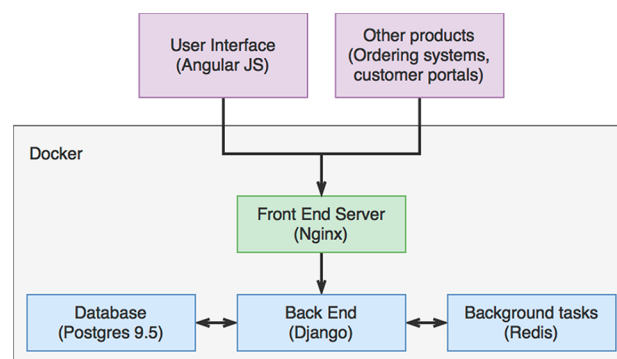


Figure 5. How Leaf LIMS is packaged for use. The system is usually deployed on Docker (gray box) but functions the same if each component is installed manually. The backend can be run separately if a user interface is not required or a custom user interface (UI) is required. All requests are passed to the Nginx server which either displays the UI or passes the request directly to the API on the backend. Requests to the API query the database for information or create a background task if the request requires significant run time. When distributed through Docker, two packages are created: front end (green box) and back end (blue boxes) but are usually installed through the docker-compose tool which installs these along with all the other requirements (the database and background task handling).

The back end provides a REST (Representational state transfer) API written in Python 3 (specifically 3.5) and utilizes the Django and Django REST Frameworks to create an easily extensible and highly capable code base. Data is stored in a Postgres 9 database and takes advantage of some of the newer features regarding the storing and querying of JSON data. The system design is modular in nature which makes it simple to add new features and thus make regular improvements to the versatility of Leaf LIMS. Various open source technologies are used through the system to prevent reinvention such as Redis for running background tasks. The API is designed to be able to communicate with third party solutions therefore this provides token based authentication, meaning these third parties are not required to store the username or password.

The front end of Leaf LIMS was designed to be system agnostic allowing it to be run on any system that offers a modern, standards-compliant web browser. To this end it was written in HTML and utilizes the AngularJS and Angular Material libraries to support complex interactions available in native applications on Windows or Mac OS. Taking advantage of HTML5 technology it can function on a PC or mobile with ease, adjusting itself to suit screen restrictions (Figure 6).

Any LIMS is complex and requires constant development to ensure it maintains parity with advances in synthetic biology. To ensure that during this development process the system

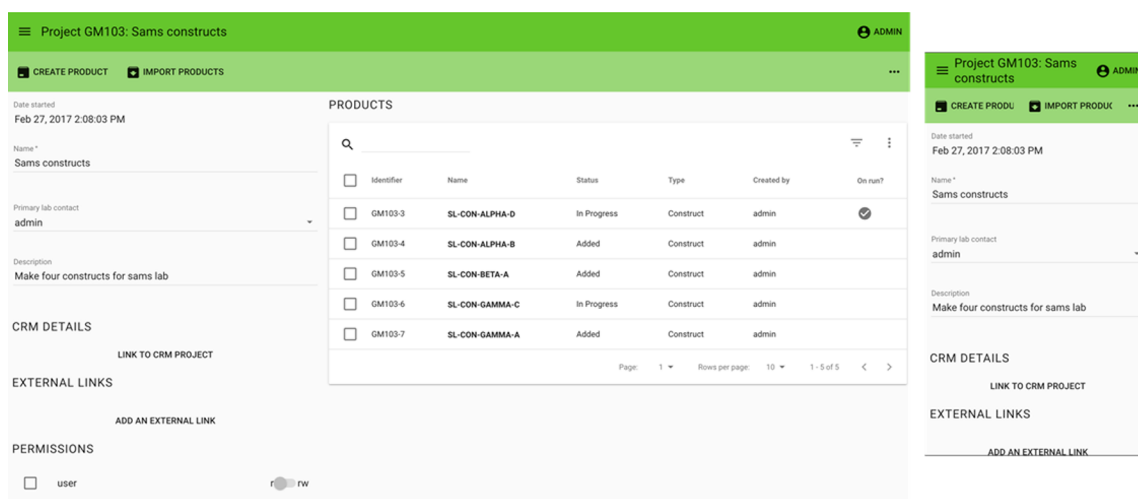


Figure 6. Project details page at different screen resolutions. Leaf LIMS adapts the page to suit the screen resolution, repositioning elements so they can still be used. There is no degradation in functionality at mobile sized resolutions and all tasks can still be performed.

maintains its correct functionality, Leaf LIMS provides a comprehensive test suite. Using this, any changes can be audited, allowing any changes in operation to be identified and fixed before deployment.

Leaf LIMS is open source and available under a permissive license. The code, including documentation, is available on GitHub under the Leaf LIMS project (<https://leaflims.github.io>). All contributions toward the project are welcome, including bug fixes and new features.

Leaf LIMS is designed to be deployed easily to any system that supports Docker. Docker provides a lightweight virtualized environment which reduces issues with conflicting dependencies and allows the various components to be deployed with minimal technical knowledge.²¹ The Docker images are available for use on Docker hub and can be integrated into other deployments. Using the docker-compose tool²² it is possible to indicate the required prerequisites (e.g., Postgres 9.5) and have Docker automatically fetch the required versions and run them in the configuration required. Any updates (including adding in more prerequisites) simply require stopping the existing instance and rerunning docker-compose to upgrade to the newest version. Configuration is kept minimal with only minor editing of files required. The back-end Docker image uses Daphne as a server to provide the API while the front-end uses the Nginx server to serve the HTML content and reverse proxy API requests to the back-end.

Leaf LIMS may be used to communicate sensitive data (e.g., client confidential) and therefore mandates the use of secure socket technology. When deploying, an SSL certificate is required and all connections are automatically routed to the secure port on the Web server.

AUTHOR INFORMATION

Corresponding Author

*E-mail: thomas.craig@liverpool.ac.uk.

ORCID

Thomas Craig: 0000-0001-7029-310X

Daniel Swan: 0000-0001-8978-8129

Author Contributions

Thomas Craig wrote most of the system presented as well as writing and collating revisions to the manuscript. Richard

Holland wrote the test code, alert code and contributed heavily to code clean up. James Johnson and Rosalinda D'Amore came up with the initial brief for the system as well as contributing to editing the manuscript. Hannah V. McCue, Anthony West, Valentin Zulkower and Hille Tekotte provided testing, contributions of ideas and manuscript editing. Daniel Swan and Robert P. Davey managed the project at Earlham with Robert managing Richard Holland during his time on the project. Robert P. Davey, Yizhi Cai, Christiane Hertz-Fowler, Anthony Hall and Mark Caddick were PIs on the project and provided support, financing and helped to write and edit the manuscript.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We would like to thank the BBSRC for funding this work (BB/M00094X/1; BB/M025659/1; BB/M025640/1). We would also like to thank our colleagues within GeneMill, the Earlham Institute and the Edinburgh Genome Foundry for providing valuable insight and support, and Richard Holland for developing the test suite, alert framework and code clean up.

REFERENCES

- (1) Engler, C., Youles, M., Gruetzner, R., Ehnert, T. M., Werner, S., Jones, J. D., Patron, N. J., and Marillonnet, S. (2014) A golden gate modular cloning toolbox for plants. *ACS Synth. Biol.* 3, 839–843.
- (2) Lee, M. E., DeLoache, W. C., Cervantes, B., and Dueber, J. E. (2015) A Highly Characterized Yeast Toolkit for Modular, Multipart Assembly. *ACS Synth. Biol.* 4, 975–986.
- (3) Casini, A., Storch, M., Baldwin, G. S., and Ellis, T. (2015) Bricks and blueprints: methods and standards for DNA assembly. *Nat. Rev. Mol. Cell Biol.* 16, 568–576.
- (4) Smanski, M. J., Bhatia, S., Zhao, D., Park, Y., B A Woodruff, L., Giannoukos, G., Ciulla, D., Busby, M., Calderon, J., Nicol, R., Gordon, D. B., Densmore, D., and Voigt, C. A. (2014) Functional optimization of gene clusters by combinatorial design and assembly. *Nat. Biotechnol.* 32, 1241–1249.
- (5) Kanigowska, P., Shen, Y., Zheng, Y., Rosser, S., and Cai, Y. (2016) Smart DNA Fabrication Using Sound Waves: Applying Acoustic Dispensing Technologies to Synthetic Biology. *J. Lab Autom* 21, 49–56.

- (6) Earlham Institute. (2017) MISO (Managing Information for Sequencing Operations), <http://www.earlham.ac.uk/miso>.
- (7) Savitsky, M., Diprose, J. M., Morris, C., Griffiths, S. L., Daniel, E., Lin, B., Daenke, S., Bishop, B., Siebold, C., Wilson, K. S., Blake, R., Stuart, D. I., and Esnouf, R. M. (2011) Recording information on protein complexes in an information management system. *J. Struct. Biol.* 175, 224–229.
- (8) Johnson, J. R., D'Amore, R., Thain, S. C., Craig, T., McCue, H. V., Hertz-Fowler, C., Hall, N., and Hall, A. J. (2016) GeneMill: A 21st century platform for innovation. *Biochem. Soc. Trans.* 44, 681–683.
- (9) GenoLogics. (2017) BaseSpace Clarity LIMS, <https://www.genologics.com/clarity-lims/>.
- (10) Barillari, C., Ottoz, D. S., Fuentes-Serna, J. M., Ramakrishnan, C., Rinn, B., and Rudolf, F. (2016) openBIS ELN-LIMS: an open-source database for academic laboratories. *Bioinformatics* 32, 638–640.
- (11) BikaLabs. (2017) BikaLIMS, <https://www.bikalims.org/>.
- (12) Atlassian. (2017) JIRA Software—Issue and Project Tracking for Software Teams, <https://www.atlassian.com/software/jira>.
- (13) Synthace. (2017) About Antha, Synthace, <http://www.synthace.com/about-antha/>.
- (14) Ham, T. S., Dmytriv, Z., Plahar, H., Chen, J., Hillson, N. J., and Keasling, J. D. (2012) Design, implementation and practice of JBEI-ICE: an open source biological part registry platform and tools. *Nucleic Acids Res.* 40, e141.
- (15) Stracquadanio, G., Yang, K., Boeke, J. D., and Bader, J. S. (2016) BioPartsDB: a synthetic biology workflow web-application for education and research. *Bioinformatics* 32, 3519–3521.
- (16) Salesforce.com. (2017) What is Salesforce?, <https://www.salesforce.com/uk/crm/what-is-salesforce/?d=7013000000NRZQ>.
- (17) Beal, J., Cox, R. S., Grünberg, R., McLaughlin, J., Nguyen, T., Bartley, B., Bissell, M., Choi, K., Clancy, K., Macklin, C., Madsen, C., Misirlı, G., Oberortner, E., Pocock, M., Roehner, N., Samineni, M., Zhang, M., Zhang, Z., Zundel, Z., Gennari, J., Myers, C., Sauro, H., and Wipat, A. (2016) Synthetic Biology Open Language (SBOL) Version 2.1.0. *J. Integr. Bioinf.* 13, 291.
- (18) Benson, D. A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., and Sayers, E. W. (2013) GenBank. *Nucleic Acids Res.* 41, D36–42.
- (19) Titian Software. (2017) Mosaic, <http://www.titian.co.uk/mosaic-products/mosaic/>.
- (20) Olhovskiy, M., Williton, K., Dai, A. Y., Pasculescu, A., Lee, J. P., Goudreault, M., Wells, C. D., Park, J. G., Gingras, A. C., Linding, R., Pawson, T., and Colwill, K. (2011) OpenFreezer: a reagent information management software system. *Nat. Methods* 8, 612–613.
- (21) Merkel, D. (2014) Docker: lightweight Linux containers for consistent development and deployment. *Linux J.* 2014, 2.
- (22) Docker, Inc. (2017) Docker Compose, <https://docs.docker.com/compose/>.