Abstract

Software system design to support scale in mammalian cell line engineering.

David W. McClymont*, Baird McIlwraith*, Althea Green*, Sam Coulson*, Elizabeth Scott^

Cell line engineering (CLE) is the process of gene editing living cell lines for various purposes such as protein production, fluorescent tagging, compound screening tools, reference standards or diagnostics. Manufacturing processes of the line were typically low throughput and we aimed to increase the capacity of the platform to meet market demand. Software is required to match any throughput and equally as important as any lab based manual or automated methods. To support this increased throughput, a Laboratory Information Management System (LIMS) was developed which contained a data model that represents the CLE process and tracks all projects at each stage. A connection was also made to our own internal ordering system to track projects and register engineering designs based on a markup language we developed called Genetic Engineering Markup Language (GEML). Additionally, we developed an ML based method for analysing clones using an automated imaging platform. The combined wet-lab and software approach allowed us to treble the capacity of the platform allowing us to reach the potential of engineering biology.

1. Introduction

Cell lines are defined as cultures of cells, derived from a single source, which can be propagated repeatedly in a laboratory setting, whilst retaining specific phenotypes and characteristics For example, Henrietta Lacks was an African-American woman whose cancer cells are the source of the HeLa cell line, the first continuous cancer cell line and one of the most important cell lines in medical research[1]. Since then, many different cell lines have been made available, originating from a variety of different species and tissues as continuous or immortalised cell cultures. [2]. Cell lines may also be derived from primary tissues, termed primary cell lines, creating cultures with much higher biological relevance but a limited lifespan in culture. Finally, induced pluripotent stem cells (IPSCs) are another example of a cell line widely used in across the research landscape. IPSCs are pluripotent cells which have been reprogrammed from terminally differentiated adult cells, these cells can be cultured indefinitely in the lab whilst also maintaining their ability to differentiate into any cell type [3].Using viruses, such as lentiviruses, or endonuclease proteins, such as Cas9, genomes of cells can be edited to create new, engineered cell lines, for example by knocking in or out of specific genes or inserting biological tags or creating fusion proteins [4].

CRISPR (Clustered Regularly Interspaced Short Palindromic Repeats) and the Cas9 enzyme is one of the most used genetically engineering techniques to produce cell lines. These cell lines play a role in all aspects of cell biology research, from disease modelling to drug screening, as well as being used for production of biologics. A set of parts can be defined which allow the creation of the desired engineered cell line. For Cas9 editing these are as follows; a cell line, a guide RNA (gRNA), DNA sequence to be inserted in the form of a single stranded oligonucleotide or plasmid (if required) and the primers needed to sequence the appropriate regions for verification.

1.1. CLE KI/KO Nomenclature

Cell line engineering (CLE) can be used to create knock out (KO) cell lines (Figure 5), whereby the expression of a specific target gene is knocked out. A sgRNA designed to bind to a specific DNA sequence in the target gene together with Cas9 protein can be electroporated, transfected or transduced in order to enter the cell [6]. The Cas9 endonuclease binds to the desired gene causing a double stranded break in the DNA. The break is repaired by the internal cellular machinery in a process known as non-homologous end joining (NHEJ). NHEJ, however, is error prone, and can result in the introduction of insertions or deletions (indels) into the gene sequence. These insertions or deletions can introduce frameshifts, altering the DNA sequence of the gene preventing protein translation.

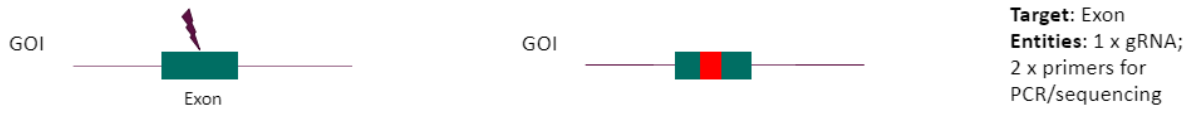

Figure 1: Frameshift Knock out showing the guide being targeted to a Gene Of Interest (GOI) to introduce a frameshift mutation (red) which causes the transcription of that DNA sequence into protein to be altered in such a way that the expression of functional protein is eliminated.

Just as Cas9 can be utilised to create KO cell lines, this same technology can be manipulated to allow for integrations of specific DNA sequences into the genome of a cell line, known as a knock in (KI). KI's allow for example to allow for over expression of specific genes, adding protein tags (e.g. fluorescence proteins or purification fusion tags), mutations such as single nucleotide polymorphisms (SNPs) and removing introns [7]. Alongside the Cas9 protein and the sgRNA, a donor DNA sequence included in the targeting reaction, and this makes use of another of the cells internal repair mechanisms, known as homology directed repair (HDR). Donor DNA sequence can be a plasmid, synthesised single stranded oligonucleotides (ssODN) or double stranded DNA (dsDNA). These DNA sequences contain overlapping sequences homologous to the site that the gRNA is targeting, known as homology arms. The homology arms 'trick' the HDR machinery into incorporating the DNA sequence into the genome.

Figure 2: Homology directed repair and insertion of DNA sequence at the location of interest.

Within synthetic biology, where design and engineering principles are paramount, many methodologies have been created to establish language and definitions around engineering microbes but less so for mammalian cell engineering. Concepts such as datasheets for reusable DNA 'parts' with standardised functionality have enabled the standardisation and reuse of parts in various genetic designs [8]. The synthetic biology open language (SBOL) attempts to allow the communication of genetic designs in a graphical manner. SBOL graphically created designs in a formalised manner that can be utilised using a graphical user interface and connected to a Laboratory Information Management System (LIMS) for manufacture [9]. Glyphs

refer to genetic circuit design and can map directly onto a specified sequence. Ontologies such as Cellosaurus have also been developed to determine accession numbers for cell lines[10]. DNA foundries have also utilised the concept of designs and parts to define production of plasmids including verification of the end result [11].

While many workflows can be developed in commercial LIMS packages, for entirely bespoke processes that have high business utility custom software that integrates with those systems can be developed. Software systems that support automation are often more important than the automaton or processes themselves. Traditional lab automation can e.g. easily load plates into an automated microscope and capture those images. However, high throughput microscopes create a bottleneck in data processing and analysing. Image processing tools such as cell profiler and Ilastik can be used by may need customisation and configuration for the cloud[12,13].

In cell line engineering (CLE), clonal populations of engineered cells are required so that cells populations come from the same single cell progenitor to ensure they are genetically identical. Limiting dilution is a technique which uses low cell densities, calculated in such a way that the volume dispensed in a well has a maximum/ of 1 cell per well across a plate [14]. The distribution means that it is mathematically impossible, using this method, to have every well contain exactly one cell, only with one cell as the maximum. Empty wells and wells with cell-counts greater than one have probabilities following the Poisson distribution with the same lambda (Figure 3). Furthermore, across large numbers of plates, imaging methods need to be developed to determine which of those wells have one cell while excluding the remainder.

Cell staining is often used to aid segmentation in microscopy to aid the highlighting of cells as the background of the well is not stained and therefore black. However, fluorescent stains can be genotoxic. Therefore, brightfield techniques are preferred in CLE processes [15]. However, using traditional segmentation methods on brightfield images leads to high noise and variability when reading across a plate, making it difficult to analyse wells with sufficient resolution [16]. Therefore, we developed ML image analysis models using Ilastik to improve the signal to noise ratio and segmentation capability. These were then orchestrated into a cloud-based application for processing and interrogation.

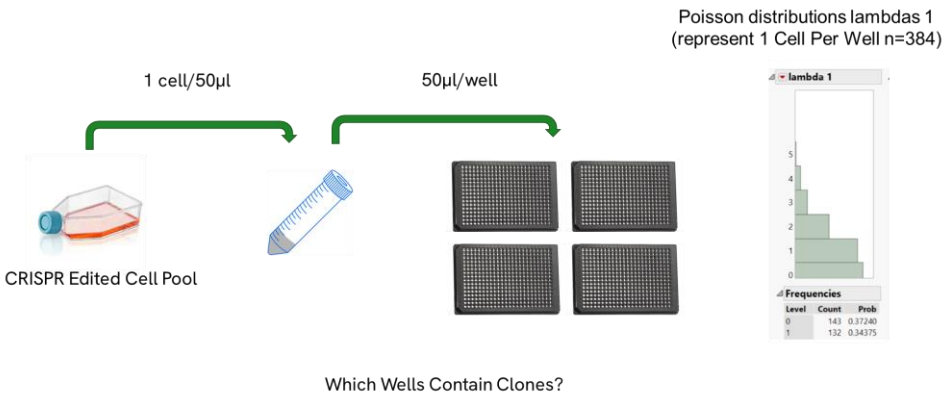

Figure 3: Single cell dilution. Limiting dilution follows the Poisson distribution aiming to have a lambda of 1, therefore having the highest probability of having 1 cell per well.

2. Methods

A set of LIMS were documented that contained CRISPR design tools, Electronic Lab Notebook (ELN), inventory, sample management and dashboard capability. Request for proposal (RFP's) were sent to vendors which resulted in Benchling (Benchling Inc.) chosen as the system that would be used to design and build the LIMS workflows to support CLE.

Data models for cell line, reagents and the CLE process were created and mapped into 'entities' within the Benchling system. ELN pages were created that tracked each stage of the process (transfection, single cell cloning and expansion). Pages were linked together into a process using a request/execution cycle. Dashboards for tracking metrics were written in SQL and ran on the Benchling Insights package. The CLE website was connected to the LIMS and its output converted to GEML using azure function apps and azure service bus messaging service.

To support this work, an automated imaging workcell was built and integrated using Green Button Go software (GBG, Biosero). The system consisted of a robot arm pf-400 (Brook), automated incubator (Liconic), barcode reader (Keyence), plate hotels and a Cell Metric scanning imager (Advanced Instruments). GBG triggers a custom file uploader written in C# to upload images captured to the cloud. Ilastik[13] was used to create the brightfield segmentation model. Images were then processed in the cloud using Ilastik headless mode and Azure Batch (Microsoft Inc.) for orchestration. The data processing pipeline automatically scales virtual machines based on need. Resulting data was stored in Cosmos DB (Microsoft) where a web application was written using asp.net framework for interaction with the data.

All research outlined in this paper is for research use only (RUO) not for use in diagnostic procedures.

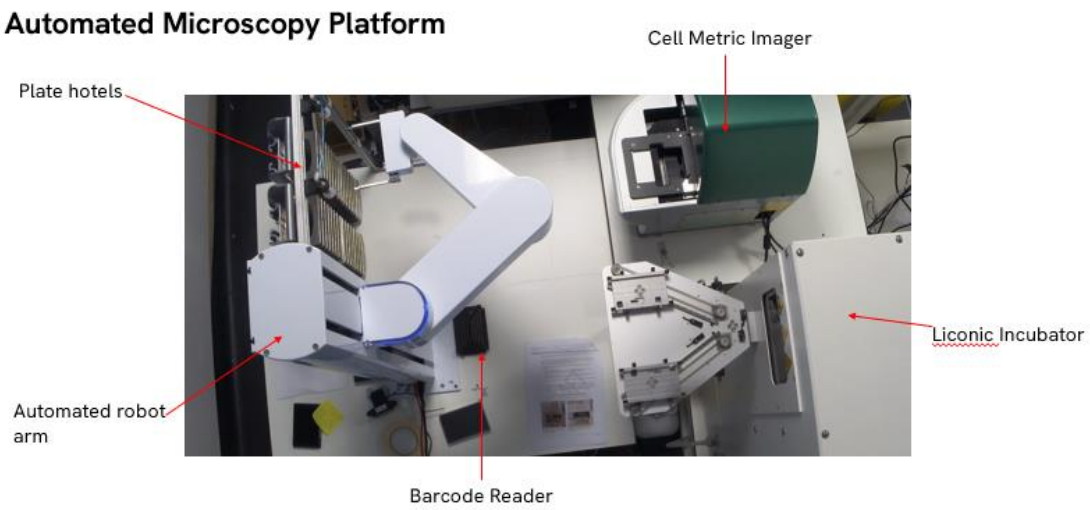

Figure 4: Automated brightfield image capture system. Robotic arm connects the automated incubator to the brightfield microscope for image capture

3. Results

3.1. Data Model and Entities

3.1.1. Cell Line Data Model

Cell line entities were linked in a 4-level hierarchy to ensure consistency in naming conventions and to maintain correct connections to their parental cell lines (Figure 5). The highest level was the canonical cell line object, which contained the cell line name from Cellosaurus and was matched 1-1 with accession numbers from the Cellosaurus database. Name were not free text fields within the system, meaning that all cell lines were given names based on referencing the canonical cell line id, from which the name of all different cell entities were derived. The metadata field 'Aliases' show all the different names that Cellosaurus has brought together under the single accession number.

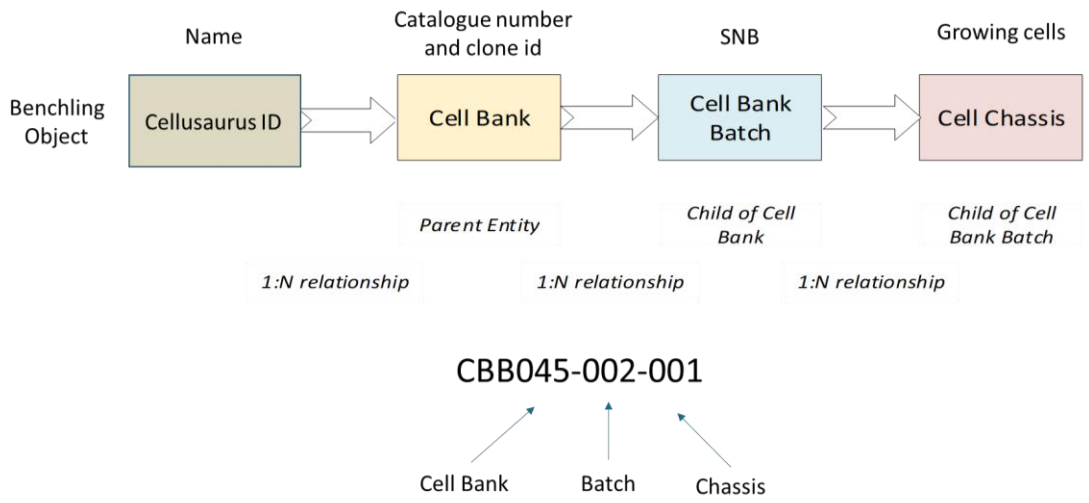

Figure 5: Entity relationship diagram for Cell Lines. Cell Line Data Model showing the 4-level hierarchy of the cell model

The second level was 'Cell bank', which represents a particular source or clone of a given cell line. The entity has a parent-child relationship with an instance of a Cellosaurus ID and obtains its name from the name field of the parent object. The third level, 'Cell Bank Batch', represents the batch and lot number of a given frozen vial. It contains metadata related to physical processing, such as date banked and passage number. The fourth level, 'Cell Chassis', represents a batch of live cells in culture. A batch of cells in culture can either be used for cell line engineering or propagated again for further use, with an associated increase in passage number. Each 'Cell Chassis' entity created at this level represents a culture of cells used to make an engineered cell line, and this can increment as required.

The structured identification code therefore follows the pattern of Cell bank ID, Cell Bank Batch ID, and finally the Cell Chassis ID of the cells. For example, CBB045-002-001, represents the first chassis and second batch of cell bank 45. The combination therefore represents all information required to pinpoint the exact provenance of the cells used in a cell line engineering project.

3.1.2. Reagents Data Model

Reagents used in CLE projects follow a simple hierarchy and were given a parent and child batch level connection. The parent entity represents the immutable characteristics of the object such a DNA sequence or species, whereas the batch level represents the specific instance of that object such as source or concentration. Entities that follow this pattern are guide RNA (gRNA), oligonucleotide donor sequences (single stranded and double stranded) and plasmids. The structured naming system for these entities followed the typical naming convention, using the ID of the parent object with the number of the batch appended to it e.g. gRNA001-05.

3.1.3. Design Data Model

To represent the combination of cell lines and reagents within the LIMS, a grouping entity called Design ID' brings those entities together into a single meta-entity (Figure 6). A programme ID (PRG ID) refers to a cell line deliverable for a given customer and contains connections to IDs in other systems, such as ecommerce.

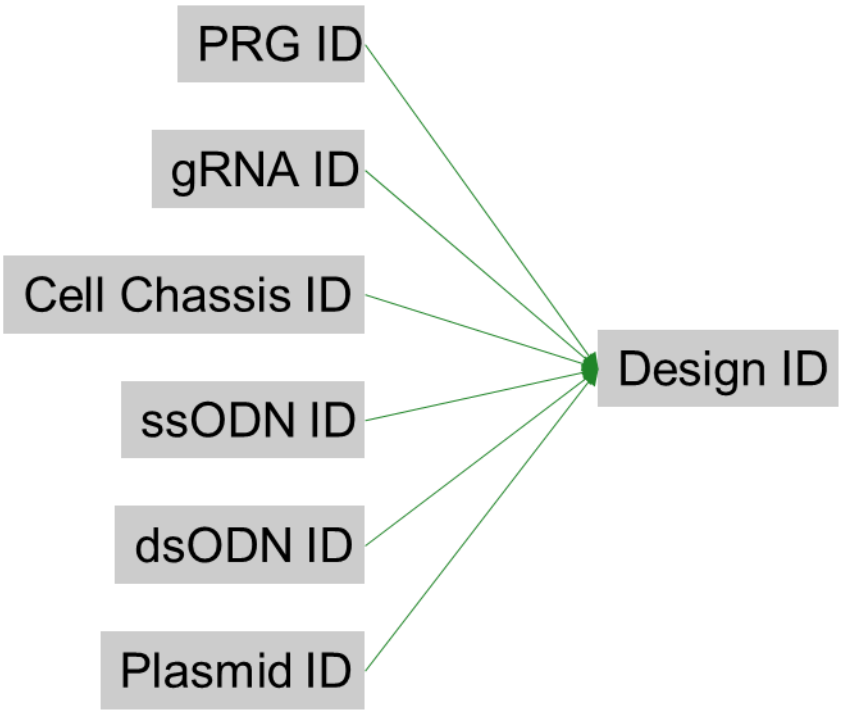

Figure 6: Entity relationship diagram for design data model. The design ID brings together specific instances of all the entities required to build a cell line engineering project.

Specific forward and reverse primers are designed and created for each CLE project and used to sequence the area of the genome has been targeted during the CRIPSR/Cas9 editing event. Individual primers follow the parent-child batch method described for reagents. A primer pair meta-entity brings the forward and reverse primer into one grouping. The primer pair id is then linked to the gRNA ID, representing the specific area that will be amplified.

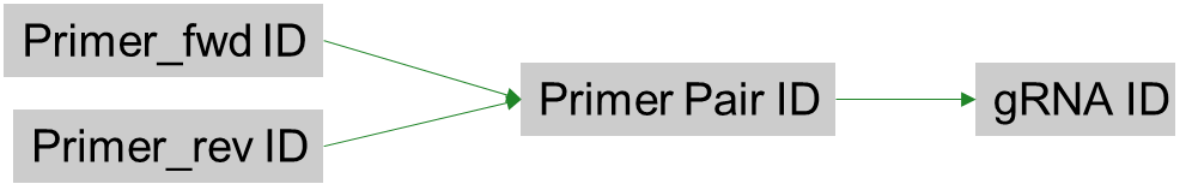

Figure 7: Entity relationship diagram for primers and primer pairs data model. A forward and reverse primer is brought together as a primer pair id that is linked to the gRNA for which the region is amplified.

3.2. GEML

Having a robust data model that encapsulates the complete set of entities required for a CLE project would require manually creating entities correctly, with the correct connections in place to ensure the cell line can be made to specification. Registering a typical project would need creation of 60 entities on average, with 139 entity links. Manual entry of this information directly would a single user >1 hour.

A method was therefore developed to allow end users to configure their gene edited cell line on the website themselves and automate the process of order entry and entity registration into the LIMS. A structured markup language was created to ensure robust immutability in the messages being passed between systems and to ensure that projects were registered correctly to produce the desired result.

An XML format called Genetic Engineering Markup Language (GEML) was created that combines the sequences of the gRNA, primers and Cell line with the correct hierarchical connections to register a design. Therefore, GEML is a canonical, immutable, unambiguous design specification for how to build a Cell line from a series of parts. The XML is enclosed in GEML tags which encloses all subsequent tags used for the project. The <Order> tag encloses information such as customer name and sales order ID. The <engineering> tag takes all the elements related to the manufacturing process such as engineering type (KO or KI), designs containing sgRNA sequences and their related primer sequences, as well as metadata associated with those sequences. The final <manufacturing> tag section outlines the cell line information and various deliverables such as number of clones to ship.

An order was created on the website using the configurator to define the requirements (Figure 8). The customer enters species, cell line and the gene target, as well as their details for delivery. Doing so creates an Azure Service Bus message that records those details which are then processed into a sales order. The sales order is then processed by an Azure function app and converted into GEML for that specification (Figure 9). A service bus message containing the GEML is picked up by a subsequent function app that registers all the required entities in LIMS with the appropriate connections in place.

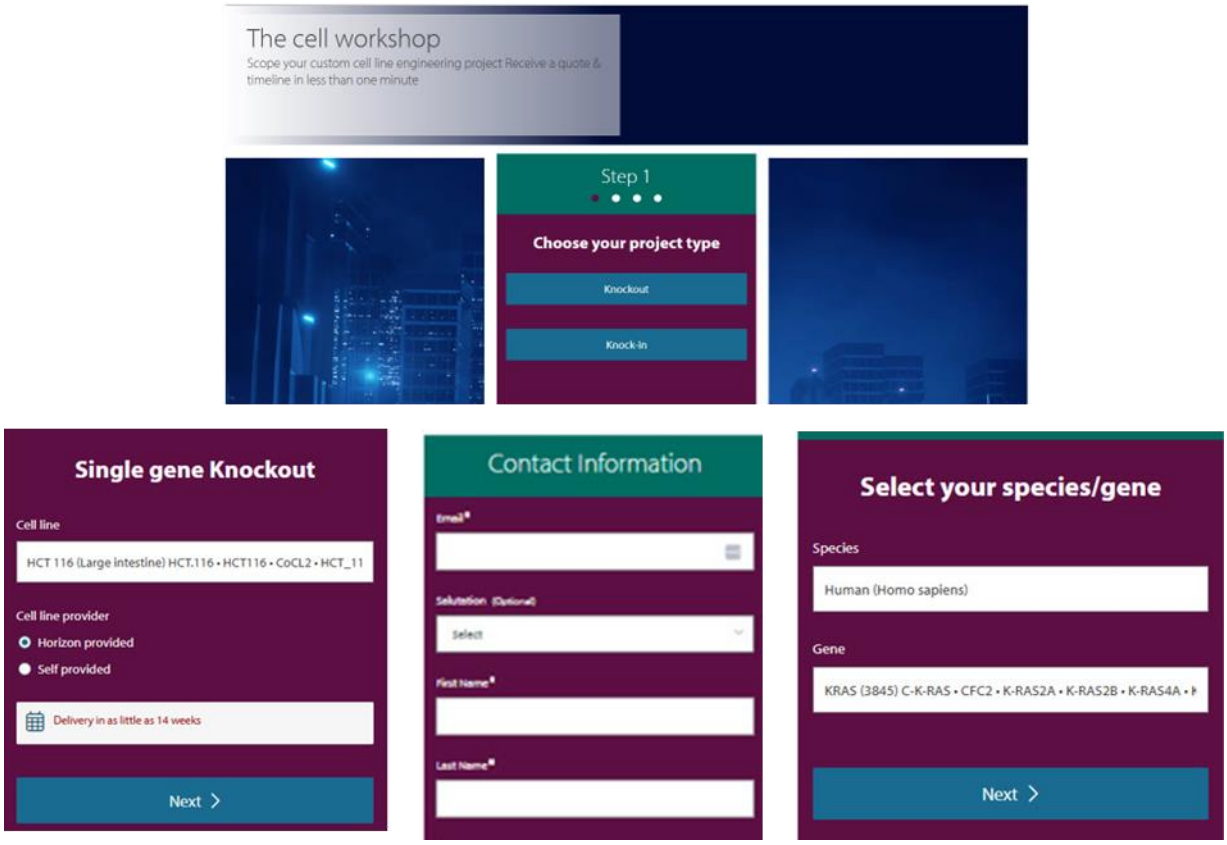

Figure 8: Screenshots from CLE Configurator tool. Customers entered data at each stage detailing the specifications for their desired cell line.

Figure 9: Process flow from creation on website to registration in LIMS. Arrows represent service bus messages

3.3. LIMS Development

With all the correct entities registered, a system to manage and track manufacturing was also developed. CLE splits into multiple phases, that then at each stage mutate the entities into new related entities depending on the processing at each stage. The LIMS has a feature called Requests, where entities are placed in a request which is executed by processing though an electronic lab notebook page. Starting with the design entities, a request is made for those designs to undergo electroporation. The ELN page tracks the processing of those entities. After electroporation, linked entities called 'Pools' are created, which have a parent entity of the design itself. These pools represent a mixed population of cells with a variety of editing events. The pool entities that are considered successful via sequence verification, i.e. identification of the correct target sequence within the pool, are included in a request for single cell cloning. Single cell seeding of selected pool entities are executed through an ELN page, which again tracks that process. At the end of the single cell seeding process, another set of linked entities called 'Clones' are created with a parent relationship to the pool they were derived from. The Clones are again verified by sequencing and those identified as having the correct target sequence, are included in a request for expansion. The last expansion phase ELN consists of growing cell cultures in larger vessels to produce cell numbers sufficient for shipping.

3.3.1. Request Execution Cycle

The request execution process accommodates QC verification, rework loops and dead ends at any stage of the process and enables high levels of flexibility to run many possible pathways simultaneously. Projects have multiple designs, producing many child pool entities. Based on data collected, only successful pools move to the next stage and unsuccessful pools are discarded. The same occurs for clones derived from those pools, where many clones are created, but only successful clones are expanded to be shipped as product.

New designs can be created and executed if no designs are successful. For example, new designs could have the same gRNA with a different cell chassis, or completely new gRNA designs themselves. Multiple pools can be sent to the single cell cloning process or single cell cloning could be performed multiple times if material remains. Pools can also be used as cell chassis' themselves on new designs, either with the same gRNA, to increase editing or with a different gRNA for editing at a different gene loci. A clone can also be used as a cell chassis for a subsequent engineering project, accommodating multiple gene edits and tracking all stages of the process *n* levels deep. All lineages and rework routes are tracked within the same program due to the hierarchical natures of the connections between the entities and their execution through the requests system.

Level	**Type**	**Execution Stage**	**Parent**
1	Program	Initiation	
2	Design	Electroporation	1
3	Pool	Single Cell Cloning	2
4	Clone	Cell Expansion	3

Figure 10: Hierarchy and linkage for entities at each stage of the process

3.3.1. Dashboards

To allow scientists and managers to have an easy overview of the status of projects, we developed a series of Dashboards using the Benchling Insights subsystem. Users can also use pre-defined filters to drill down into the data through the systems in-built search functionality. Dashboards are backed by the SQL editor, which we used to pull all the relevant data pertaining to a project from across the LIMS. Results tables, entities and ELN page statuses were queried, bringing all the different strands of a project together into one concise view. The dashboards also provided direct clickable links to the relevant entity or ELN page, allowing users to easily jump directly to the details of the project if required.

Dashboards were created for each execution stage of the process (Initiation, Electroporation, Single Cell Cloning, Cell Expansion) to allow scientists and managers to manage the platform (Figure 11). For each stage, there are blocks that display the list of active elements to allow both access and status management. Each row in the block links to other pertinent information like due dates and project name. This information was brought together using SQL, generating an- at a glance view of the active projects. The dashboard blocks include one for Incoming Requests, showing a list of those requests waiting for action, effectively a to-do list for the scientists and managers. Another block shows 'Request Execution Info' and details in progress ELN pages with a progress status. The next block shows 'Outgoing Requests', a list of requests from the perspective of the current stage to the next. The final block 'Re-Executions' shows requests that are considered repeats or rework loops.

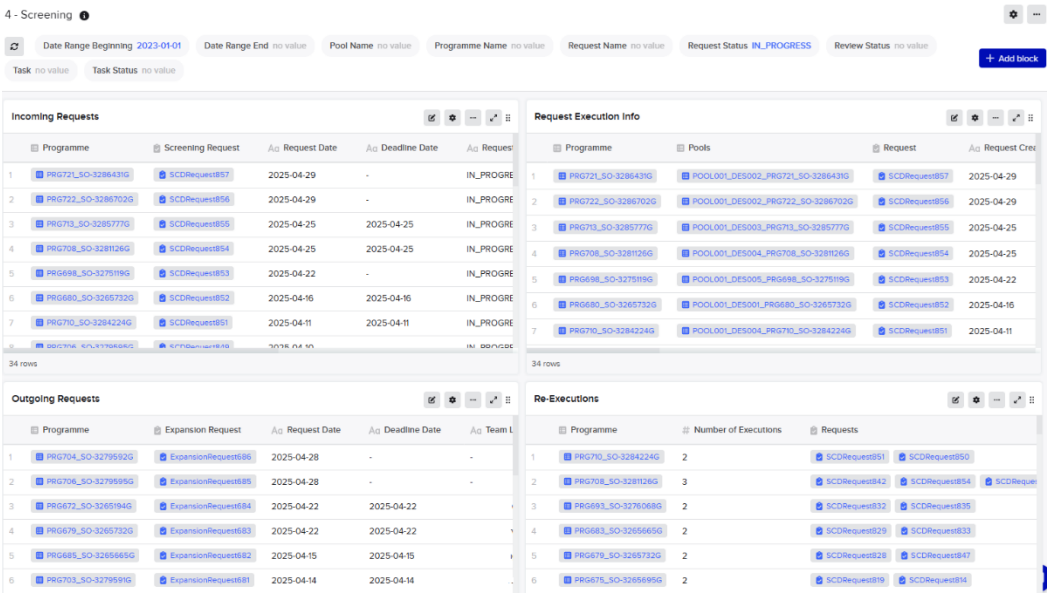

Figure 11: Examples of Dashboards created for the Single Cell Cloning stage of the process.

Further project overviews were created showing platform level metrics for all active projects in the Project Overview dashboard. 'Days Since Last Stage' was a measure used to accurately and reliably track how many days had elapsed since a project had entered its most recent stage in request execution cycle. Any projects that were stuck at a specific stage of the process for an excessive amount of time could be easily identified. To get this information, the number of days that had passed between the current date and the creation date of the relevant request was obtained using SQL. 'Furthest Stage Reached' defined the furthest stage in the request execution cycle that the project had reached in the overall workflow. Due to rework loops, the furthest stage reached was not always the stage the project was currently in. Therefore, latest stage reached was defined which was the latest stage of the workflow defined by time.

The graphing capabilities of Insights Dashboards were also used to provide high level visual metrics. Metrics included project completions per week/month, average repeat rate per project and request execution stage, average project lead time, number of repeats per project type and average time spent in stage. The programme metric dashboards were SQL queries connecting data from across the Benchling ecosystem, such as request dates, entity metadata fields and Results table contents. Repeats were calculated by counting number of requests in each stage of the request execution cycle. The time spent in each stage was returned by calculating the time delta between a Request's creation and completion. Data needed to be binned by month, which was achieved by extracting the day and month of each data point, then grouping and sorting the data by those dates. There were also date filters for the periods of interest and project types.

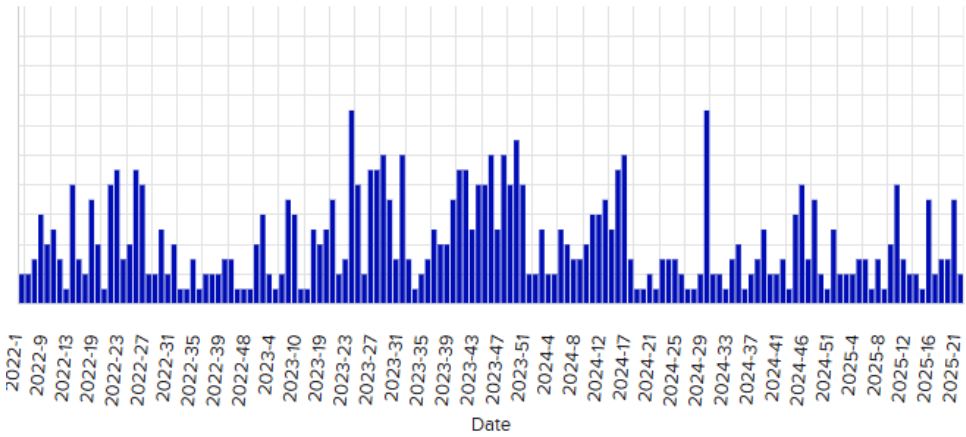

Figure 12: Dashboard displaying an overview of number of processes carried out per week over time

3.3.1. Datapack

Datapacks are PDF and physical documents that are delivered to customers, and contain details about the generated Cell line, such as sequencing information, sgRNA and primer sequences, as well as parental cell line details. The Datapack allows the customer to see the verification steps taken to ensure their edited cell line meets the manufacturing specification, as well as details about other reagents that can be ordered for further experiments.

As the LIMS system contains a data model that tracks the entire lineage of all entities at each stage of the CLE process, information can be gathered by working back from the final clone. The clone is the end node of a tree and, therefore, it is possible to pull all information required for a Datapack from one piece of information. To do this, the Benchling Lookup function is used. In a table, the leftmost column acts as a 'key' that can then be used to link to any related entities in the entire hierarchy. A Clone ID can be used to look up the pool and Design ID entity it was created from. Once the Design ID entity is found, the linked guide sequence and Cell line information can be located.

The lookup tables are placed in an ELN page where multiple lookups are chained together into multiple tables, displaying all the data required in a single page. By chaining tables together, all data required for the Datapack can be brought to one page by entering only one entity at the top. The ELN Datapack page contained logos and is exported as a PDF and sent to the customer. The automated process produced a data pack instantly whereas manually collecting all the data and producing a similar document took around 2 hours.

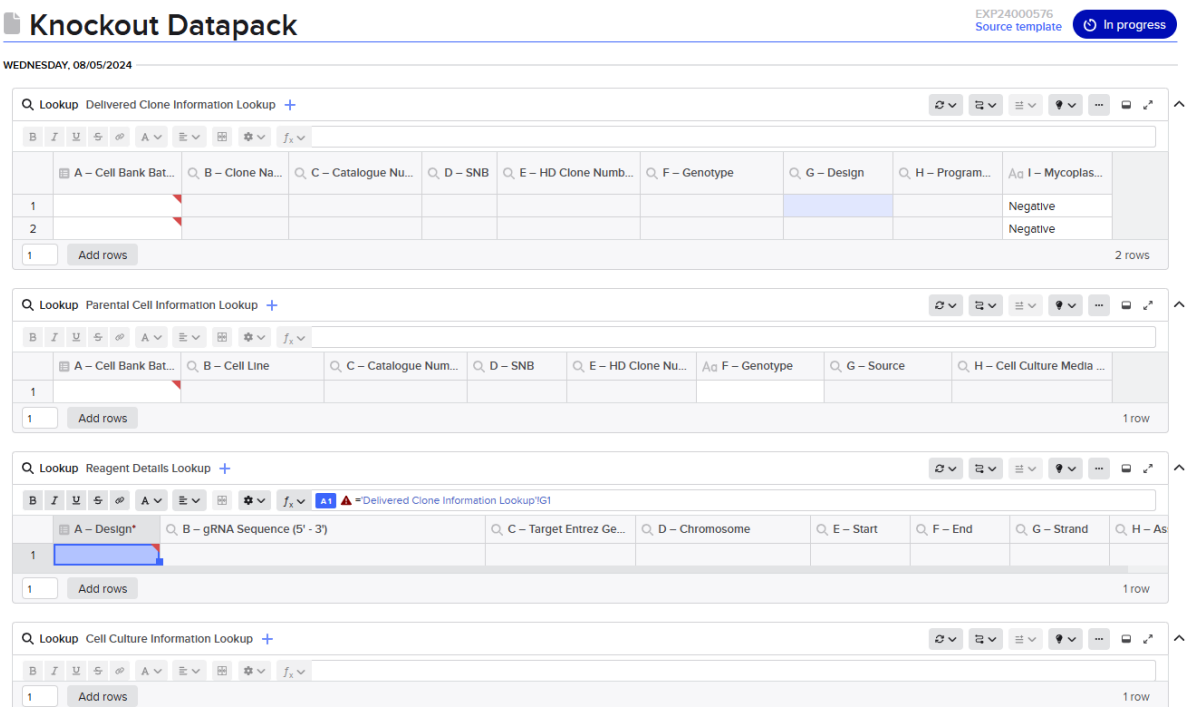

Figure 13: Datapack layout showing lookups within the same table and chained tables.

3.4. Custom Software Development

While the LIMS was developed by configuring a commercial software platform, for processes that are not universal, custom software needs to be built. The greatest bottleneck for the CLE process was the interrogation of brightfield images to produce a list of wells with clonal populations, while excluding empty wells and those derived from multiple cells. A model was created using Ilastik. Ilastik gives live feedback on segmentation, as the expert user teaches the model what is cells and what is not. Scientists annotated training sets of cells against areas of images that contain well aberrations or background and instantly see the result of that segmentation.

3.4.1. Ilastik ML Model

The robustness of the model was tested to ensure its suitability and benefits over existing methods. The main source of noise within segmentation-based imaging methods was false positive edge detection. Within the greyscale brightfield image the change in intensity at the edge of 384-well cell culture plates was classified as cells with other, more basic analysis methods (Figure 14). When tested on an empty 384-well cell culture plate, however, the Ilastik model measured 0 confluence on almost all wells tested.

To test the sensitivity of the system for detecting low cell numbers in these wells, the lower limit of detection was determined. Small colonies with less than 1% confluence could be detected across multiple cell lines (Figure 15). In classified images, areas containing cells were identified, with limited to no detection of areas of the well that did not have cells, even at low confluence. Other possible scenarios were explored to test the lower level of detection and robustness with cell doublets or single cells that land in proximity within a well (Figure 16). Due to the nature of limiting dilution, doublets and single cells that are close to each other could be

misinterpreted as clones, particularly if the read frequency was not high enough. HAP-1, HCT-116 and HEK-293 were tested over time showing the increasing size of colonies were detected successfully (Figure 17).

For each image, the system creates a classified image using the model that detects cells and not well background. From that black and white image contiguous shapes that represent colonies can be detected. The count of discrete closed shapes then represents a colony count, and this is recorded per time point. Colony count is the mechanism used to exclude wells, with wells containing >1 colony excluded from further analysis. Since clones grow out radially from the single cell, with a low enough sensitivity, colony counts at early time points are used to exclude those wells.

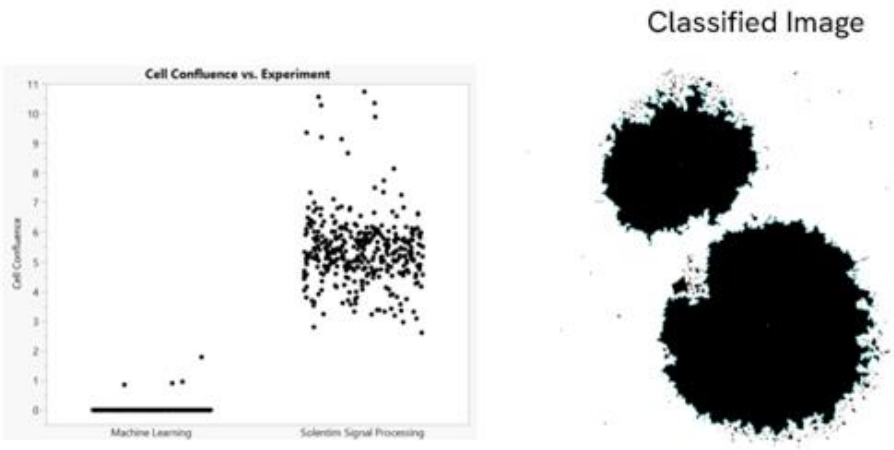

Figure 14: Confluence measurements from empty 384-well plates and example classified imaged used to measure confluence.

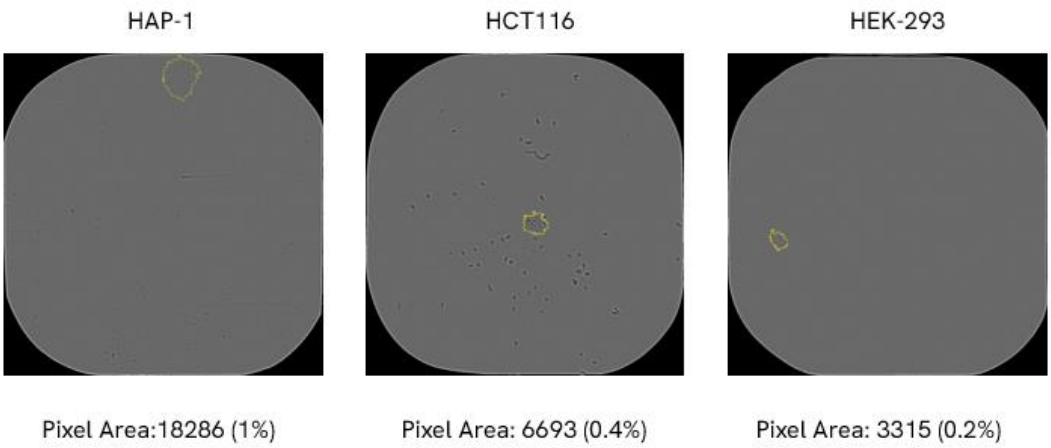

Figure 15: Lower limit of detection across cell lines. Small colonies with less than 1% confluence could be detected across multiple cell lines and non-cell material in wells not detected as colonies.

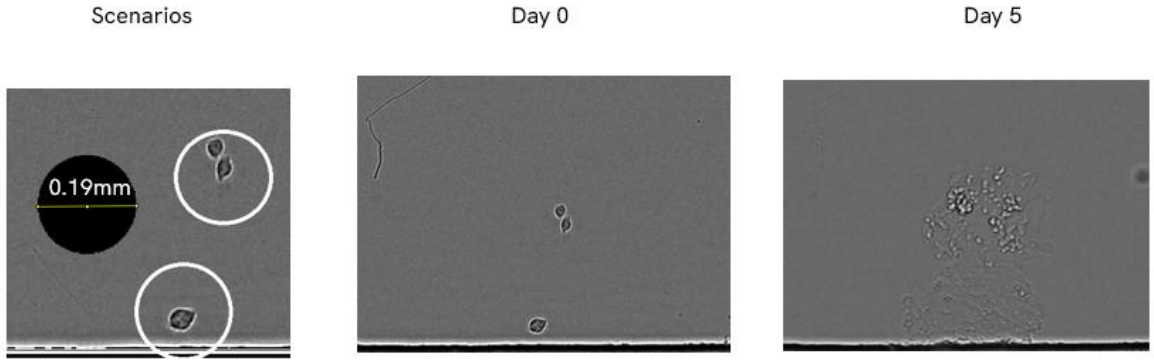

Figure 16: Doublets and cell proximity Images. Due to the nature of limiting dilution, doublets and single cells that are close to each other could be misinterpreted as clones, particularly if the read frequency was not high enough

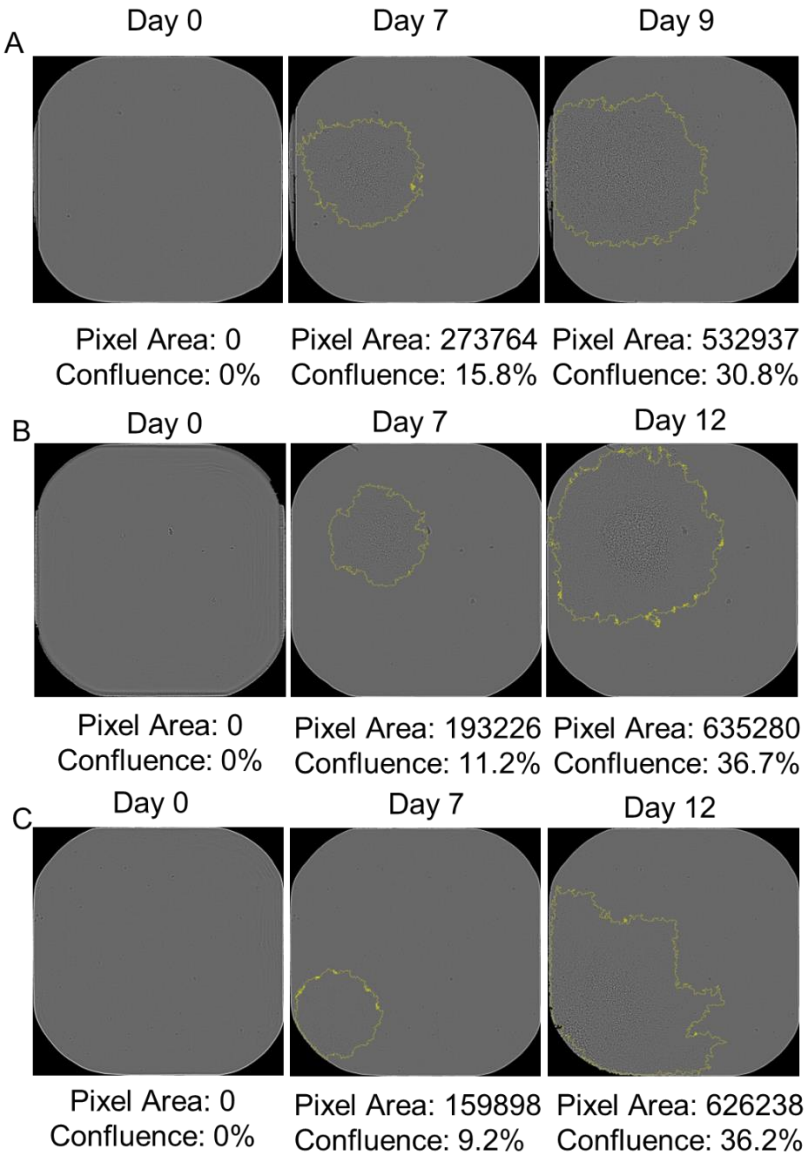

Figure 17:Example images showing detected colonies across time course experiment. A is Hap1, B is HCT116 and C is HEK-293

3.4.1. Cobra Web Application

The Ilastik model pipeline produces the confluence and colony count per well but not a list of clonal wells. Therefore, a methodology was developed to obtain a list of those wells that the system had determined with confidence had only derived from one cell (Clonal wells). Colony counts were tabulated under several scenarios to determine what states changes over time would be considered 'clonal' (Figure 18). By considering colony counts pairwise, shifting by one position across showed that state changes 0-1 and 1-1, with leading 0-0 are allowed. All other state changes result in exclusion from the clonal set. Therefore, by using a simple rule set, all other scenarios representing undesirable outcomes are excluded e.g. multiple cells in wells and colonies that merge and are mistakenly counted as one colony.

A software platform was developed that allowed scientists to produce a picklist of clonal wells and the ability to interrogate images and QC data. The main interface view shows the user a list of studies that are currently active in the system. A study is considered a set of cell culture plates (>10) which contain cells that have undergone limiting dilution. A table shows the date the set of plates were last imaged and the difference between the largest and smallest confluence to give an overview of those studies that need to be actioned.

Focusing on an individual study, a list of all wells is shown with the colony counts over time. Samples that only have state 0-1 or 1-1 are listed under a tab called clonal (Figure 19). The sample view can exclude, in bulk, all images captured on a specific day if there were issues processing images or problems with the optics on the instrument. Further tabs also show lists of samples where clonality is undetermined i.e. they may be clonal or may show unallowed pairwise state changes. These are termed deviations, and there are tabs listing samples with one or two pairwise state changes outside the allowable set. The ability to recover samples near expected value allows QC and interrogation to recover samples that may have been false negatives in error.

Individual samples can be interrogated in further detail in a per-sample view. The view brings together all the images captured from a single well over time. The output from data processing at each stage is tabulated colony size/confluence of each colony at each timepoint. Individual colonies can be excluded or included which can make the overall output more accurate if required (Figure 20). The clonal state is automatically calculated, so if any exclusion or inclusion changes the overall state in such a way that its grouping changes, it will be added to the appropriate list.

After QC, a list of sample ids that are considered clonal is used to produce a pick list that can be downloaded in CSV format and used for further processing by an automated liquid handler. The sample IDs are converted into the physical barcode of the 384-well cell culture plate and the well which they are located int. Studies are then marked as complete and cannot be altered further. Across 429 studies 80% of those studies were above the cut off value of 96 (Figure 21).

Colony count per well per day

Day	0	4	6	7	8	11	
A1	0	0	2	2	2	2	Mixed Population
B1	0	0	0	3	2	1	Merged colonies
C1	0	1	1	1	1	1	Clonal Fast Growing
D1	0	0	0	0	1	1	Clonal Slow Growing
E1	1	1	1	1	1	1	Not captured early timepoints.

Figure 18: Truth table outlining the colony counts per day that are considered a clonal population highlighted in yellow.

Clonal | 1 Deviation | 2 Deviations | Excluded

SAMPLE ID	DAY 0 ✓	DAY 2 ✓	DAY 6 ✓	DAY 8 ✓	DAY 14 ✓	DAY 16 ✓	MODIFIED	GROWTH DIFF.
S0HMJ8V9IM33K1	0	0	0	1	1	1		30.41
S0HMJ8V9IM33AE	0	0	0	1	1	1		19.60
S0HMJ8V9IM30Q3	0	0	0	1	1	1		13.37
S0HMJ8V9IM30NR	0	0	0	1	1	1		12.16
S0HMJ8V9IM30NB	0	0	0	0	1	1		8.29
S0HMJ8V9IM33JP	0	0	0	0	1	1		4.85
S0HMJ8V9IM30IQ	0	0	0	0	1	1		3.45
S0HMJ8V9IM3024	0	0	0	0	0	1		2.52
S0HMJ8V9IM31O9	0	0	0	0	1	1		2.34
S0HMJ8V9IM32JU	0	0	0	0	1	1		2.14
S0HMJ8V9IM2TS9	0	0	0	0	1	1		1.76
S0HMJ8V9IM2UBI	0	0	0	0	1	1		1.71
S0HMJ8V9IM32T4	0	0	0	0	0	1		1.40
S0HMJ8V9IM33JF	0	0	0	0	1	1		1.26
S0HMJ8V9IM30PB	0	0	0	0	1	1		1.25

Total: **109** Page Size: 15

1 2 3 4 5

Figure 19: Study view showing list of samples and the colony counts over time. Tab selected is the clonal tab with allowable pairwise states 0-0 0-1 and 1-1 and are sorted by size (growth diff). Tabs for 1 and 2 deviations and excluded can also be selected.

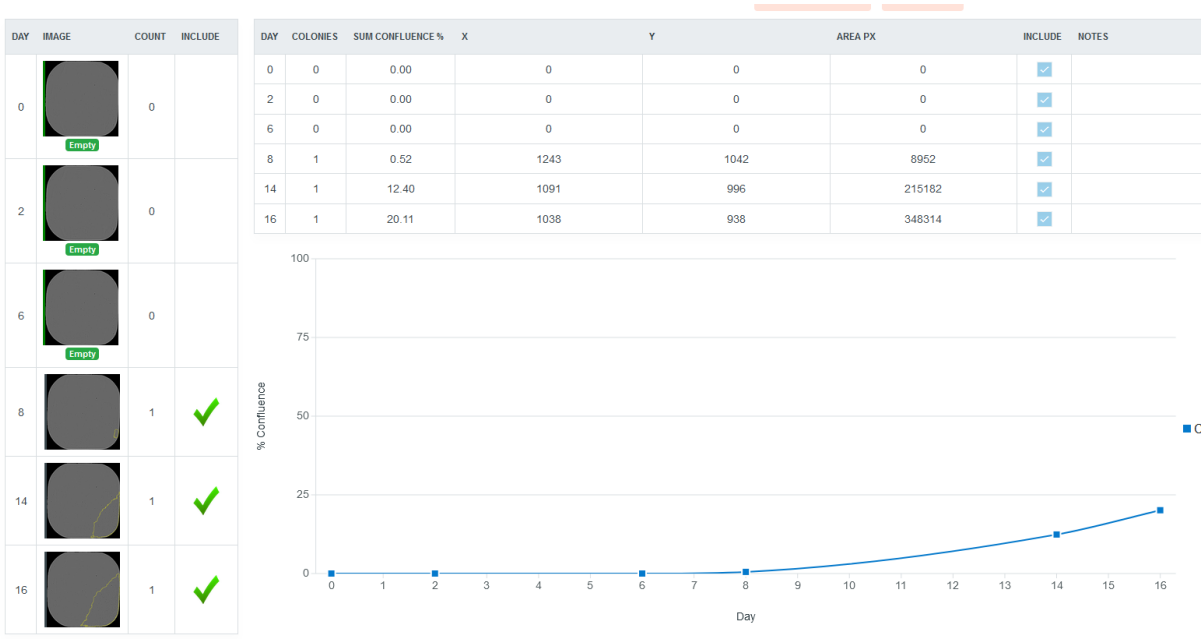

Figure 20: Sample view shows images for each time point from a specific well. Table shows individual colonies which can be included or excluded from analysis. Line connecting points show confluence increasing over time.

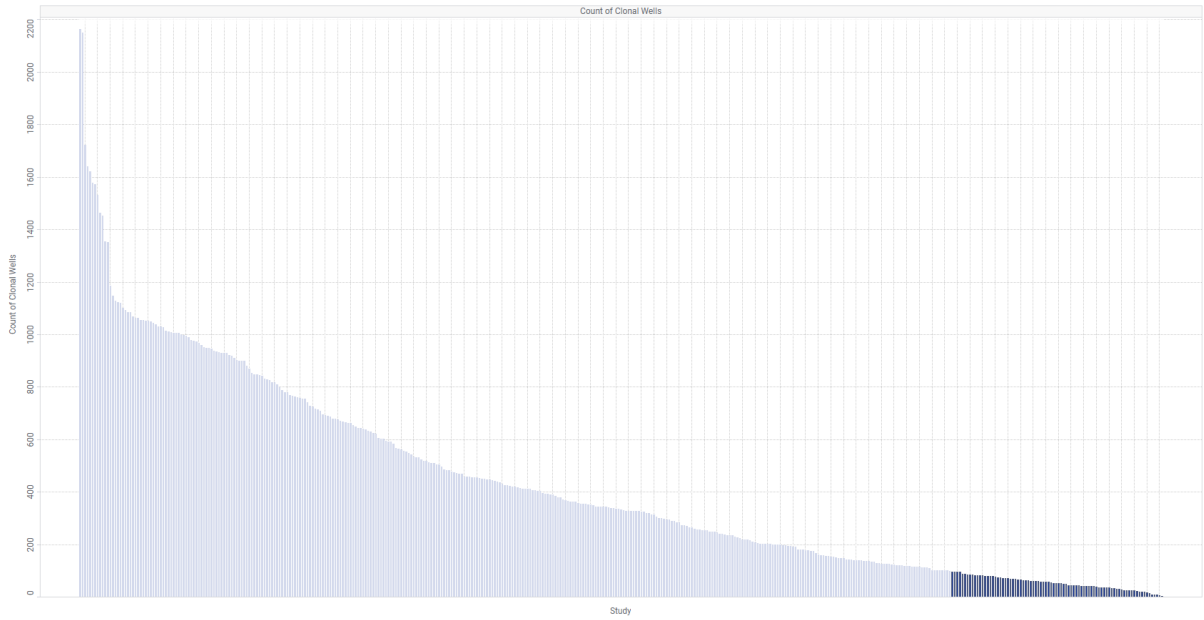

Figure 21: Count of samples considered clonal over complete study. Total in set 429 with those below the cut off of 96 clones highlighted in dark blue.

4. Discussion

Cell line naming conventions have evolved organically over many years with cell lines that were the same often obtaining multiple different names. The commonly used HEK-293 cell line can be called Hek293, HEK;293, HEK/293, HEK-293, HEK 293, 293 HEK, 293, 293 Ad5, Human Embryonic Kidney 293. Many attempts have been made to produce ontologies that limit the freedom associated with naming cell lines such as Cellosaurus, CLO: The cell line ontology, and

the Cancer Cell Encyclopaedia (CCLE) [10,17,18]. Many of these cell lines can be cultured indefinitely in a laboratory setting, and with the development of modern genome engineering techniques such as CRISPR/Cas9, can be engineered to create new cell lines with a variety of genotypes, for example disease models and fluorescently tagged cells [4].

A data model was developed for cell lines allowing this engineering to take place with full tracking of each cell at any stage, whilst generating a standard ontology for identification of cell lines and the engineering outputs at various stages of the process. The starting point for cell lines used for engineering are frozen batches with the lowest possible passage, as long term culture can cause undesired alterations in the genome, metabolism or protein expression [19]. The term chassis is commonly used in synthetic biology as the host of any engineering parts in microbes [20]. We transposed this term to mammalian cells growing in culture and gave an ID number based on passage. With Cellosaurus controlling the Cell line name, the Cell bank and Cell bank batch representing the frozen vials and together with the Cell chassis, all entities were linked in parent-child relationships. A robust model was generated allowing all possible engineering projects carried out by our platform.

The concept of parts coming together into genetic engineering designs was taken from synthetic biology [21]. For editing using CRISPR/Cas9 technology, the target loci within the genome is encoded in the specific gRNA sequence. Therefore, the nucleotide reagents required, both gRNA and DNA (plasmid or ssODN), designed for the specific targeting event required, come together with the Cas9 enzyme as parts. The Design ID was the entity that represented that concept in the data model. It also brought together those parts (also entities in the system) as a meta entity that represented the specific engineering event. The GEML defined a specification for the outcome of the design that any sequencing results could be measured against. The development of the data models and design specification borrowed concepts from synthetic biology. Doing so was required for developing the data models outlined as precise language allowed concepts to be defined in engineering terms leading to entities and concepts that the robust data model could be made from. Without doing so, designing software systems to track scientific processes is difficult.

The data models for cell lines and designs were built in the LIMS system for use in CLE. The request execution cycle of the LIMS system represented the stage gates where child entities are created where the cells undergo a transformation or selection of some kind. At each stage of the process, the requests could be executed again, allowing for tracked re-work loops seamlessly built into the model and created by carrying out the project within the system. The ELN pages used to manage the process at each stage were templatised and were designed around the steps at each stage of the process. However, multiple templates or ad-hoc changes to that page were allowed to build in flexibility to the process e.g. different types of transfection or data analysis. Therefore, flexibility was inherent in the system while still fitting into the rigid stage gates required for tracking.

Projects could not be executed in LIMS without using the data models and request execution system it was built on. By forcing this structure, we created a single source of truth where emergent useful features are possible, resulting in new ways of working and realising increased capacity. As much as possible, data entry through the processes was designed to be passive rather than active, so data within the models was created by virtue of running the process itself. Dashboards could be created bringing together information and links that were relevant to specific stages of the process. These gave tailored views to users for each part of the process providing information such as project ID, status, date created, linking the ELN pages associated

with those projects. Automated data packs were possible as all entities and data was linked together in the model, produced from a single piece of information. Overall platform status metrics were used for tracking performance criteria and to generate kaizen improvement project ideas.

The data models in LIMS were extracted from GEML, the latter created from customer specified engineering event attributes such as cell line and edit type. These attributes were captured via the CLE Configurator Tool, which aimed to replicate the simplicity of the customer facing web-based interfaces used to connect the ecommerce systems and automated labs producing synthetic nucleotides. For synthetic nucleotides, capturing sequences is easy because customers often know exactly what sequence they require, so it can be captured on a form and transferred to a manufacturing system queue[21]. GEML built on this simple system and extended it to the more complex process of cell line engineering. It placed an order in the ecommerce system and was used to generate all entities required for a project, whilst initiating ordering workflows for the reagents.

A key part of each project was validating the clonality of engineered cells. For this an automated imaging platform was built. This automated imaging platform could produce up to 240 images per day. Manual analysis methods were impractical for such large numbers of images. Therefore, it was important to develop a software solution for image analysis that kept pace with the image generation rate of the automated microscope to prevent workflow bottlenecks.

Our approach was to use supervised machine learning to create a model so that the large amounts of data typically required for unsupervised learning was not needed. The Ilastik user interface used the expert knowledge to mark-up areas of the images that contained cells and those that did not, with instant feedback on how changes updated the classifications. A representative set of images was created for inconsistencies in the wells such as dust and marks resulting from the effects of irradiation that would occur over time. The GUI software gave immediate feedback when choosing pixels for each group classified, making building an accurate model easier for the user.

We developed an automated imaging workcell and workflow data analysis pipeline in the cloud, together with a web application to manage the system. The platform produces a list of putative clones identified over the scanning period that are used to pick samples to move to the next stage. We have, to date, autonomously collected and analysed over 16.8 million images. The model cannot identify a single cell in brightfield, but measures discrete colony counts down to 1% confluence. If images are captured at a high frequency at the early stages, discreet small colonies can be resolved and excluded if more than one is detected. Rather than producing a 'black box' model, our software designed to automate as much as possible any decision making, while retaining the ability to interrogate, interact with and QC samples to improve the quality of the desired outputs.

Here we show how the interaction of configured software systems and custom programs that can represent physical process and data analysis are equally as important tools as robotic automation and are required to develop modern biotechnology platforms. LIMS systems and robust data models facilitating a single source of truth allow for many useful paradigms or metrics to be created as questions arise. The platforms described seamlessly manage multiple hundreds of projects per year and up to 30 users simultaneously. When embarking on any large-scale automation project, building sufficient software infrastructure needs to considered. Without such systems, increased scale cannot be realised.

1. Masters, J. R. HeLa cells 50 years on: the good, the bad and the ugly. *Nat. Rev. Cancer* **2**, 315–319 (2002).
2. Stacey, G. & MacDonald, C. Immortalisation of Primary Cells. in *Cell Culture Methods for In Vitro Toxicology* (eds. Stacey, G. N., Doyle, A. & Ferro, M.) 27–42 (Springer Netherlands, Dordrecht, 2001). doi:10.1007/978-94-017-0996-5_3.
3. Passier, R., Orlova, V. & Mummery, C. Complex Tissue and Disease Modeling using hiPSCs. *Cell Stem Cell* **18**, 309–321 (2016).
4. Bukhari, H. & Müller, T. Endogenous Fluorescence Tagging by CRISPR. *Trends Cell Biol.* **29**, 912–928 (2019).
5. Doudna, J. A. & Charpentier, E. The new frontier of genome engineering with CRISPR-Cas9. *Science* **346**, 1258096 (2014).
6. Kim, T. K. & Eberwine, J. H. Mammalian cell transfection: the present and the future. *Anal. Bioanal. Chem.* **397**, 3173–3178 (2010).
7. *Selecting for CRISPR-Edited Knock-In Cells*.
8. Exley, K. *et al.* Utilising datasheets for the informed automated design and build of a synthetic metabolic pathway. *J. Biol. Eng.* **13**, 8 (2019).
9. Galdzicki, M. *et al.* The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nat. Biotechnol.* **32**, 545–550 (2014).
10. Bairoch, A. The Cellosaurus, a Cell-Line Knowledge Resource. *J. Biomol. Tech. JBT* **29**, 25–38 (2018).
11. Rajakumar, P. D. *et al.* Rapid Prototyping Platform for Saccharomyces cerevisiae Using Computer-Aided Genetic Design Enabled by Parallel Software and Workcell Platform Development. *SLAS TECHNOLOGY: Translating Life Sciences Innovation* vol. 24 291–297 (2019).

12. Carpenter, A. E. *et al.* CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.* **7**, R100 (2006).
13. Berg, S. *et al.* ilastik: interactive machine learning for (bio)image analysis. *Nat. Methods* **16**, 1226–1232 (2019).
14. Staszewski, R. Cloning by limiting dilution: an improved estimate that an interesting culture is monoclonal. *Yale J. Biol. Med.* **57**, 865–868 (1984).
15. Ge, J. *et al.* Standard fluorescent imaging of live cells is highly genotoxic. *Cytometry A* **83A**, 552–560 (2013).
16. Tse, S. *et al.* A combined watershed and level set method for segmentation of brightfield cell images. in *Medical Imaging 2009: Image Processing* vol. 7259 1147–1156 (SPIE, 2009).
17. Sarntivijai, S. *et al.* CLO: The cell line ontology. *J. Biomed. Semant.* **5**, 37 (2014).
18. Ghandi, M. *et al.* Next-generation characterization of the Cancer Cell Line Encyclopedia. *Nature* **569**, 503–508 (2019).
19. Prasad, C. P., Tripathi, S. C., Kumar, M. & Mohapatra, P. Passage number of cancer cell lines: Importance, intricacies, and way-forward. *Biotechnol. Bioeng.* **120**, 2049–2055 (2023).
20. Adams, B. L. The Next Generation of Synthetic Biology Chassis: Moving Synthetic Biology from the Laboratory to the Field. *ACS Synth. Biol.* **5**, 1328–1330 (2016).
21. de Murieta, I. S., Bultelle, M. & Kitney, R. I. Data model for biopart datasheets. *Eng. Biol.* **2**, 7–18 (2018).
22. Armer, C., Letronne, F. & DeBenedictis, E. Support academic access to automated cloud labs to improve reproducibility. *PLOS Biol.* **21**, e3001919 (2023).
23. Hoose, A., Vellacott, R., Storch, M., Freemont, P. S. & Ryadnov, M. G. DNA synthesis technologies to close the gene writing gap. *Nat. Rev. Chem.* **7**, 144–161 (2023).
24. Wildey, M. J., Haunso, A., Tudor, M., Webb, M. & Connick, J. H. Chapter Five - High-Throughput Screening. in *Annual Reports in Medicinal Chemistry* (ed. Goodnow, R. A.) vol. 50 149–195 (Academic Press, 2017).