Designing Genomes using Design-Simulate-Test Cycles

Joshua Rees[1,2*^], Oliver Chalkley[1,3,4*^], Oliver Purcell[5], Lucia Marucci[1,3,6+], Claire Grierson[3,5+^]

[1]BrisSynBio,University of Bristol, Bristol BS8 1TQ, UK;
[2]School of Biological Sciences, University of Bristol, Bristol Life Sciences Building, 24 Tyndall Avenue, Bristol, BS8 1TQ, UK;
[3]Department of Engineering Mathematics, University of Bristol, Bristol BS8 1UB, UK;
[4]Bristol Centre for Complexity Science,Department of Engineering Mathematics, University of Bristol, Bristol BS8 1UB, UK;
[5]Prospect Bio, 150 N Hill Drive, Ste 14, Brisbane, CA 94005, USA;
[6]School of Cellular and Molecular Medicine, University of Bristol, Bristol BS8 1UB, UK;

*Co-first authors +Co-last authors ^Corresponding authors
Corresponding authors: Prof. Claire Grierson (claire.grierson@bristol.ac.uk), Dr Lucia Marucci (lucia.marucci@bristol.ac.uk), Oli Chalkey (o.chalkley@bristol.ac.uk), Joshua Rees (joshua.rees@bristol.ac.uk.

Abstract

In the future, entire genomes tailored to specific functions and environments could be designed using computational tools. However, computational tools to design cells are scarce. Here we present work implementing computational design-simulate-test cycles for genome optimisation based on whole cell modelling. Similar approaches could be adapted to any goal in genome design, but to demonstrate feasibility, we selected the identification of minimal genomes as a proof of concept, using the first (and currently only published) whole cell model, which is for the bacterium *Mycoplasma genitalium*. Minimal genomes are an ideal goal to test our ideas because there is a very simple functional assay - the cell can either replicate or not. Our computational design-simulate-test cycles produced novel *in-silico* minimal genomes smaller than *JCVI-Syn3.0*, the smallest genome ever synthesised in the lab, and identified 11 redundant essential genes. This work brings computational genome design a step closer.

Introduction

One of the goals of synthetic biologists is to be able to design genomes to produce cellular products via optimised cell factories [1], which are more robust, energy efficient, and controllable. The by-product of this research is a greater understanding of the essentials for cellular life, which many hypothesise could provide insight into early life on earth [2].

The largest scale efforts in genome editing to date include *JCVI-Syn3.0*, a 50% reduction from the wild-type *Mycoplasma mycoides* [3], several strains of *E.coli* reduced by 38.9% [4] and 35% [5] *in-vivo* and a reduction of 77.6% as a free-genome inside *S.cerevisiae* [6], and two strains of *B.subtilis* reduced by 36% [7]. These were produced by trial and error laboratory approaches rather than intentional design, using very cost and time intensive techniques that are only available to select laboratories. This results in laboratories following a small number of research avenues with limited ability to backtrack [8]. Although the methodology of reducing existing genomes has drawbacks, attempting to build novel organisms from scratch is currently

infeasible, due to technology constraints causing the cost per base pair printed to be incredibly large at the microbial genome scale.

When designing genomes the identification of minimal genomes is understood to be a good proof of concept [8]. Minimal genomes are defined as reduced genomes containing only the genetic material essential for survival, presuming that an appropriately rich medium is present [3] [8] and in the absence of external stresses. An individual cell is usually defined as "living" if it can reproduce successfully, so a gene is *"essential"* if it is indispensable for enabling the cell to survive until successful reproduction and its removal prevents a cell from successfully dividing. A *"non-essential"* gene can be removed with division still possible [8] [9]. This is classically tested by knocking out or suppressing genes individually and observing the impact on the cell.

Genome design is currently made difficult by a lack of biological knowledge about which genes are essential, due to conservation of function and dynamic classifications.
Gene function is conserved over genes, with functionally equivalent genes having independent evolutionary origins resulting in differing genetic code. These are labelled as non-orthologous gene displacements [8] and prevent genomic insights being shared across species.
Gene essentiality is no longer thought of as static and binary, the requirement for specific genes changing dependent on the combined effect of the external cellular environment and the genomic context.[8] The genomic context changes each time a gene is removed or suppressed. Some essential genes become dispensable with the removal of a particular gene (i.e. the former removing a toxic byproduct of the latter) [8].
Likewise some non-essential genes become essential when a functionally equivalent gene is removed, leaving only a single pathway to a required cellular metabolite. These are called redundant essential genes, and the cellular death that occurs when redundant essentials are removed together is referred to as synthetic lethality [3] [17].
This results in a "gradient" of gene essentiality, with some essential genes being essential in fewer contexts than other essential genes, and some non-essential genes being dispensable less frequently than others. A recent review [8] suggested genes could be assigned quantitatively to four categories: no essentiality (dispensable in all contexts), low essentiality (dispensable in some contexts), high essentiality (indispensable in some contexts), and complete essentiality (indispensable in all contexts).

Design is further challenged by the fact that the number of potential genome designs is a factorial. If we consider an organism with 500 genes, the number of possible genomes that we could search through numbers 1.22×10^{1134} (for comparison the age of universe in seconds is $n \times 10^{17}$). This is an amount that we do not currently have the computational power to investigate in full, so how do we investigate this number of solutions, using only the currently available knowledge, and within a reasonable time frame?
By using a single bacterial species with a small genome, keeping the external environment constant while considering the genetic context of the cell, and cross comparing experimental results at scale, we increase our chances of

producing successful genome designs and identifying redundant essential genes; however, as previously noted, this scale of work is not currently possible in the laboratory.

We propose to use a computational methods to attempt to solve this problem. We used the *Mycoplasma genitalium* whole-cell model [10], the currently only existing model that fully considers genomic context, describing the smallest natural culturable self-replicating organism [11]. This formalism models a single *Mycoplasma genitalium* cell from random initial conditions until the cell successfully divides or reaches a time limit of 13.89 hours (simulated time). It combines 28 cellular submodels, accounts for 401 of the 525 *M.genitalium* genes, with parameters based on 900 publications and more than 1,900 experimental observations, resulting in a reported accuracy of 79% [1]. Outside of the single gene knockout simulations conducted in the original paper, it has been used to make novel biological discoveries by investigating discrepancies between the model and real world measurements [10,12], to model genetic circuits in the context of the cell [13], and to make predictions about the use of existing antibiotics against new targets [14].

Here, we used the whole cell model to design *in-silico* minimal genomes: our computational genome design tools (named LEGO and GAMA) found possible reduced *M.genitalium* genomes smaller than the currently smallest genome produced in the laboratory (*JCVI-Syn3.0*). Simulation analysis identified 11 redundant essential genes, one redundant essential pair, and 29 high essentiality genes. We believe that our computational design-simulate-test cycles can be used in the future for designing and engineering genomes for a broad spectrum of uses and are applicable to current subcellular models and future whole-cell models across species and applications.

Results

Genome Design Tools: LEGO & GAMA

We produced two methods (LEGO and GAMA) which both use the *Mycoplasma genitalium* whole-cell model to generate minimal genome designs (Table 1). They do this by conducting cycles of whole-cell model simulations in three steps: Design (select possible gene deletions), Simulate (the genome minus those deletions), Test (analyse the *in-silico* cell produced). Those simulations that produce successfully dividing cells are used to inform the next cycle of simulations. The number of gene deletions is increased in each subsequent simulation cycle (while maintaining cellular division), thereby producing a smaller and smaller genome. LEGO and GAMA differ in their design steps but use the same simulate and test steps.
LEGO and GAMA generated 1000s of genome designs, for brevity the designs presented in this paper are those with the most genes deleted.

LEGO is a genome design tool that uses a divide and conquer algorithm, with an initial design step breaking down a problem into sub-problems that can be solved, and subsequent combination steps combining the sub-problem solutions to solve the original problem. This is inspired by current laboratory methods and can generate genome designs within two days (see Methods). It is a coarse grained method producing deletion segments of groups of genes rather than individual genes (see Table 1). It currently uses only genes defined as

non-essential by single gene knockouts (see Initial Input), to enable manual simulation management and data storage.

GAMA is a genome design tool that uses a modification of a genetic algorithm. It uses genes defined as non-essential by single gene knockouts in the guess and add steps, but incorporates essential genes in the mate step. It is a granular method, producing deletion segments of groups of genes and individual genes, making it capable of producing a greater number of predictions but takes two months to generate genome designs. It also requires multiple supercomputers to handle the number of simulated genome designs, and custom management code to the coordinate multiple supercomputers and store the resulting data.

	LEGO	GAMA
Methodology	Divide and Conquer Algorithm	Genetic Algorithm
Computing Requirements	Supercomputer	Supercomputers
Restrictions	Non-essential genes only*	Management code
Deletion Segment Size (Genes)	100s - 10s	100s - 1s
First Genome Designs	2 days	2 months
Predictions	10s	1000s

TABLE 1 Differences between LEGO and GAMA genome design tools Both tools use the *Mycoplasma genitalium* whole-cell model and computational design-simulate-test cycles to design genomes. They differ in their design approach, with the resulting differences outlined above. Details on algorithm implementation, as well as calculations for first genome designs timings are provided in the methods and supplementary information. *LEGO uses a subset of the genes within the model to enable manual simulation management and data storage.

Initial Input

The *Mycoplasma genitalium* whole-cell model takes a list of genes as input, that tells the model which genes are available for that particular simulation. The output is thousands of data streams about the cell, from which we graph six categories to display cell division, replicated DNA, produced RNA, produced Protein, and growth (increase in mass).

To generate input for our methods, and to provide background data on gene essentiality absent of any other genomic changes, we simulated single gene knockouts in an otherwise unmodified Mycoplasma genitalium *in-silico* genome

(previously reported by Karr et al[10]. Each of the 359 protein-coding genes was simulated individually, with a minimum of six replicates for each gene. We found that, when knocked out individually, 147 genes were non-essential (producing a dividing simulated cell) and 212 genes were essential; their removal resulting in the cell being unable to divide.
Overall, our single gene knockout simulations agree 99% with Karr et.al's [10] findings, with disagreements over the essentiality classifications of 4 genes. It is to be expected that the majority of the *Mycoplasma genitalium* genome would be essential (212/359, 59%) as *Mycoplasmas* have reduced evolutionary redundancy due to their obligate parasitic nature, genetic history, and lack of need for adaptability [15]

LEGO Method and Results

The initial design step of LEGO divides the input into 36 deletion segments (A,B,..,Z,aa,..,jj) ranging in size from 100% of the input (147 genes) to 6.25% of the input (10 genes) (see Supplementary information). The build and test steps are then conducted: in individual simulations each of the 36 deletion segments were removed from the genome, and the reduced genomes simulated. The deletion segments that successfully produced a dividing *in-silico* cell were carried forward to the next step.
The combination step starts with the largest successful deletion segment (e.g. Segment A), which was then matched with all other non-overlapping and successful deletion segments (which reduces the number of segments dramatically) to create the largest deletion design (e.g. Segment A + 5 segments). Following this, smaller deletion designs were generated by comprehensively matching the non-overlapping deletion segments in all possible iterations of smaller combinations (e.g. Segment A + all combinations of 4 smaller segments, Segment A + all combinations of 3 smaller segments, Segment A + all combinations of 2 smaller segments, Segment A + all combinations of 1 smaller segment). This matching process was then repeated with the second largest deletion segment, and subsequent deletion segments until the first 6.25% deletion segment was reached.
All of these designs were built and tested, and the resulting *in-silico* cells analysed to inform the next cycle of simulations (see Figure 1A for an example). This combination step was repeated until no further segments could be removed without preventing division. Once the combination step was exhausted, simulation results were cross compared to identify single gene deletions that were candidates for removal, which were then simulated.

LEGO produced results quickly, the first combination step producing a genome design that deleted 120 genes (a 30% reduction) comparable to current lab based efforts in other species. [4,5,7] The subsequent addition of single gene deletions increased the number of deletions.
In total, the smallest LEGO designed genome managed to delete 140 genes (Figure 1B), leaving a total genome size *in-silico* of 261 genes (named LEGO_261), producing a simulated cell which replicates DNA, produces RNA and protein, grows, and divides.

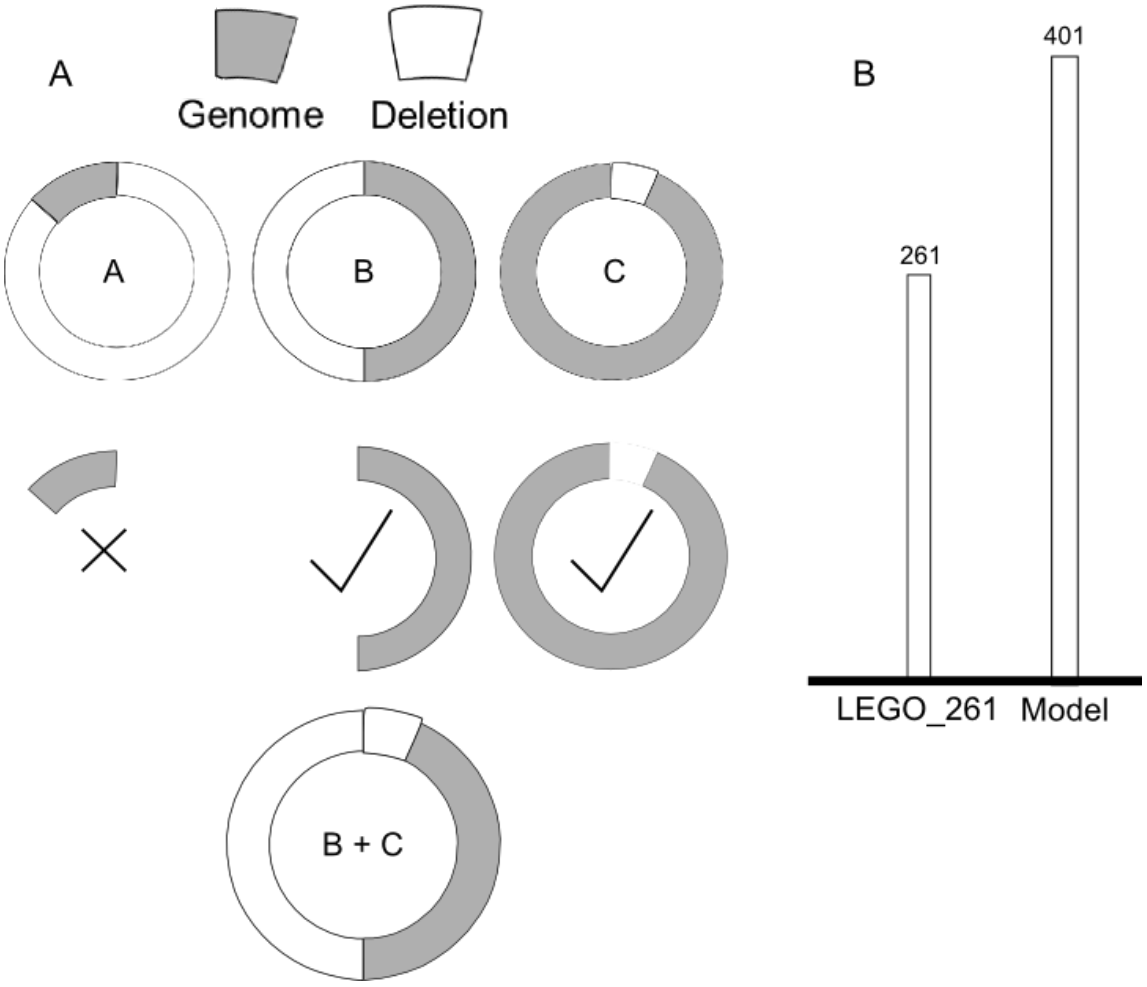

Figure 1 ***LEGO Method for Genome Design: Example and Results*** (A) Using the *Mycoplasma genitalium* whole-cell model an initial 36 gene deletion segments, ranging in size from 10 to 147 genes, were selected and individually simulated (A, B, C). Those producing a dividing *in-silico* cell were combined, simulated, and analysed (B + C). This process is repeated until no more genes can be deleted. (B) The smallest LEGO designed genome removed 140 protein-coding genes resulting in an *in-silico* genome of 261 genes (LEGO_261, 100 replicates). Details of simulations completed and settings are available in the methods and supplementary information.

GAMA Method and Results

The GAMA process is summarised in Figure 2. GAMA performs two sequential design steps (guess and add) followed by repeated cycles of mate steps (referred to as generations).

During the guess step all of the non-essential genes are segmented into 4 sets of ~50 random genes (A, B, C, D in Figure 2). Each is split into 400 subsets, ranging between 50 - 100% of the ~50 gene set. These subsets of genes are knocked-out, and the remaining genome simulated. If a cell successfully divides, its genome is carried forward to the next step.

During the add step one viable subset is randomly selected from each of 2, 3 or 4 of the sets of ~50 genes. These viable subsets are combined into one new

genome and simulated. 3000 new subsets are created and tested, and those that produce a dividing cell are taken forward to the mate step.

The mate step refers to mating in the sense of a genetic algorithm. Only individuals that divided are included and these are assigned a "fitness" based on an objective function (i.e. ranked by the number of genes deleted). A genetic algorithm is traditionally 'seeded' in the first generation, by randomly guessing values of the input parameters, to produce an initial set of fittest individuals. Instead, GAMA selects the fittest 50 viable knockout sets produced by the add stage.
Two of these viable genomes are randomly picked, such that smaller genomes with a larger number of knockouts are more likely to be picked, and labelled parent1 and parent2. A random number of genes (2 to 358) are selected from parent1. The name of the gene and it's deletion status are recorded. The genes not selected in parent1 are identified in parent2 and their deletion status also recorded. The combined list of genes and their deletion status in parents 1 and 2 is used to create a child genome, with some additional random gene knockouts selected from a pool of all protein-coding genes (including essential genes). One thousand children are created per generation (iteration of the mate step). Once all of the children are simulated the list of 50 fittest individuals is updated to take into account new results and carried forward to the next generation.
The stopping criteria of the mate step is once 10 generations have completed without producing a smaller genome.

In total, the smallest GAMA designed genome deleted a maximum of 165 genes, leaving a reduced *in-silico* genome of 236 genes (named GAMA_236), surpassing the results found by the LEGO method and producing a simulated cell which replicates DNA, produces RNA and protein, grows, and divides.

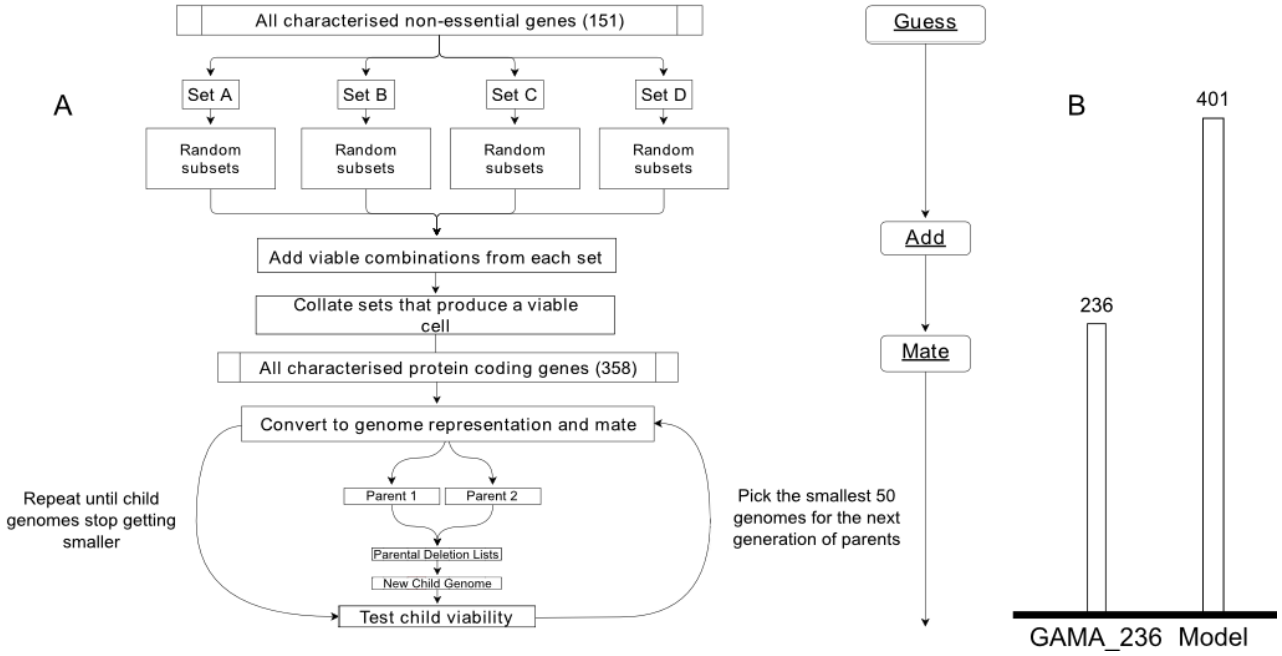

Figure 2 ***GAMA Method for Genome Design: Example and Results*** (A) Using the *Mycoplasma genitalium* whole-cell model to test for genome viability, GAMA reduced the *in-silico* genome in three stages, Guess, Add and Mate. Guess: split all non-essential genes into 4 groups of roughly 38 genes and randomly pick 400 subsets of greater than 24 genes from each group to knockout *in-silico*. Add: between two and four of the groups are randomly selected and then one viable knockout combination is taken from each of them. The union of all these knockout sets is taken to create one larger knockout set which can then be tested *in-silico*. This is repeated 3,000 times. Mate: takes the top 50 smallest genomes that produce a viable cell from the "add" stage and mates them. This mating is done by randomly selecting two parents from the 50 fittest individuals such that smaller genomes are more likely to picked. A new genome is created by randomly combining parts of the two parent genomes. Finally a number of genes are randomly mutated which creates a child genome to test. One generation of the mate stage consists of 1,000 children and once the viability is known the top 50 fittest individuals is updated to take into account the new results. This is repeated to create new generations of child genomes until no smaller genomes are found for 10 generations. Details of simulations completed and settings are available in the methods and supplementary information. (B) A comparison of genome size between the unmodified *Mycoplasma genitalium* whole-cell model genome and the GAMA_236 genome.

Comparing GAMA and LEGO

To analyse whether GAMA and LEGO identify similar minimal genomes created a dendrogram showing the distance between genomes of all of the available genomes of 261 genes produced by GAMA (122) and the smallest LEGO genome LEGO_261 (Figure 3). This showed a significant difference between the two types of reduced genomes, demonstrated by the ARI distances between them (see details in Supplementary Information). As additional evidence to support this divergence, we found four genes removed in LEGO_261 whose individual deletions (in addition to the 165 deletions) in GAMA_236 did not produce

viable *in silico* cells, and six genes removed in GAMA_236 whose individual deletions (in addition to the -140 deletions) in LEGO-261 were not tolerated.

We were interested in how consistently LEGO_261 and GAMA_236 produced a successfully dividing *in-silico* cell. We ran 100 simulations each of an unmodified *Mycoplasma genitalium* whole-cell model, a single gene knockout of a known essential gene (MG_006), LEGO_261, and GAMA_236, and analysed the results to see if they produced a dividing *in-silico* cell (or no division in the case of MG_006) and we used the results to assign a phenotypic penetrance score (how often an expected phenotype occurred). The unmodified *Mycoplasma genitalium* whole-cell model demonstrated a consistent phenotype (99% divided), as did MG_006 (0% divided). LEGO_261 was slightly less consistent (92% divided). GAMA_236 was less consistent, producing a dividing *in-silico* cell 18.5% of the time. This is not unexpected given the greater number of gene deletions, including 29 high essentiality genes (see below).

We were also interested in looking at the range of behaviour displayed by GAMA_236 and LEGO_261. 100 simulations for each genome were simulated, graphed (Figure 4), and analysed. The unmodified *Mycoplasma genitalium* whole-cell model (Figure 4, top row) shows the range of expected behaviour for a healthy cell that successfully divides (in line with previous results [10]). Growth is generally up and to the right throughout the simulation, with most simulations stopping around 10 hours as they divide. Protein production and cellular mass increase over time. RNA production is constantly fluctuating but increases over time. DNA replication (plotted as chromosome) follows a relatively steep gradient, with some simulations delaying the initiation of DNA replication past ~9 hours. The simulation that does not divide is a random wild type cell that did not successfully replicate, presumably due to its particular set of initial conditions. Division occurs as the cell diameter rapidly shrinks, occurring between 6 - 11 hours.

By comparison, LEGO_261 (Figure 4, middle row) displays slower growth over time, which additionally seems to be capped to a lower maximum than wild type. Protein production and cellular mass are generated more slowly and show some erratic behaviour. Protein production begins after a delay of one hour. The range of RNA production is narrower, but the rate is increased. DNA replication takes longer and initiation can occur later (at 11 hours). The simulations that reach a DNA replication value of 2 and continue to 13.89 hours have division defects. Cell division time is shifted to the right, occurring between 8 and 13.889 hours, with a number of simulations failing during DNA replication and cell division and some simulated cells maintaining a constant cellular diameter.

Compared to the unmodified genome, GAMA_236 (Figure 4, bottom row) shows a much greater range of growth rates. Some grow at a faster rate than the unmodified genome, a middle range is comparable to LEGO_261, and a lower range show very low / decreasing growth. GAMA_236 shows a greater delay in the start of protein production, ranging between 2 - 5 hours. TSome simulations show similar behaviour to LEGO_261, while others show a slower rate of protein production. Cellular mass falls into two groups, similar to LEGO_261 and slower, as well as displaying a large number of metabolic

defects that do not produce any growth in mass over period of the simulation (these defects can also be seen in DNA replication and cell division). A reduced range of behaviour is displayed in RNA production, which progresses at a much slower rate. There are two classes of DNA replication phenotype: one group replicates DNA within the timeframe of the unmodified genome; the second group replicates more slowly, some simulations successfully completing the process of DNA replication, others failing to replicate before the simulation ends. Cell division occurs across a greater range of times (6 - 13.889 hours). Metabolically defective cells and cells that do not complete DNA replication can be seen maintaining a constant cell diameter.

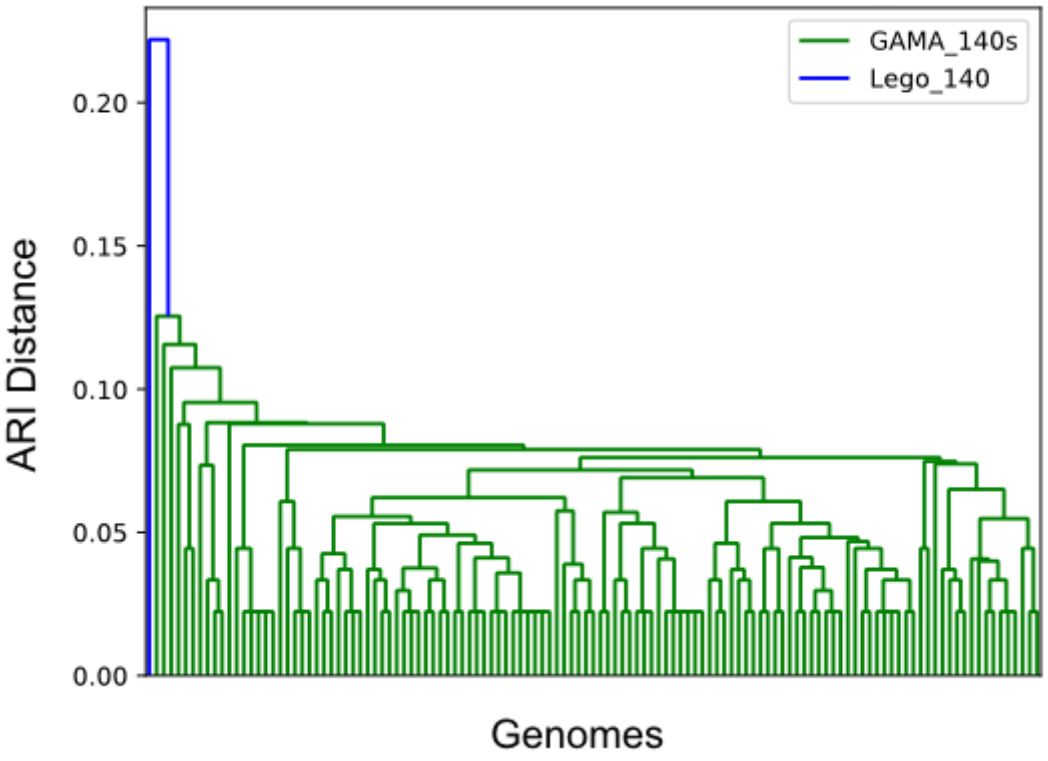

Figure 3 Dendrogram showing the similarity of genomes produced by LEGO and GAMA There were 122 viable genomes found by GAMA that have the same genome size as LEGO_261. The distance metric, ARI distance, between two genomes is 1-ARI which is in the range [0, 2], where ARI is the adjusted rand index. The adjusted rand index (ARI) is a metric that measures the similarity of two sets of elements and is in the range [-1, 1].

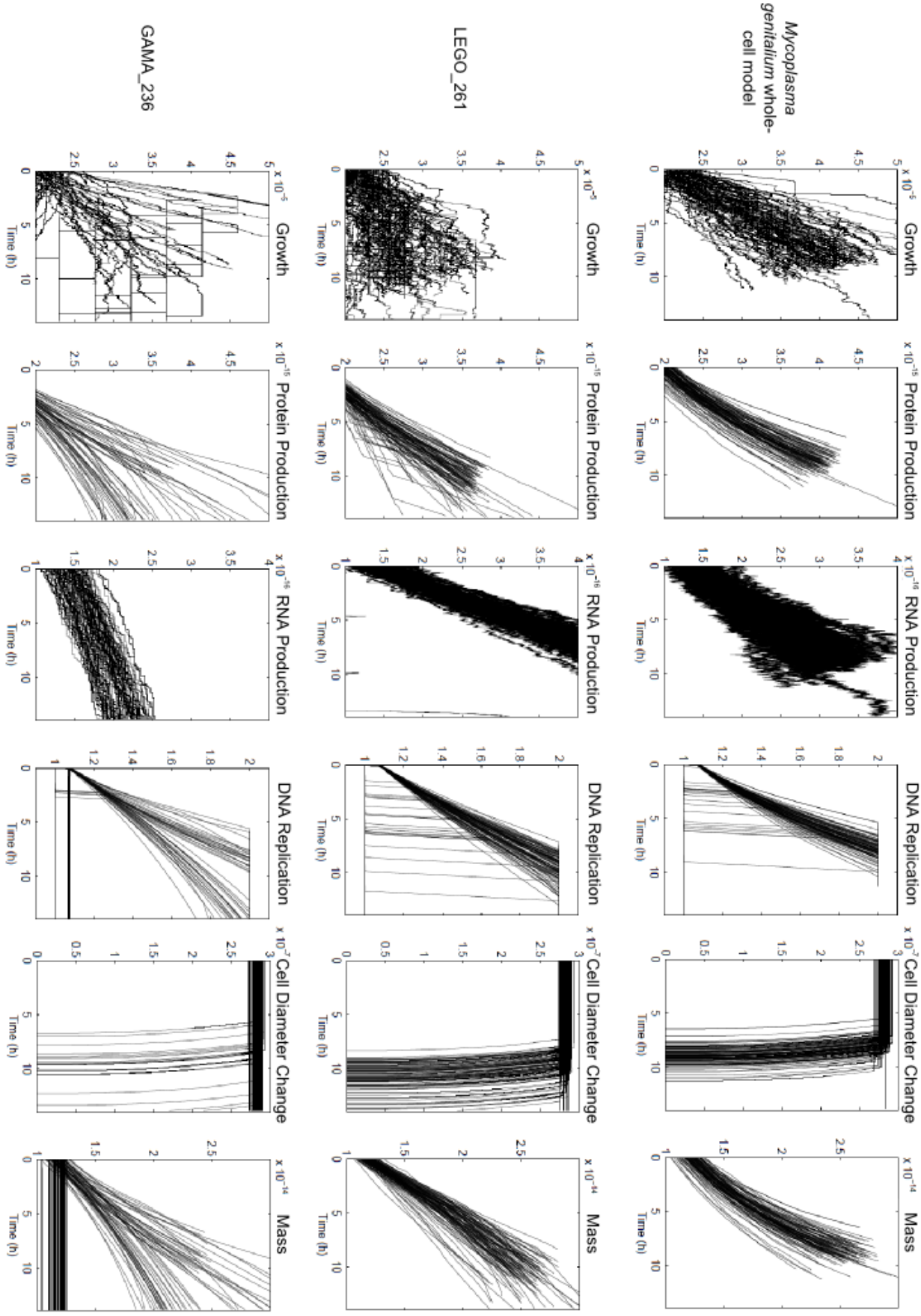

Figure 4 Behavioural Comparison of unmodified *Mycoplasma genitalium* whole-cell model, LEGO_261, and GAMA_236 Unmodified and designed genomes were simulated in the *Mycoplasma genitalium* whole-cell model (100 replicates). Here second-by-second data for six cellular variables is aggregated for the 100 replicates and plotted, giving a phenotypic range for each *in-silico* genome. The unmodified genome shows the range of expected phenotypes for a cell that successfully divides. LEGO_261 and GAMA_236 show deviations in phenotype caused by gene deletions.

Redundant Essential and High Essentiality Genes

As previously stated, we found 4 genes removed by LEGO_261 and 6 genes removed by GAMA_236, that could not be removed from the other without preventing division. We confirmed that these 10 genes (MG_289, MG_290, MG_291, MG_298, MG_310, MG_427, MG_033, MG_410, MG_411, MG_412) could be removed individually from an unmodified *Mycoplasma genitalium in-silico* genome (independently repeating work done by Karr et al.[10]). These results suggest that in single gene knockout studies these would be classified as non-essential. However, our results reported here demonstrate that these are actually redundant essential genes. One additional gene, MG_305, was found to be redundantly essential in both GAMA_236 and LEGO_261.

As a step towards identifying genes that can functionally compensate for these 10 genes we searched prior simulations to find additional genomes from which any of the 10 had successfully been removed. A -79 deletion set deleted the six genes removed by GAMA_236, and a -15 deletion set removed one of the 4 genes (MG_412) removed by LEGO_261, inferring that compensatory genes are not deleted in these sets. We initially focused on MG_412 and the other three genes removed by LEGO_261 (MG_033, MG_410, MG_411), to search for genes with matching biological functions removed by GAMA_236 and the -79 set.

We could not find any matches for MG_033 (named glpF, and involved in the glycerol metabolic process), suggesting that it is being compensated for by seemingly unrelated biological functions [8]. Further research is required to understand why this gene has become essential in the context of the GAMA_236 genome.

MG_411, named pstA, is involved in the phosphate ion transmembrane transport process. MG_410, named pstB, does not have an associated GO term, nor does MG_412. Of the six genes deleted by GAMA_236, two have associated GO terms, MG_289 and MG_291.

We conducted combinatorial gene knockouts between this subset of five genes (MG_289, MG_291, MG_410, MG_411, MG_412) in an unmodified *Mycoplasma genitalium* whole-cell model genome, to see if we could induce synthetic lethality through knocking out redundant essential genes.

MG_289 and MG_291 were found to form a functional group, as were MG_410, MG_411, MG_412. These genes could be deleted individually and in functional groups from an otherwise unmodified *Mycoplasma genitalium* whole-cell genome, still producing a dividing *in-silico* cell. However, any double gene deletion combination that involved one gene from each functional group resulted in a

cell that could not produce RNA, produce protein, replicate DNA, grow or divide.

This suggests possible explanations for the results seen in GAMA_236 and the -79 deletion set, (previously deleting MG_291 and being unable to delete MG_410, MG_411 or MG_412), as well as the results seen in LEGO_261 and the -15 deletion set (previously deleting MG_412 and being unable to delete MG_289 or MG_291).

We are currently investigating the other genes identified as redundantly essential (MG_033, MG_290, MG_298, MG_305, MG_310, MG_427).

Interestingly, we found 29 genes knocked out in GAMA_236 that were defined as essential by single knockout *in silico*, in the genomic context of the unmodified *Mycoplasma genitalium* whole-cell model (see Supplementary Information for gene list). These could be removed in the genomic context of GAMA_236 without preventing division, demonstrating that some essential genes are less essential than others. These would be classed as having high essentiality on Rancati et al's[8] quantitative essentiality scale. Interactions within the remaining genome must be compensating for their removal which requires further research to understand fully.

Discussion

We have created two genome design tools (LEGO and GAMA) that used computational design-simulate-test cycles to successfully produce *in-silico* minimal genomes. Even including the genes of unknown function that are currently not modelled, our minimal genome designs are smaller than *JCVI-syn3.0* (the currently smallest genome produced in the laboratory)[3] and 28 - 53 genes smaller than the most recent predictions for a reduced *Mycoplasma* genome [17]. Additionally, we identified 11 redundant essential genes, one redundant essential gene pairing, and 29 high essentiality genes. We intend to test these results and minimal genome predictions experimentally to ascertain the accuracy of the model and the functionality of our designed genomes.

It was already known that single gene knockout studies are likely to inherently underestimate minimal genome size: redundant genes will be scored in single gene knockouts as non-essential [3,16,17]. The removal of even a single redundant essential pair will prevent your designed genome from surviving. We found 11 redundant essential genes within 359 protein-coding genes, giving a 3% chance of errors in genome design in even this evolutionarily reduced genome. Additionally, reliance on single gene knockout studies narrows the scope of genome design. A broader scope enables unforeseen discoveries, such as the 29 high essentiality genes identified by GAMA, which would not have been targeted for deletion by traditional laboratory methods. We believe that single gene knockout classifications are unreliable for minimal genome design, as essentiality classifications have to take into account genomic context.

There are limitations to our work. Models are not a perfect representation of reality. Our results could be biased by model assumptions and parameters, with some of the construction and validation of the model being based on data from other bacterial species [10]. Our conclusions are dependent on the accuracy of the model, we currently have to assume that the model is internally correct and will be externally generalisable.

Additionally, multigenerational simulations of the whole-cell model (while maintaining genomic context) are currently unavailable. If our predictions are correct they may only be applicable to the first generation of cells.

Finally, there are genes of unknown function that the model cannot account for. These genes may change the genomic context such that a gene that has already been removed would be required for an essential function, preventing our genome designs from being successful. We currently cannot know the impact of these genes until our predictions are tested in the lab.

Here we establish methodologies for computational design-simulate-test cycles that could be applied to other models, including models of other species, and, with refinements, could be used to design genomes optimised for particular functions. Our computational design-simulate-test cycles could be extended to include experimental build and test steps in wet laboratories; the predictions produced by computational design-simulate-test cycles should produce quicker and cheaper laboratory results, which, in turn, will drive model refinements where needed, opening up this area of research to wide and interdisciplinary research communities.

The tools we presented are adaptable to future whole-cell models. An *E.coli* whole-cell model is under development at the Covert Lab in Stanford, and would offer a good opportunity to conduct build and test steps in the lab.

We believe computers and algorithms are the future of genome design and these methods in combination with whole-cell models and new gene editing techniques will allow us to produce cells designed for particular purposes with much greater efficiency and increasing functional understanding.

Author Contributions

C.G, L.M, O.C for attaining initial funding.
C.G, L.M, O.P, O.C, J.R involved in ideation.
O.C development of the initial code, GAMA method, results, and section, Figure 2 and Figure 3.
J.R development of analysis processes, LEGO method, results, GO analysis, redundant essential identification, Table 1, Figure 1, Figure 2, writing paper.
C.G, L.M, O.P editing and feedback on paper.

Acknowledgements

Thanks to Sophie Landon for providing feedback on the paper and ideas presented throughout the research period. LM is supported by the Medical Research Council grant MR/N021444/1 to LM, and by the Engineering and Physical Sciences Research Council grant EP/R041695/1 to LM. OC, LM and CG

are supported by a BrisSynBio, a BBSRC/EPSRC Synthetic Biology Research Centre (BB/L01386X/1), flexi-fund grant. OC is supported by the Bristol Centre for Complexity Sciences (BCCS) Centre for Doctoral Training (CDT) EP/I013717/1;JR is supported by an EPSRC Future Opportunity scholarship.

Statistics

We are currently developing a custom bootstrap statistical method to test our simulation results. We will update this draft with those results when they are complete.

Code Availability

All of the code produced as part of this research will be freely available on Github, which will include a bundled version of the model with our analysis and runner scripts.

Model Availability

The Mycoplasma genitalium whole cell model is available for free from the Covert Lab's Github:https://github.com/CovertLab/WholeCell. We run the model with the standard settings, though we use our own version of the SimulationRunner.m called MGGRunner.m.

---

1. Choe, D., Cho, S., Kim, S. C. & Cho, B. K. Minimal genome: Worthwhile or worthless efforts toward being smaller? *Biotechnol. J.* **11**, 199–211 (2016).
2. Juhas, M., Reuß, D. R., Zhu, B. & Commichau, F. M. Bacillus subtilis and Escherichia coli essential genes and minimal cell factories after one decade of genome engineering. *Microbiology* **160**, 2341–2351 (2014).
3. Hutchison, C. A. *et al.* Design and synthesis of a minimal bacterial genome. *Science* **351**, 1414–U73 (2016).
4. Iwadate, Y., Honda, H., Sato, H., Hashimoto, M. & Kato, J.-I. Oxidative stress sensitivity of engineered Escherichia coli cells with a reduced genome. *FEMS Microbiol. Lett.* **322**, 25–33 (2011).
5. Hirokawa, Y. *et al.* Genetic manipulations restored the growth fitness of reduced-genome Escherichia coli. *J. Biosci. Bioeng.* **116**, 52–58 (2013).
6. Zhou, J., Wu, R., Xue, X. & Qin, Z. CasHRA (Cas9-facilitated Homologous Recombination Assembly) method of constructing megabase-sized DNA. *Nucleic Acids Res.* **44**, e124 (2016).

7. Reuß, D. R. *et al.* Large-scale reduction of the Bacillus subtilis genome: consequences for the transcriptional network, resource allocation, and metabolism. *Genome Res.* **27**, 289–299 (2017).
8. Rancati, G., Moffat, J., Typas, A. & Pavelka, N. Emerging and evolving concepts in gene essentiality. *Nat. Rev. Genet.* **19**, 34–49 (2018).
9. Szostak, J. W., Bartel, D. P. & Luisi, P. L. Synthesizing life.
10. Karr, J. R. *et al.* A whole-cell computational model predicts phenotype from genotype. *Cell* **150**, 389–401 (2012).
11. Fraser, C. M. *et al.* THE MINIMAL GENE COMPLEMENT OF MYCOPLASMA-GENITALIUM. *Science* **270**, 397–403 (1995).
12. Sanghvi, J. C. *et al.* Accelerated discovery via a whole-cell model. *Nat. Methods* **10**, 1192–1195 (2013).
13. Purcell, O., Jain, B., Karr, J. R., Covert, M. W. & Lu, T. K. Towards a whole-cell modeling approach for synthetic biology. *Chaos* **23**, 025112 (2013).
14. Kazakiewicz, D., Karr, J. R., Langner, K. M. & Plewczynski, D. A combined systems and structural modeling approach repositions antibiotics for Mycoplasma genitalium. *Comput. Biol. Chem.* **59 Pt B**, 91–97 (2015).
15. Glass, J. I. *et al.* Essential genes of a minimal bacterium. *Proc. Natl. Acad. Sci. U. S. A.* **103**, 425–430 (2006).
16. Dobzhansky, T. Genetics of natural populations; recombination and variability in populations of Drosophila pseudoobscura. *Genetics* **31**, 269–290 (1946).
17. Glass, J. I., Merryman, C., Wise, K. S., Hutchison, C. A., 3rd & Smith, H. O. Minimal Cells-Real and Imagined. *Cold Spring Harb. Perspect. Biol.* (2017). doi:10.1101/cshperspect.a023861