Network Modeling Analysis in Health Informatics and Bioinformatics (2024) 13:22
https://doi.org/10.1007/s13721-024-00455-4

REVIEW ARTICLE

Technologies for design-build-test-learn automation and computational modelling across the synthetic biology workflow: a review

Richard Matzko[1] · Savas Konur[1]

Received: 5 January 2024 / Revised: 9 March 2024 / Accepted: 18 March 2024

Abstract

Motivated by the need to parameterize and functionalize dynamic, multiscale simulations, as well as bridge the gap between advancing in silico and laboratory Synthetic Biology practices, this work evaluated and contextualized Synthetic Biology data standards and conversion, modelling and simulation methods, genetic design and optimization, software platforms, machine learning, assembly planning, automated modelling, combinatorial methods, biological circuit design and laboratory automation. This review also discusses technologies related to domain specific languages, libraries and APIs, databases, whole cell models, use of ontologies, datamining, metabolic engineering, parameter estimation/acquisition, robotics, microfluidics and touches on a range of applications. The discussed principles should provide a strong, encompassing foundation for primarily dry laboratory Synthetic Biology automation, reproducibility, interoperability, simulatability, data acquisition, parameterization, functionalization of models, classification, computational efficiency, time efficiency and effective genetic engineering. Applications impact the design-build-test-learn loop, in silico computer assisted design and simulations, hypothesis generation, yield optimization, drug design, synthetic organs, sensors and living therapeutics.

Keywords Automation · Bioinformatics · Chemical reaction networks · Computer assisted design · Computational modelling · Databases · Genetic engineering · Machine learning · Synthetic biology · Systems biology

1 Introduction

The future of Synthetic Biology (SB) was seen as a model-based (Zhang et al. 2017) engineering discipline (Zhang et al. 2017; Konur and Gheorghe 2015; Xia, et al. 2011) involving the reprogramming of cells (Nielsen, et al. 2016), applicable to biotechnology, achieved primarily by DNA manipulation (Chandran et al. 2010). SB "parts" containing DNA sequence information can be combined together into devices for modular reuse (Bilitchenko et al. 2011) for artificial genetic recombination. This involves DNA construct production from small circuits up to the genome scale (Storch et al. 2020), where genetic constructs refer to composites of genetic sequences that can contribute to the overall system behaviour at various localizations. This review analysed and elucidated these aspirations, emphasizing automation provided by computational methods in manipulating bioregulatory circuitry, embedded systems, robotics, microfluidics, and the potential of machine learning (ML) within the workflow. Addressing these challenges had direct implications in our current research pertaining to SB Computer Assisted Design (CAD) (Matzko et al. 2023; Konur et al. 2021) with consideration for data acquisition, the implementation of small orthogonal or genome scale models, laboratory automation and ML. Given that automation was proposed as providing efficiency in design and application as compared to manual labour, as well as the potential for decreased error rate (Gurdo et al. 2023), research into cost-effective, high-throughput design-build-test-learn (DBTL) cycles for parameter space exploration should provide laboratories with research advantages. In addition to in silico modelling, this paper addresses such automation options.

✉ Richard Matzko
rmatzko@bradford.ac.uk

✉ Savas Konur
s.konur@bradford.ac.uk

[1] School of Computer Science, AI and Electronics, University of Bradford, Bradford BD7 1HR, UK

Published online: 03 May 2024

Our ongoing research continued to expand on prior published work in multicellular simulation modelling (Matzko et al. 2023), and Synthetic Biology CAD research related to facilitating the design of bioregulatory constructs and gene regulatory circuits via Infobiotics Workbench (Konur et al. 2021). The trajectory of this work would relate to the pursuit of the extension of Synthetic Biology CAD to the multicellular modelling domain. Such models, particularly involving kinetics, were considered to be "virtually absent" (Gurdo et al. 2023). However, operating via the School of Computer Science AI and Electronics and in collaborative exchanges with the Chemical Biology department, it was our conviction to maintain a translational component through the lens of bioinformatics approaches. That said, computational SB CAD would have many overlaps with Systems Biology. Given the above rationale it is clear how the topics of this review are connected involving relevant data standards, databases and data mining for parameterization, network analysis and modelling methods, whole cell models, minimal genomes, biochemical pathways and network model generation, SB suites, ML, laboratory automation, enabling organizations, combinatorial construct design languages, circuit design, genetic optimization and genetic construct assembly automation. Rife with information, it is our contention that this work can provide beneficial insights for many researchers, and a key intention of this work is as a robust, noteworthy reference in the fields of computational biological modelling and translational, automated Synthetic Biology.

SB engineering has been conceptually subdivided into DNA synthesis, DNA optimization, genetic component determination, construct design from the components, and transformation and transfection into host chassis/organisms (Oberortner et al. 2017). Computational resources have been categorized into specification, design, assembly and building, testing and analysis, data, simulation and sequence editing (Appleton et al. 2017). SB has vast potential for design across the extreme complexity of biological systems. In fact, SB can even be applied to hybrid systems, for example a bioreactor contains mechanical components within its operations. As noted in the literature, SB applications might utilize part/plasmid combinations, biochemical/genetic network languages, construct design languages, Multiplex Automated Genome Engineering, RBS (Ribosomal Binding Site) design, CRISPR/Cas9, liquid-handling automation, high-throughput cloning, microfluidic device design automation, microfluidic milling and lithography, primer design, flow cytometry, deterministic and stochastic time-course simulations, multicellular simulations, reaction–diffusion, sequence alignment (e.g. BLAST), restriction enzyme cut predictions, codon optimization and rational pathway design (e.g. via OptFlux, Cobra 2.0, OptForce (Kahl and Endy 2013)). Software popularity has varied over time, for example Vector NTI had fallen significantly in use, where a modern alternative is Geneious, a molecular biology and sequence analysis tool (Dotmatics. Geneious by Dotmatics. 2023), featuring various molecular cloning methodologies, mapping and de novo assembly, primer design, sequence analysis and phylogenetics. Such tools can be used in SB CAD, e.g. SnapGene (Dotmatics. Snapgene 2022) can be used for cloning and construct generation simulations, Gateway cloning simulations, Gibson Assembly and primer-directed mutagenesis.

The domain of SB is extensive and challenging with great potential to tackle unaddressed concerns, e.g. in healthcare. This review identified in silico and laboratory automation opportunities vital to the design-build-test-learn workflow with the intention to provide the reader with clarity, scope and modernity, particularly from the computational perspective. By assessing cutting-edge ML breakthroughs with the essentiality of combinatorial practices, alongside automated hardware and bioregulatory network and genetic manipulations, this review offered a unique understanding of the DBTL concept, elucidating concepts across SB, bioinformatics, systems biology and biotechnological hardware. The paper serves as a reference for technologies across SB and computational modelling workflows. This review work has already yielded us practical software engineering bioinformatics research outcomes in the form of a cyto-histological genetics encyclopedia and network explorer, BioNexusSentinel, available on GitHub (Matzko 2023), which demonstrated that targeted computational biology software engineering was made possible by insights from this review, and that this review could hence be revisited for selective updates, expansions and concepts. The technician/researcher is encouraged to make informed decisions regarding the presented resources, with scope for expanding on and developing custom approaches from the extensive subject-specific insights that this paper provides, whether in silico or translational.

It is our contention that this review provides uniquely integrated insights spanning a host of the many vital disciplines, providing a unique perspective on the vast range of opportunities and challenges that are faced for generating increasingly complex Synthetic Biology engineered solutions. The review is written to support such engineers and interested parties in understanding the many challenges by integrating insights from data standards, modelling, genetic design, circuit design, ML, assembly planning, combinatorial methods, in silico design automation and laboratory automation at the hardware and software levels. Certainly, it is felt that this review offers an exceptional scope, and increased clarity on the DBTL concept than previous work that we have encountered, as well as deeper insights informatically, including through in silico modelling, through to robotic translation. Our offering provides intricate insights, for instance including biological domain specific languages, libraries and APIs, databases, whole cell models, parameter estimation/acquisition for evaluating and predicting systems, generously compiled into this concise paper. The implications of this work are significant. With many medical challenges still remaining unresolved, it is vital to consider this paper's potential to stimulate thinking for in silico computer assisted design, hypothesis generation and testing, and the wide range of technological benefits that Synthetic Biology has the potential to bring about, whether through optimized smart therapeutics, biofabrication or otherwise.

Hence, our major contribution with this holistic and carefully formulated review is to provide the reader with accessibly communicated resources to foster developments towards translatable, automated Synthetic Biology pipelines considering the DBTL cycle. The research methodology and contents of the paper are discussed in Sect. 2.

2 Research methodology

This paper details a literature review related to ongoing technical work at our institute, made accessible to a wider audience and carried out from the dry laboratory perspective. This review aimed to augment our research regarding the extension of bioregulatory time-course simulations in Synthetic Biology CAD software (Konur et al. 2021) spatially into multicellular simulations (Matzko et al. 2023), whilst maximizing the objective of translatable computational CAD given collaborative interactions with the Chemical Biology department. Translatability would be considered as far as downstream robotic automation within the DBTL loop. Thus the research would span the DBTL cycle.

Data standards (Sect. 3.1) would be required to house the informatics from which upstream to downstream translation could manifest, and this paper details many such standards. Naturally, databases (Sect. 3.2) would need to be sought to provide the relevant data in useable form. And where data might not be in readily useable form, data mining could be considered (Sect. 3.3), particularly with the ongoing revolution in artificial intelligence. Upon the foundational discussion of data we investigated modelling implications enabled by these data standards and data acquisition strategies (Sects. 4.1, 4.2, 4.3, 4.4) and the state of the art in open source Synthetic Biology software suites (Sect. 4.5). Having discussed the logistical hierarchy from data to modelling, the technical translational component could be addressed. Hence, the DBTL loop was introduced based in the literature (Sect. 5) along with relevant ML for the domain (Sect. 5.1) with implications in affecting the loop. With the observation of ongoing manual work in the Chemical Biology laboratory, automation was explored as part of an investigation into accelerating and improving these methodologies (Sect. 5.2), and these ideas would be expounded on through the literature by exploring combinatorial design strategies (5.3). Combinatorial strategies were deemed crucial in high throughput experimentation, which is associated with bioregulatory genetic circuit design principles (Sect. 5.4), finally culminating in the necessary considerations of genetic optimization (Sect. 5.5) and the automated planning of assembly protocols to physically generate genetic constructs of interest (Sect. 5.6). The essentiality of experimental data acquisition is also discussed in the context of Sect. 5.

Search criteria would include themes of artificial intelligence, ML and datamining for Synthetic Biology, natural language processing, systems biology model archives, Synthetic Biology automation, Synthetic Biology parts repositories, systems biology tools, Synthetic Biology tools, analytical methods including Gillespie Algorithms and Flux Balance analysis, kinetics parameterization, genome scale and whole cell models, genetic optimization and protein folding. The research was executed in the context of wider multicellular simulation research (Matzko et al. 2023; Matzko 2023) and within the context of the Chemical Biology laboratory at the University of Bradford, which from our observations evidenced a heavily manual and iterative, low-throughput research cycle, albeit with sophisticated analytical modalities and careful experimental planning. This paper documents review work intersecting both these requirements.

The research drew from attendances at the 9th International Work-Conference on Bioinformatics and Biomedical Engineering June 2022 in Gran Canaria (Matzko et al. 2022), Synthetic Biology UK November 2022 and The Festival of Genomics & Biodata January 2024 in London.

3 Data in synthetic biology

3.1 Data standards

To sustain reproducibility, engineering fields utilize worksheets and biology uses minimal information standards, e.g. MIAME for microarrays and MIFlowCyt for flow cytometry (Myers et al. 2017). SB standards were recommended for describing parts, genetic construct designs, sequences, assembly methods, vectors, integration points for transformation, CRISPR-based integration and host/chassis organism identity. A lack of quantitative parts datasheets was proposed to be a limiting factor in SB CAD design (Lux et al. 2011).

Many exchange standards are built upon the Extensible Markup Language (Swat, et al. 2009). The Systems Biology Markup Language (SBML) (SBML 2022) represents biological/biochemical networks, including mathematically, and has been harnessed in automated methods (Keating et al.

2020). Tools and APIs can validate, analyse and simulate SBML models, which are commonly simulated via ordinary differential equations (ODE) and stochastic Gillespie algorithms. SBML can harness ontologies or semantic web technologies allowing software to explore network metadata. SBML can be translated to and from domain specific languages (DSLs) such as Antimony (Smith et al. 2009) and IBL (Konur et al. 2021), but typically lacks genetic details (Baig, et al. 2020). By contrast, the Synthetic Biology Open Language (SBOL) allows hierarchical, modular, annotated and extensible genetic design representations (Appleton et al. 2017). The FASTA format primarily contains nucleotide or amino acid (AA) sequence information, whilst GenBank and Swiss-Prot offered annotation capabilities. SBOL can also represent experimental details, unique identifiers, ontologies and uniform resource identifiers, including for external models, and was put forward to address GenBank format limitations regarding representing experimental data and genetic construction documenting (Ham et al. 2012).

Other formats might be encountered whilst investigating SB modelling/data. In the multicellular domain, NUFEB (Li et al. 2019) used VTK, POVray and HDF5 (.h5) output formats. Meanwhile, the COMBINE standard can be used to archive various standards for sharing (Myers et al. 2017). Pretrained ML model formats can depend on the framework or format of choice, e.g..h5,.pb,.safetensors,.pt,.pth,.onnx.

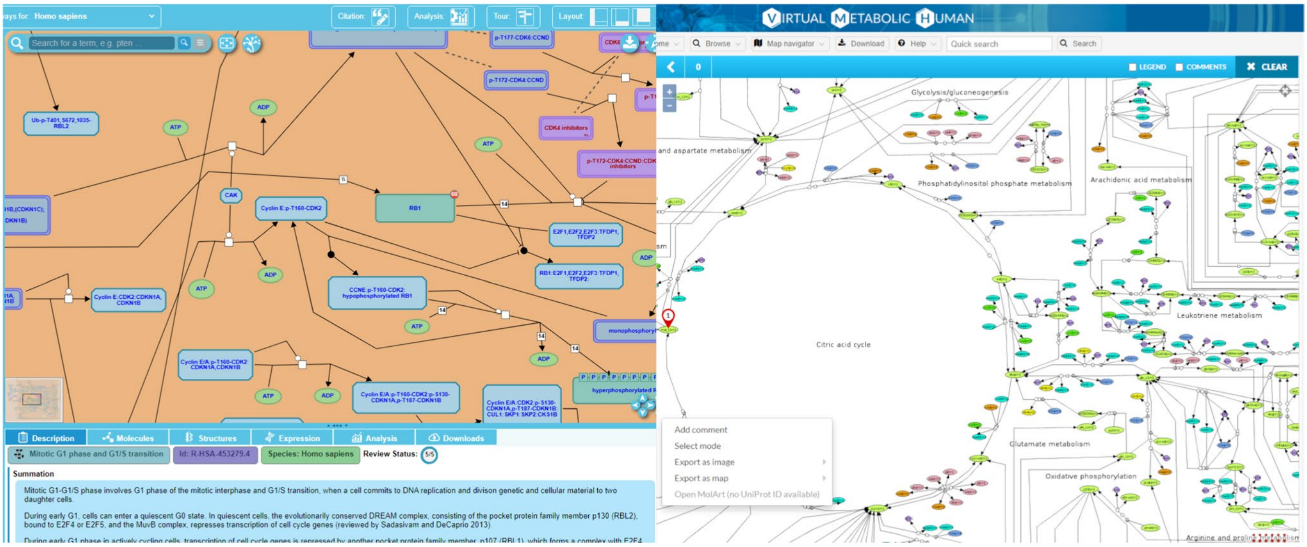
3.2 Databases

Computational modelling for SB requires experimental data, ML tends to require large amounts (Rampasek and Goldenberg 2016; Perrakis and Sixma 2021). Data in literature and within online databases includes chemical reaction pathways, kinetics data, protein data, genomic data and expression data. To fulfil its potential both in de novo design and specific applications, e.g. medical, SB must fully explore applicable data and not confine itself to parts repositories.

The NCBI archive (Oberortner et al. 2017) provided access to genomes, with AA and nucleotide sequence data available in FASTA and GenBank formats. Design repositories for SB, such as SynBioHub (McLaughlin et al. 2018), and the iGEM Registry of Standard Biological Parts were available. JBEI-ICE (Joint BioEnergy Institute's Inventory of Composable Elements) was a registry for access to biological parts (Ham et al. 2012) with a collection of connected tools. Computational model repositories included BioModels (Biomodels Repository 2022), BiGG Models (Systems_Biology_Research_Group. BiGG Models 2023) and the CellML repository (The_CellML_Project. CellML Model Repository 2022; Büchel et al. 2013). An annotated SBOL parts registry was SBOLme for metabolic engineering (Myers et al. 2017). MetaCyc, KEGG, the Nature Pathway Interaction Database (PID), Reactome and WikiPathways contained curated biochemical pathways (Büchel et al. 2013). With the Human Metabolome Database, human metabolite data was searchable including 3D structures, diseases, proteins, pathways and reactions (Wishart, et al. 2018). The Protein Data Bank (PDB) and UniProt were available as protein-related resources. Specialized databases like the Transporter Classification Database also existed. The ChEBI database provided chemical data of biological interest (Keating et al. 2020). A detailed exploration of EMBL-EBI and NCBI can be encouraged. The Pan-Cancer Atlas (Miles and Lee 2018) aimed to assist precision medicine. gnomAD database has been referenced in phenotyping studies (Rosenhahn et al. 2022) and provides allele population scale frequencies, also classified for pathogenicity.

The Reactome pathway browser (Reactome. Reactome Pathway Browser. 2022) provided a map separated according to cellular functions, allowing the identification of annotated genetic mutations associated with disease phenotypes. Reactome was arguably more ergonomic than Recon3D's (Brunk et al. 2018) extensive interactive browser (Recon 2022) (Fig. 1). The Reactome Knowledgebase is manually curated (Gillespie et al. 2022) and concerns molecular data emphasizing human disease and physiology; detailing gene expression and mutations. Reactome possessed information on 52.5% of the predicted protein-coding human genome (10,726 genes). Reactome utilized Gene and Disease Ontology annotations and Gene Set Analysis was supported, with datasets available from ExpressionAtlas and Single Cell ExpressionAtlas. Reactome used Systems Biology Graphical Notation (SBGN) for its pathway diagrams, visualized using Cytoscape.js. The druggable genome could be visualized with annotations provided by Reactome IDG. In a March 2024 email from QIAGEN, a company operating with hundreds of millions of dollars, they stated the connectivity of Reactome pathways to their commercial QIAGEN Ingenuity Pathway Analysis (QIAGEN IPA) service (QIAGEN. QIAGEN Ingenuity Pathway Analysis (QIAGEN IPA). 2024).

Expression Atlas (EBML_EBI. Expression Atlas. 2022) and the Human Protein Atlas (HPA) (Human_Protein_Atlas. The Human Protein Atlas 2022) were resources for phenotypic expression profiles. The HPA contained histological section graphics with marker expression levels, protein function details, survival rates, and used external resources such as the Cancer Genome Atlas. RNA-seq data was available, which uses Next Generation Sequencing to sequence the transcriptomic profile of cells. Transcriptomics data acquisition can also arise from DNA microarray technology (Gurdo et al. 2023), however the use of probes compared to RNA-seq restricts detection to known sequences. Protein localization/compartmentalization can be associated with specific functions, which cells achieve via trafficking (Watson et al. 2022). Localization data was available at the Gene Ontology Cellular Component and Jensen COMPARTMENTS


ays for Homo sapiens
Citation
Analysis
Tour
Layout
Search for a term, e.g. pten
Ub-p-T401,S672,1035-RBL2
ATP
ADP
CAK
Cyclin E:p-T160-CDK2
RB1
p-T177-CDK6:CCND
p-T172-CDK4:CCND
CDK4 inhibitors
p-T172-CDK4:CCND:CDK inhibitors
E2F1,E2F2,E2F3:TFDP1,TFDP2
RB1:E2F1,E2F2,E2F3:TFDP1,TFDP2
CCNE:p-T160-CDK2:hypophosphorylated RB1
Cyclin E:CDK2:CDKN1A,CDKN1B
Cyclin E/A:p-T160-CDK2:CDKN1A,CDKN1B
Cyclin E/A:p-T160-CDK2:p-S130-CDKN1A,p-T187-CDKN1B
Cyclin E/A:CDK2:p-S130-CDKN1A,p-T187-CDKN1B:CUL1:SKP1:SKP2:CKS1B
monophosphoryl
hyperphosphorylated R
14
Description
Molecules
Structures
Expression
Analysis
Downloads
Mitotic G1 phase and G1/S transition
Id: R-HSA-453279.4
Species: Homo sapiens
Review Status: 5/5
VIRTUAL METABOLIC HUMAN
Browse
Map navigator
Download
Help
Quick search
Search
LEGEND
COMMENTS
CLEAR
Glycolysis/gluconeogenesis
and aspartate metabolism
Phosphatidylinositol phosphate metabolism
Arachidonic acid metabolism
Citric acid cycle
Leukotriene metabolism
Glutamate metabolism
Oxidative phosphorylation
Arginine and proline metabolism
Add comment
Select mode
Export as image
Export as map
Open MolArt (no UniProt ID available)

Summation

Mitotic G1-G1/S phase involves G1 phase of the mitotic interphase and G1/S transition, when a cell commits to DNA replication and divison genetic and cellular material to two daughter cells.

During early G1, cells can enter a quiescent G0 state. In quiescent cells, the evolutionarily conserved DREAM complex, consisting of the pocket protein family member p130 (RBL2), bound to E2F4 or E2F5, and the MuvB complex, represses transcription of cell cycle genes (reviewed by Sadasivam and DeCaprio 2013).

During early G1 phase in actively cycling cells, transcription of cell cycle genes is repressed by another pocket protein family member, p107 (RBL1), which forms a complex with E2F4

Fig. 1 Reactome (Left) and Recon3D via Virtual Metabolic Human (Right) are impressive resources for reaction networks with the capacity to download detailed computational models

databases. The HPA was considered the gold-standard in protein localization.

The SABIO-RK online database offered scientist-curated biochemical kinetics data (Rojas et al. 2007), with reaction information obtained via databases including KEGG. Parameters included rate/equilibrium/dissociation/inhibition constants and maximal velocities (Golebiewski et al. 2007). Export could be in the SBML format (Rojas et al. 2007) and SABIO-RK has been used for kinetic model generation (Büchel et al. 2013; Dräger et al. 2015). Integration of SABIO-RK queries was reported for CellDesigner and SYCAMORE (Golebiewski et al. 2007).

This subsection noted many useful resources, however with countless bioinformatics resources undoubtedly many were excluded from this compilation. Our research implicated the importance of experimentally derived pathway networks coupled with omics resources, with different types of omics potentially presenting with different layers of regulatory control, and hence different perspectives on the true state of a biological system. In fact, the current biological state is the result of the physical molecular configuration resultant of the temporally past upstream interactome. It is the task of the biological modeller or researcher to understand the implications of experimental assays and interpret bioinformatics resources at different regulatory levels to infer a complete picture of the present state. For example, RNA-seq data is evidently highly popular, but restricted to the transcriptome, with uncertainty to the true downstream state of the system, discernible from the metabolome or proteome. Neither does RNA-seq represent the true capacity of a given genome, only that which is transcriptionally active in the present or past. Indeed, a range of techniques are available for data collection across the omics (Gurdo et al. 2023).

3.3 Data mining

Biological text mining tools are capable of "named entity recognition" (NER) and functional enrichment analysis (Baltoumas, et al. 2021). Functional enrichment analysis aims to identify genes that might be over or under expressed in particular phenotypes, e.g. via g:Profiler2 and aGOtool. NER can use ontologies and "concept-normalization" to map a word or phrase to a term (Pattisapu, et al. 2020). OnTheFly utilized the EXTRACT tagging service for this purpose (Baltoumas, et al. 2021), and also possessed Optical Character Recognition. aGOtool could locate documents related to identified genes and proteins, achieved through a text corpus from PubMed. The STRING and STITCH APIs could be used to assess protein interactions with resulting node-based graphs such as of interaction evidence and binding affinities.

2023 was a breakout year for machine learned large language models (LLMs) (Else 2023) trained on large volumes of "human-generated text", an eminent example being ChatGPT by OpenAI. Such technology was proposed to serve fields as diverse as stem cell research (Cahan and Treutlein 2023). Biomedical language models included BioBERT, PubMedBERT and BioGPT (Luo, et al. 2022), trained on vast corpora of biomedical literature. BioGPT is a domain-specific generative Transformer language model trained on 15 million PubMed abstracts. BERT utilized "masked language modelling" with probabilistic sentence predictions. Instead, the Generative Pre-Trained Transformer (GPT) would predict word tokens, including via Byte-Pair encoding (Vaswani, et al. 2017). LLMs can also assist with programmatic tasks. We have considered the possibility of extending our ongoing research (Matzko 2023) through the use of LLMs. Graph neural networks are another domain that could be considered (Gurdo et al. 2023).

4 Biochemical/bioregulatory modelling and analysis methods

In order to perform simulations, which have hypothesis generation and predictive potential, models must be established. This section details simulation and chemical reaction network (CRN) resources and principles, as well as introducing Synthetic Biology CAD software for genetic circuit design.

4.1 Network analysis and modelling methodologies

Simulators solve biochemical reactions and transitions by operating on syntactically compatible models. An example is libRoadRunner (Choi et al. 2018) with stochastic and ODE support (Available from 2022). NGSS (Next Generation Stochastic Simulator) (Sanassy et al. 2015) for Gillespie algorithms was discussed in our previous work (Matzko et al. 2023; Konur et al. 2021), alongside SSAPredict for algorithm selection based on model topology. Reaction-based models can be interrogated by parameter estimation, sensitivity analysis and parameter sweep analysis (Riva et al. 2022) at considerable computational expense. Thus, the move to GPU from CPU architecture was encouraged. Model analysis can be performed via numerical analysis, e.g. on matrix representations of state, or statistical analysis on stochastic runs (Appleton et al. 2017). Kinetic parameter estimation is possible via genetic algorithms, particle swarm and hill-climbing methods. BioPSy and COPASI software provided parameter estimation capabilities. Sensitivity analysis assesses the dynamics of a system relative to its parameters.

Gene regulatory networks involve the manipulation of "cis-regulatory module" DNA sequences for the activation or inhibition of transcription (Delile et al. 2017), and have been described as bipartite directed graphs (Yaman et al. 2012) modellable in Boolean fashion or through probabilistic differential equations (Delile et al. 2017). Contrasted with kinetics models, Boolean models can provide a convenient simplification (Karagöz et al. 2021) with utility in modelling domains such as signalling cascades (Letort et al. 2019) or phenotypic states (Rubinacci et al. 2015).

Stochastic simulation algorithms (SSAs), whilst computationally intensive by contrast to deterministic ODEs, are said to produce accurate simulations retaining the inherent stochasticity of biological metabolic networks (Sanassy et al. 2015). This arises from their discrete modelling contrasted to the continuous nature of deterministic ODEs. Classical kinetics was considered unsuitable for genetic regulatory systems, which involve large fluctuations in species counts (Appleton et al. 2017). Stochastic simulations assess propensities of reactions over successive infinitesimal time intervals, rendering them computationally expensive under conditions of high propensity. Hence the existence of hybrid-algorithms using both stochastic and ODE methods in COPASI (Hoops et al. 2006). The argument was made for the use of bond graphs in dynamic biological modelling (Pan et al. 2021) to correct for thermodynamic inconsistencies, e.g. via BondGraphTools for Python. A major challenge to kinetics modelling besides computational expense is the limited availability of experimentally determined kinetics data. Kinetics modelling was thus deemed "cost-prohibitive" (Gurdo et al. 2023). However, a lack of kinetics data was considered a limitation in translatable, cost effective modelling for certain expression systems. The possibility of using machine learning to enhance kinetics parameterization is noted in Sect. 5.1.

Flux balance analysis (FBA) can guide metabolic engineering of interacting pathways (Sekiguchi et al. 2021). FBA is a kinetic rate free, constraint-based approach utilizing an objective function (Motamedian et al. 2017) that mathematically analyses the flow (e.g. mmol/gDW/hr) through a metabolic network (Orth et al. 2010), associated with the field of fluxomics (Gurdo et al. 2023). For growth, the objective function may be the maximization of biomass (Motamedian et al. 2017; Dukovski et al. 2021). FBA has been used to predict missing reactions and gene knockouts for optimized end-product formation, e.g. knockouts by modulating upper and lower flux bounds (Rowe et al. 2018). However, without kinetic parameters, chemical concentrations are undefined and FBA is confined to steady state evaluations (Orth et al. 2010). FBA tools included Escher-FBA, OptFlux, COBRA Toolbox, COBRApy, PSAMM and FAME (Rowe et al. 2018). COBRA stands for constraint-based reconstruction and analysis (Gurdo et al. 2023). FBA optimization of flux values via objective function at the genome-scale was considered to be extremely rapid even on conventional hardware (Dukovski et al. 2021). FBA uses a stoichiometric matrix with rows of metabolites and columns of reactions to simulate under a steady state assumption. However, a limitation of FBA was described as a lack of "explicit gene regulation". Also FBA presents with flux inaccuracies (Gurdo et al. 2023). Amongst FBA variants, thermodynamic flux analysis is an alternative that considers the Gibb's free energy to drive reactions, such as via the pyTFA package (Lent et al. 2023). Due to the Michaelis Menten proportionality between Vmax and enzyme concentration [E], in this method perturbations of Vmax would be used to simulate variable [E] under factors such as assumed promoter strength for the enzyme.

COPASI (COPASI. COPASI 2022) is an open-source biochemical simulator (Hoops et al. 2006), with GUI (Graphical User Interface) version, capable of model editing and analysis. Operating on CRNs, COPASI has deterministic ODE capabilities, stochastic algorithms, ODE/stochastic hybrid methods, steady state computations, stoichiometric network analysis, sensitivity analysis, metabolic control analysis, optimization, parameter estimation and flux analysis. Kinetic functions could be defined and chosen from an integrated library. Optimization used objective functions, steepest descent, genetic algorithms and evolutionary strategies for maximizing or minimizing model variables.

4.2 Whole cell models

Recon3D may be the most extensive public human metabolic network model, containing 3,288 open reading frames, 13,543 reactions, 4140 metabolites (Brunk et al. 2018) and 12,890 protein structures. Contrast this scale to EcoCyc-18.0-GEM (Weaver et al. 2014) for E. coli and Path2Models (Büchel et al. 2013) in Fig. 2. Other genome scale metabolic reconstruction models for E. coli and other organisms are available on BiGG Models (Systems_Biology_Research_Group. BiGG Models 2023). Recon3D could be explored on the Virtual Metabolic Human website (VMH. Virtual Metabolic Human. 2022), including via Recon Map 3 (Recon 2022). Pathway enzymes could be cross-referenced with databases such as KEGG, PDB, CHEBI, PharmGKB and UniProt via external links.

Recon3D utilized a subset (17%) of human proteins from UniProt to generate a 10,600 reaction computational model made available at BiGG models (UCSD_SBRG. BiGG Models. 2019). Recon3D possessed 3D protein structural information from the PDB and included atom-scale models produced through homology modelling via protein sequence alignment. Metabolite structures were included from various sources. Structural data was hence achieved for 85% of the human metabolome, including the aforementioned 12,890 protein structures. Drug metabolic perturbation effects were assessed, assisted by resources such as the Connectivity Map (Broad_Institute 2022).

4.3 Minimal genomes

Minimal genomes can present as a starting point for developing synthetic biological systems. Mycoplasma genitalium contains only 525 genes (Sleator 2016). Comparisons with other bacteria provided rationale for estimating 256 essential genes, whilst other methods suggested 375 genes via transposon mutagenesis data. JCVI-syn3.0 was a physiologically stable synthetic cell developed with an approximately minimal genome, based on Mycoplasma mycoides (Rees-Garbutt et al. 2020). 240 essential genes were identified, along with quasi-essential genes numbering 229 with minor or major cell abnormalities. The method utilized the Tn5 transposase.

The JCVI-Syn3.0 researchers computationally assessed tens of thousands of gene knockouts for implementation with Mycoplasma genitalium ATCC 33530/NCTC 10195. The model was parameterized from 900 publications and 1900 experimental observations and such models of Mycoplasma genitalium are perhaps the most complete of any cell. Minesweeper and GAMA algorithms performed deletions with subsequent simulation ensuring that division still occurred in silico. These algorithms produced tens of thousands of

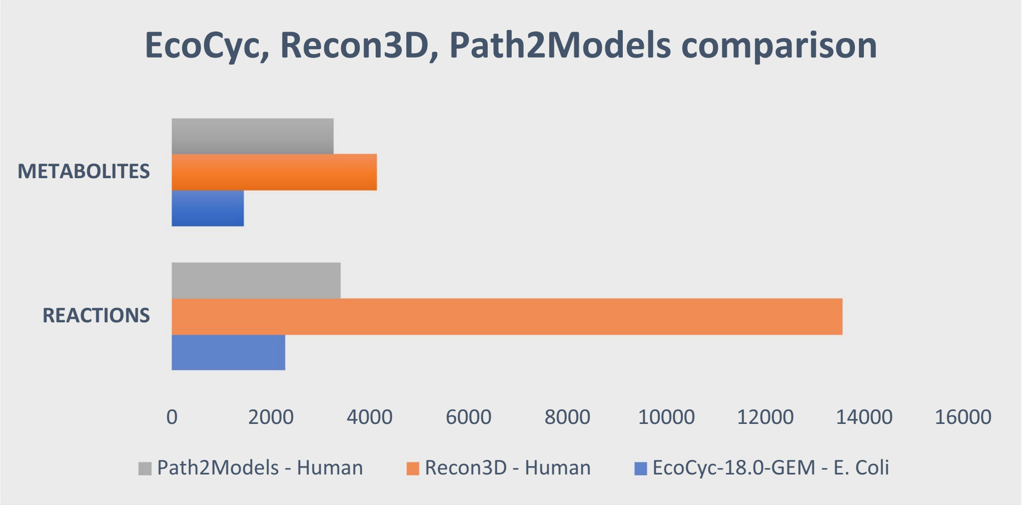


Fig. 2 A comparison of three genome scale models. Recon3D was by far the most data-rich SBML model encountered, as evidenced also on the BiGG model website (Systems_Biology_Research_Group. BiGG Models 2023)

genomes having used 3000 CPUs operating over months. GAMA primarily knocked out genes less likely to disrupt division, followed by random knockouts and recombination, predicting a 360 gene minimal genome. The in silico cell could grow/divide in a simulated SP4 growth medium. Reduced Gene Ontology category terms from UniProt permitting continuity included DNA repair/replication/topology, transcription, regulation, the cell cycle/division, protein transport/folding, lipid production and RNA processing. BLAST (sequence alignment) was used to compare JCVI-Syn3.0 to the GAMA_237 and Minesweeper_256 models. The whole cell model of Mycoplasma genitalium (Karr and Brandon;. 2015) could be run through SimulationRunner.m or MGGRunner.m via MatLab.

4.4 Biochemical pathway/network model generation and optimization

Chemical Reaction Networks (CRNs) were considered critical for modelling in both Synthetic and Systems Biology (Poole et al. 2022), with ongoing efforts to automate the process, with tools created for synthetic network generation (Riva et al. 2022). Despite the successes of constraint-based (flux balance) approaches, explicit concentration-based modelling requires kinetics data (Rosmalen et al. 2021). For kinetic networks, rate laws must be defined (Dräger et al. 2015). A model might be outlined and subsequently parameterized (Poole et al. 2022), perhaps with estimates. Tools capable of defining rate laws included COPASI, CellDesigner and SABIO-RK (Dräger et al. 2015). Specialist tools existed, such as Odefy, which could generate differential Hill-type equations from Boolean networks. Various methods for "model reduction" existed (Rosmalen et al. 2021). Model reduction software included FastCore, NetworkReducer and minNW. Other approaches included MOMA for reduction, which was proposed in relation to next generation constraint-based modelling using GECKO, REMI, MOMENT or RBA. SMGen, with GUI, generated reaction networks with CPU parallelization (Riva et al. 2022). SMGen had SBML and BioSimWare export; where BioSimWare was used by some GPU simulators. There was no evidence that SMGen pursued biological reality beyond arbitrary constraint-generated CRNs. Models for SMGen were defined through stoichiometry and kinetic rate constants and utilized the law of mass-action.

BioCRNpyler, written in Python and programmatically scripted (Poole et al. 2022), was designed to generate SBML format CRNs with combinatorial capacity. The simulator of choice was Bioscrape. BioCRNpyler could combine modular components (essentially SB parts and devices) into large models. Alternatives to BioCRNpyler include BioNetGen, PySB, Tellurium, Virtual Parts Repository (VPR), iBioSim, COPASI and MATLAB Simbiology. Models could be constructed from species and reactions, and could take on a variety of "propensity functions" such as mass-action, Hill and user specified functions. Mechanisms included binding, cooperative binding, catalysis, Michaelis Menten, transcription, translation, dilution, degradation (nuclease/protease), activation (Hill function) and repression (negative Hill function).

SBMLsqueezer 2, also a CellDesigner plugin, made use of the SABIO-RK database via RESTful API to generate large-scale biochemical kinetics models (Dräger et al. 2015), with selectable gene-regulatory rate law alternatives including Hill-Hinze, Hill-Radde, Weaver's equation, S-systems, H-systems etc. Hill function kinetics can provide switch-like behaviour, suitable for transcription factor dynamics, and transcription is a non-linear reaction with power-law approximations connected to Taylor's theorem (Chakraborty et al. 2022). SBMLsqueezer 2 would manipulate SBML via JSBML with libSBML support (Dräger et al. 2015). Reaction type was determined by Systems Biology Ontology and MIRIAM annotations. A pipeline was suggested using a BiGG database model, or generated by KEGGtranslator, with SBMLsqueezer 2 providing kinetic law generation, and SBMLsimulator was suggested for fitting models to experimental data. For the Path2Models project, a pipeline was developed for the generation of computational biochemical pathway models in SBML from KEGG, MetaCyc and BioPAX (Büchel et al. 2013). Upon conversion to SBML, the models would have kinetic rate equations (via SBMLsqueezer) and flux bounds added. KEGG metabolic pathways are described via "processes", downloadable as KEGG Markup Language (KGML), allowing for "process-based" reconstructions, translatable to SBML via KEGGtranslator. Only 0.22% of reactions could utilize SABIO-RK, although as much as 12.2% for Homo sapiens. Path2Models only considered the simplest form of rate law for reversible reactions. Genome-scale metabolic models were generated from KEGG, primarily, and MetaCyc via libAnnotationSBML and SuBliMinal Toolbox software (RAVEN Toolbox and KEGGtranslator are alternatives). Models were specified minimal growth media. Errors were generated in terms of AA essentiality in Path2Models and it incorrectly generalized biochemical constituents for certain lifeforms. The SKiMpy Python package was recently noted for "semi-automated" kinetic model generation (Lent et al. 2023).

The conversion of SBOL to SBML has potential for automating the generation of behavioural simulations from genetic designs; an unrealized aspiration of GenoCAD (Czar et al. 2009). It was suggested that the automation of model construction on the basis of design repositories had not been achieved (Misirli, G.k,, et al. 2019), perhaps the most promising options being the VPR and SB suites such as iBioSim. The VPR was said to contain SBOL designs with corresponding SBML models (Poole et al. 2022), with sufficient metadata for automation (Misirli, G.k,, et al. 2019). An example workflow generated SBOL using Cello, with import into iBiosim for conversion to SBML (Appleton et al. 2017) and simulation via COPASI. The reverse is SBOL generation from CRNs, as performed by MoSec, a sequence generation program (Misirli et al. 2011). MoSec generated EMBL/GenBank and SBOL formatted DNA sequences from SBML or CellML models. The SBML and CellML files would require Standard Virtual Parts and MIRIAM-compliance.

Retrosynthesis can optimize and complete gaps in biochemical pathways, a tool of interest being SciFinder-N (American_Chemical_Society. 2023). Brute-force chemical pathway optimization is computationally demanding, and multithreaded RetSynth was developed to address this (Whitmore et al. 2019). RetSynth could perform FBA for product yield optimization via CobraPy and visualize the pathways. RetSynth could compile information from metabolic databases including PATRIC, KBase, MetaCyc, KEGG, MINE, the ATLAS of Biochemistry and SPRESI.

4.5 Synthetic biology suites

A "Synthetic Biology Suite" is a platform designed to house Synthetic Biology CAD requirements under a single roof. Usually the emphasis is bioregulatory genetic construct design and simulation. Figure 3 presents an overview of such technologies.

Infobiotics Workbench (IBW) is an open source SB suite. IBW integrated various binaries, such as model checkers and Gillespie algorithms, and was designed to be an effective modelling, simulation, verification and sequence generation (via ATGC) tool, with its own ontology-inspired programming language (IBL) for biological circuit design (Konur et al. 2021). IBW ran Gillespie simulations through NGSS and integrated SSA Predictor, an ML solution for identifying the optimal Gillespie algorithm for a model network topology. In practice SSA Predictor presented with inaccuracies (Matzko et al. 2023). A GPU parallelized CUDA Gillespie stochastic simulation algorithm was under development for IBW (Konur et al. 2021), although its status remained uncertain. Formal verification could check models for time course simulation conditions such as molecular quantity thresholds. IBW could automatically add terminators, RBSs via Salis' RBS calculator and spacers. Synthetic Biology genetic part sequences could be determined from the iGem repository or a local database created from Biofab and Rebase. User defined directives could guide ATGC to manage restriction sites. Case studies have used genetic regulatory networks (circuits) with molecular switches to dynamically regulate expression levels; e.g. GFP expression regulation via XOR gate constructed from genetic parts (Konur, et al. 2014). In previous iterations, IBW was intended for the design, analysis and optimization of multicellular systems (Blakes et al. 2014). Decomposition/decoupling of reaction networks could have allowed for tractable and modular optimization. Our ongoing research continued to investigate the spatiotemporal extension of the NGSS component of IBW (Matzko et al. 2023).

iBioSim modelled biochemical systems through in silico genetic circuit design, with optional multicellular grid representations. Operons could be designed in vSBOL (Visual SBOL) and an online registry could be communicated with to select parts. SBOL designs use an embedded part sequencer, SBOLDesigner (Watanabe et al. 2019). iBioSim could import and export in SBML, SBOL, Labelled Petri Net models (LPN) and SED-ML (Myers 2015). Analysis of models used deterministic ODEs, Monte Carlo, Markov Chain and FBA. A similar software, Tinkercell (TinkerCell_Website. TinkerCell. 2022), was created for the product

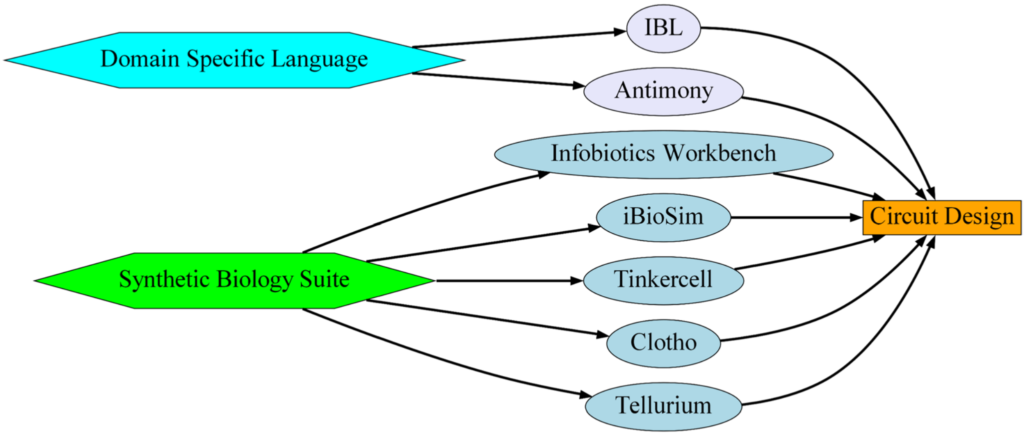


Fig. 3 A summary of Synthetic Biology Suites and Domain Specific Languages (DSLs) discussed in this section. DSLs can exist within Synthetic Biology Suites and are used in the design of bioregulatory circuits

design and analysis cycle. Plug-ins could allow for stochastic simulations, directed evolution, DNA optimization, online searches and experimental data import. Tinkercell used deterministic and tau-leaping stochastic simulations and possessed automated or manual rate equation assignments for designed constructs (Chandran et al. 2010). C, Python and Octave languages could be used for scripting. Tinkercell had text-based modelling via the Antimony language (Smith et al. 2009) and allowed for the drag and drop design of operons, including into plasmid representations. Another suite of tools, Clotho, was developed for iGEM (Internationally Genetically Engineered Machine) competitions (Xia, et al. 2011). Various Clotho apps could be used to operate on metadata objects. An interesting feature was provisional risk assessments based on NIH Guidelines, flagging Parts, Vectors and features using BLAST against virulence factors.

Tellurium, applied through Jupyter Notebook or Spyder IDE, was created for Systems Biology and SB modelling, simulation and analysis (Choi et al. 2018). It used phraSED-ML and SimpleSBML for model design and the Antimony language for translation to and from SBML. Tellurium utilized libRoadRunner for deterministic and stochastic simulations, assessing parameter changes by metabolic control analysis. Network structural analysis used libStructural and Tellurium utilized AUTO2000 for bifurcation analysis, allowing for the assessment of parametric changes, bi-stability and oscillations. Tellurium could parameter estimate by model fitting to experimental data and used a “differential evolution optimizer” from SciPy for parameterization via global optimization. Known data was contrasted to predicted via normalized root mean squared error.

5 Design automation and combinatorial approaches in synthetic biology

Previously, we mentioned combinatorial possibilities in CRN generation (Poole et al. 2022). Rational, semi-rational and combinatorial approaches to pathway design are possible (Appleton et al. 2017), with the potential to utilize genetic parts in combinatorial experiments, even with population level consequences. The power of combinatorial approaches to solve otherwise intractable problems likely overrepresented them in industry compared to rational approaches overrepresented within academia. Rational designs (Stephanopoulos 2012) can be given a combinatorial treatment to select for mutants with best performance by high-throughput, and high-throughput has been suggested for part characterization (Buecherl and Myers 2022). Genetic design automation (GDA) was described as involving part selection, combinatorial methods, assembly and analysis; with emphasis on standards and design portability of well-established parts.

Figure 4 depicts an approximated schematic for the DBTL loop for SB. In this case ML is proposed as a modality through which learning can be automatically administered to combinatorial design, however ML feedback might alternatively interact with other stages of the cycle, calibrating the automated system towards an idealized state. The test metrics would depend on the specific requirements of the product, and can be generalized as assays or micrographics. Assays may include sequencing (e.g. RNA-seq, riboseq (Foo et al. 2023)), flow cytometry, mass spectrometry, transcriptomics, metabolomics and proteomics to extract characterizations of the generated cells or cell populations. Metabolite concentration data can be considered for modelling (Gurdo et al. 2023). Microarrays might be used, as well as various forms of chromatography and DNA assays (e.g. agarose gel electrophoresis). Automated liquid handling with photometric screening was reported (Hellecke et al. 2023). Micrographic analysis is an alternative, although a variety of other testing options might be available, including the use of magnetic resonance (NMR, even MRI) and X-ray crystallography to characterize the synthetic system being generated. Imaging, such as micrographs, might take various forms, for example including whole organism behavioural studies/phenomics (Rosenhahn et al. 2022) or microbial phenomics such as growth rate and sporulation in yeast (Foo et al. 2023). Often behavioural characteristics such as growth are used as objectives functions in modelling (Motamedian et al. 2017; Dukovski et al. 2021). Electron microscopy and serial sectioning can be combined (Larsen et al. 2021) to produce digital reconstructions for analysis (Liimatainen et al. 2021), with implications in 3D culture engineering, such as tissue engineering. For instance, AutoCUTS-LM (Automatic Collector of Ultrathin Sections for Light Microscopy) possessed an ultramicrotome with collection of sections by tape at a rate of 800 per hour, coupled with scanning electron microscopy (Larsen et al. 2021). Electron microscopy was reportedly capable of resolving biological neural networks, and neuron centroid detection utilized the machine learned solution UNetDense.

Semiconductors have been designed through Electronic Design Automation (EDA) for decades (Densmore and Bhatia 2013). Biological Design Automation (BDA) was proposed to involve protocols relayed to microfluidics, liquid handling robots and bioprinters. This could be coupled with ML and an iterative design process. Microfluidic systems could provide for regulated environments for experimentation, with a parallel drawn with EDA “frequency response analysis” (Lux et al. 2011). In terms of the automated genetic design phase of a DBTL cycle (Fig. 4), Cello, GEC, BioCompiler and GenoCAD were singled out, however a manually curated library of devices is a large part of Cello’s success (Beal and Rogers 2020). In assessing the capacity of available resources, design and test were ascribed to the

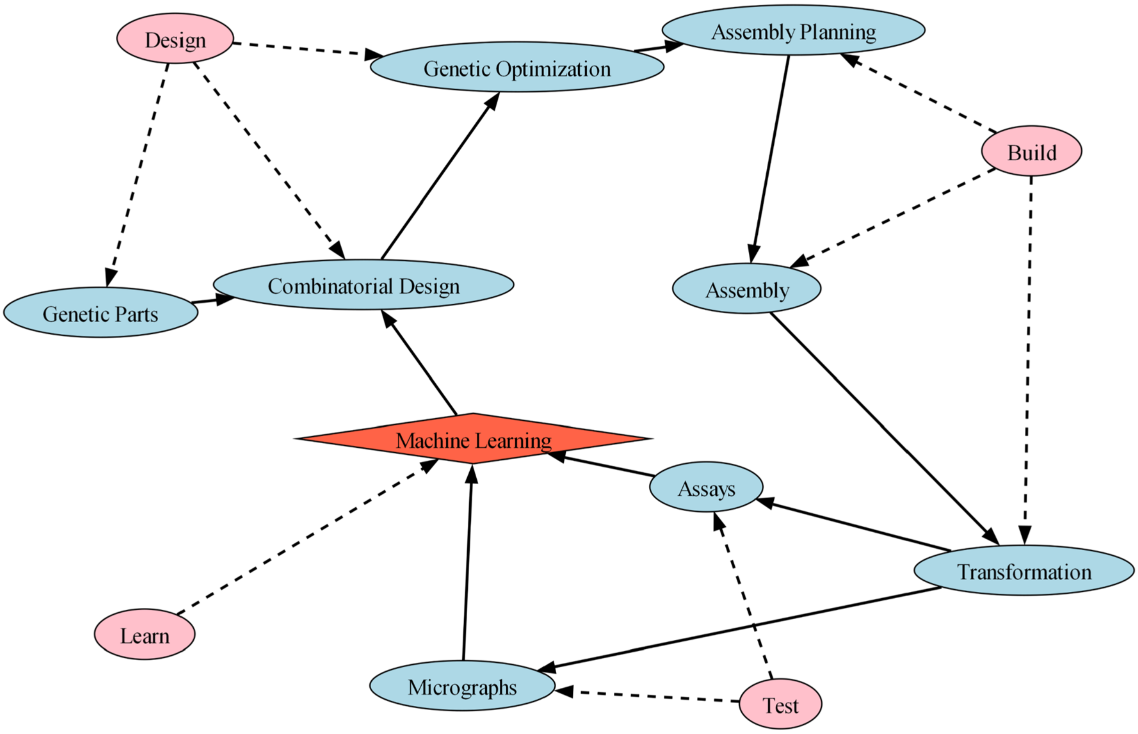


Fig. 4 A general schematic of the DBTL cycle

successes of Autoprotocol, Aquarium, Antha and OpenTrons API. Automated analytics was attributed to automated flow cytometry analysis (TASBE), other assays (Galaxy) and microscopy (SuperSegger and Fogbank).

It is worth noting that while mechanistic models have design implications, another perspective is that the modelling phase resides in the learn stage of the DBTL loop (Gurdo et al. 2023). Whilst modelling is the modality through which design is achieved, this perspective defines the learn phase as the interpretation of collected test phase data into modelling modalities.

5.1 Machine learning for synthetic biology CAD

ML (Fig. 5) can find solutions beyond human intuition (Fawzi et al. 2022). Artificial neural networks are layers of interconnected nodes operating through weighted functions (Rampasek and Goldenberg 2016). Such technology has been applied to biological research including protein folding, molecular biology, neuroimaging-based diagnosis, impact of point mutations and nucleic acid interactions. However, many biological problems have low sample sizes, which is not conducive to deep learning, although data may be manipulable to increase trainability. Thus, pre-existing data is essential, for example AlphaFold exploited motifs and evolutionary information for protein structure inference (Callaway 2022) using the data rich PDB (Varadi et al. 2022). For the design of riboswitches, the Rfam database was used (Palaniappan 2022). Perhaps kinetics data (SABIO-RK) presents as a potential target (Dräger et al. 2015). Other repositories, including metadata from the VPR (Misirli, G.k,, et al. 2019), may present with potential. Our research trajectory would lead us towards considering multi-omics (Matzko 2023). Regarding available ML frameworks, TensorFlow is an open-source example from Google (Rampasek and Goldenberg 2016) and provisioned free access to remote CPU, TPU and GPU computing via Google Colab. TensorFlow's technical complexity was simplified by high level wrappers like Keras and Pretty Tensor. Alternative deep learning frameworks include Torch7, Theano, Caffe, Neon by Nervana, Deeplearning4J and H2O-3. pyTorch Python library has proven to be convenient to use through an IDE (integrated development environment) such as Visual Studio Code on Windows. Although as noted, Google Colab provisions for remote computing, useful particularly if one is operating on limited local hardware.

Protein structure has significance to pathological states, e.g. leukodystrophy (Akdel, et al. 2022), and the structure–function relationship is a well known principle in biological study. Structural and functional predictions can be made from AA sequence motifs (Torres and Fuente-Nunez 2019), which is beneficial to protein design and docking (e.g. via Rosetta 3 (Huang et al. 2016)) and useful for in silico drug design. Docking software can evaluate the

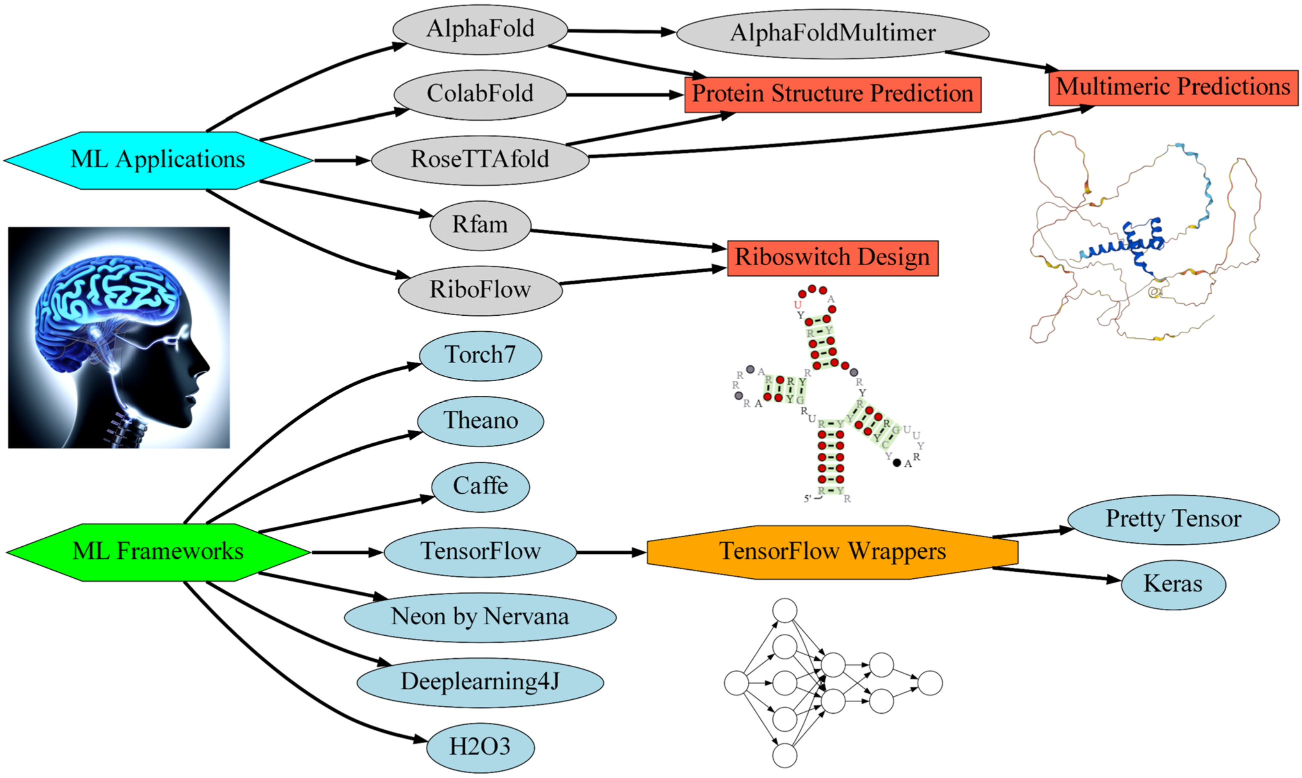


Fig. 5 Synthetic Biology applicable machine learning frameworks and some applications encountered in this section. The upper right illustrates a structure prediction from the AlphaFold Protein Structure Database (EMBL-EBI. 2023), whilst the Riboswitch structure (centre) was acquired from the Rfam database (Elixir. Rfam 2024). The lower graphic shows a very simple illustration of a multilayer perceptron artificial neural network. The above diagram is far from exhaustive and ML Frameworks, architectures and libraries continue to evolve, including architectures such as Transformers (Brown, et al. 2020) and Diffusers. The reader can be encouraged to seek documentation within an architecture of their choosing should this domain be of interest to them

ligand potential of billions of small molecules for drug development (Callaway 2022). However, small structural differences between experiment and prediction can have a significant impact on drug matches. Protein folding predictions had been made via structural homologs or physics/energetics (Brunk et al. 2018; David et al. 2022). Such predictions involved the rearrangement of an AA sequence into a favourable “low-energy state”, considered to be an intractable problem (Perrakis and Sixma 2021). However, AlphaFold made no consideration for energy minima, rather applying ML to homolog templates and multiple sequence alignment (David et al. 2022) via neural networks (Callaway 2022) upon half a century of experimental data (Perrakis and Sixma 2021). AlphaFold could predict dynamic domain behaviours, although interactions were not available in its database. RoseTTAfold and AlphaFold-Multimer were able to achieve limited multimeric predictions. ColabFold allowed the submission of an AA sequence for structure prediction (Callaway 2022). AlphaFold data could be accessed via API, which was used by archives such as UniProt to display protein structures, which also contains X-Ray determined structures from the PDB (Varadi et al. 2022), including Nobel Prize winning structural elucidations upon which AlphaFold was trained. AlphaFold can have serious structural flaws when compared to X-Ray results (Varadi et al. 2022; David et al. 2022; Thornton et al. 2021). Since the Therapeutic Target Database had only a few thousand targets compared to the tens of thousands of human proteins, new virtual screening tools for therapeutic targets might arise from AlphaFold (Tong et al. 2021). AlphaFold reportedly led to drastic improvements in identifying disorders (Callaway 2022). It can be speculated that hybrid ML and classical physical algorithms might be developed, where computationally expensive physical predictions could be used sparingly where necessary if proven to enhance model performance.

Elsewhere, Deep Learning via Python was applied to Riboswitches (Palaniappan 2022) for their classification in a project called RiboFlow, including the use of convolutional neural networks (CNNs) and bidirectional recurrent neural networks with “Long Short-Term memory” (RNNs) derived from TensorFlow (Premkumar et al. 2020). Each of the 32 to 39 riboswitch classes was regulated by a particular ligand, for example glutamine, fluoride, cobalamin etc. The Rfam database for non-coding RNAs was used to obtain FASTA sequences via File Transfer Protocol. “Feature vectors”, essentially an array of encoded data points, were obtained and normalized for ML, including mononucleotide and dinucleotide frequencies. The research presented the potential for riboswitch discovery, with class membership probabilities implying aptamer strength. Such work could be applied to riboswitch targeting drugs, such as antibiotics.

Elsewhere still, the CAD design of purpose-built living multicellular organoids was pursued (Kriegman et al. 2020), with implementations via microsurgical approximations. Evolutionary models were deemed favourable over learning methods due to the flexibility conferred to desired behaviour, however artificial neural networks were suggested for narrowing the design space. Simulations were re-constrained according to observed physical behaviours, thus tying together multiple ML methods, Synthetic Biology, surgical methods and spatiotemporal physical simulations. Physics informed neural networks might be considered for dynamic simulations of such a nature (Gurdo et al. 2023).

Whilst it is prudent to target and validate against big biological data as in above examples, computational scenarios featuring somewhat abstract kinetic enzyme pathways have been probed with ML strategies with optimization towards maximizing fluxes through specific reactions (Lent et al. 2023). The ML models would hence be able to probe the entire design space to select for the desired criteria. However, that work presented with abstractions with author acknowledged assumptions. Hence, laboratory automation, discussed next, could accelerate the process of data collection whilst generating inferable real world data for supervised learning where it is not already available. Real world biological models must be considered the gold standard, however, with high-throughput data acquisition a scenario of diminishing returns might be envisaged between the benefits of biological combinatorial experiments versus computational prediction models.

5.2 Automated laboratories and enabling organizations

DNA Assembly methods have been automated using the OT-2 (Fig. 6) liquid handling robot by OpenTrons, along with external thermocycler (Storch et al. 2020) for DNA amplification via PCR. The OT-2 system came with a python-based API for the manipulation of protocols. The combination of the OpenTrons system and BASIC assembly method was termed DNA-BOT. OpenTrons was a laboratory automation provider, and there was potential to use foundries and automated laboratories such as Strateos (Buecherl and Myers 2022) (Fig. 7). Another company, Synthace (Synthace. Synthace website. 2022), promoted DOE (design of experiments) visual scripting, translated into machine instructions using liquid handlers, dispensers and analytical devices with high-throughput. DOE can be highly parametric, which Synthace referred to as “High Dimensional Experimentation” (Miles and Lee 2018).

Standardized methods with automated laboratories run on software-prepared protocols can address experimental reproducibility issues (Miles and Lee 2018). Sensors were used for precise experimental parameterization with programmatic robotic cloud laboratories with remote access. The

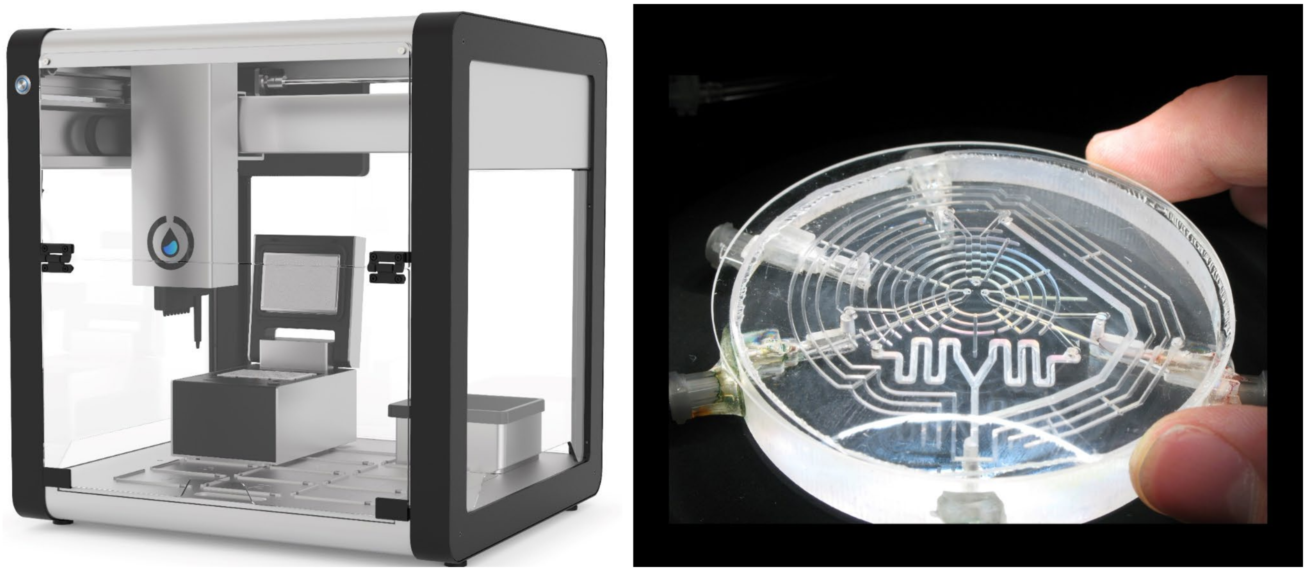


Fig. 6 The OpenTrons OT-2 (left) as compared to a microfluidic palette (NIST. 'Microfluidic Palette' 2009) (right). For efficiency reasons, microfluidics has been considered the future of biotechnology

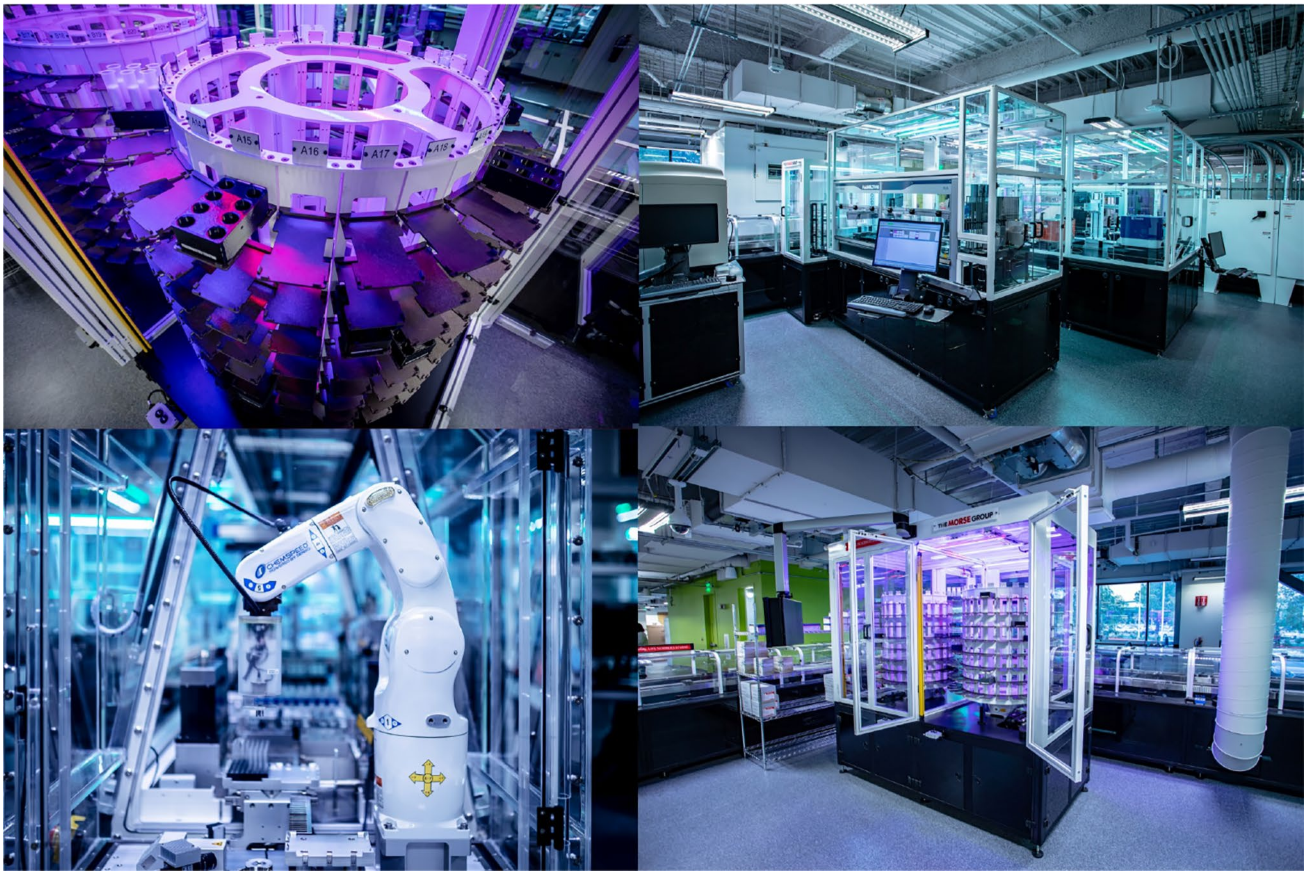


Fig. 7 Views of a Strateos laboratory demonstrating robotic automation

"Transcriptic Common Lab Environment" (TCLE) featured web-interface trackable assays controlled by a scheduler running experiments via robotics that operated via Intel Nus, miniature PCs, operating with precision liquid handling, plate management, centrifugal evacuation of plates, media switching, self-decontamination, absorbance and fluorescence validation, reagent injection, temperature control and PCR. "Autoprotocol" was developed for preparing human and computer reproducible protocols. Having already mentioned microsurgical techniques (Kriegman et al. 2020), it can be speculated that it might even be possible to include microsurgical automation protocols in certain cases. Fog or edge computing for decentralized, heterogenous systems could be considered to localize processing where appropriate, with benefits for distributed computing and latency/bandwidth reduction (Torabi et al. 2022). Strategies in this domain consider data replica placement throughout the distributed system. Given that the above automation relates to the "Internet Of Things" (IoT), such architectures may take into consideration intelligent resource scaling of such distributed systems (Etemadi et al. 2021).

While liquid-handling robotics can hasten research via high-throughput, they occupy a large amount of space, and can be expensive and wasteful (Linshiz et al. 2016). Small volume laboratory experimentation was considered the future of biotechnology. A microfluidics platform utilizing electronically controlled pneumatically actuated microvalves allowed precision fluidic control at 150nL, including mixing, routing and automatic rinsing. PR-PR was software, with GUI, for instruction generation in robotic and microfluidic devices (Oberortner et al. 2017), providing high level programming processed by LabView for solenoid microvalve control (Linshiz et al. 2016).

Biofoundries were reported as high-tech organizations for genetic reprogramming (Hillson et al. 2019). Biofoundries provided and promoted high-throughput, automated systems, CAD, ML, training, logistics, infrastructure, expertise, sustainability and standardization. The Regenerative Medicine Manufacturing Society promoted cell manufacturing for cell therapies, 3D bioprinting, bioreactors, cell counting/sorting, biofabrication of tissues/organs, AI (artificial intelligence) automation, cell harvesting, materials transport, training and supplying laboratories (Hunsberger et al. 2020). ASTM International worked towards standardizing bioinks for bioprinting, such as for drug delivery systems, tissue scaffolds, prosthetics, organoids and tissue/organ products.

Biofoundries are reported to utilize the DBTL cycle to generate thousands of microbial strain variants through parallelized strategies, with screening in microbioreactors (Helleckes et al. 2023). Investigations aimed to resolve the automation of cryopreserved samples from an automatic deep-freezer for use with downstream BioLector microbioreactors and a Tecan Freedom EVO robotics platform. The robotic setup would include a robotic manipulator arm, microplate reader, centrifuge and microtiter plate handling. The process would include disinfection, preculture thawing and optical density triggered genetic expression induction of cultures via IPTG (Isopropyl β-D-1-thiogalactopyranoside). As a result of the phenotyping assays (in this case spectrophotometric), the generation of larger datasets was deemed to have shifted the "bottleneck" of the DBTL cycle towards the learn phase.

The following involves a non-exhaustive detailing of cutting edge technologies, hardware and services encountered at The Festival of Genomics & Biodata in London 2024. Hardware included a Tecan single cell dispenser (Tecan_Trading_AG. Uno 2024), DNA fragmentation via Megaruptor 3 allowing for subsequent long-read sequencing via technologies by PacBio and Oxford Nanopore sequencers (Diagenode. 2024), as well as chromatin and DNA shearing via Diagenode's Bioruptor (Diagenode. Shearing technologies Bioruptor. 2024). Such companies offered a range of services, for example Diagenode offered ATAC-seq (Assay for Transposase-Accessible Chromatin) to analyse chromatin accessibility and ChIP-seq (Chromatin Immunoprecipitation Sequencing) to assess protein-DNA interactions. They also offered a DNA-methylation profiling range, as well as total RNA-seq and mRNA-seq. Also on display was the Promega Maxwell® Benchtop Automated DNA/RNA extractor (Promega_UK. 2024) for simplifying the purification of nucleic acids for downstream Next Generation Sequencing (NGS) and qPCR. NGS hardware included Illumina platforms (Illumina_Inc. 2024) and PromethION platform from Oxford Nanopore. The PromethION 24/48 (Oxford_Nanopore_Technologies_plc. 2024) offered a staggering 4 NVIDIA onboard GPUs, 512 GB RAM and 60 TB of storage. The single cell gene expression kit by Scale Biosciences (SCALEBIO. SINGLE CELL RNA SEQUENCING KIT. 2024) offered multiplexing, i.e. multiple cell high throughput, involving cell barcoding. Unchained Labs provisioned services for viral vector and lipid nanoparticle delivery, including hardware for lipid nanoparticle quality (Unchained_Labs 2024). Vendors also offered reagents for cell disassociation from tissue samples. Not noted at the festival, although possibly represented, would be cell sorting devices, such as via Flow Cytometry and Fluorescence-Activated Cell Sorting. Digital PCR, a more quantitative alternative to standard polymerase chain reaction, was also represented. It is easy to envision how such technologies can be linked together, including phenotypic profiling, for modern Synthetic Biology research and development, and the festival saw research representing leading organizations. For instance, sequenced data can be compared to one or more reference genomes or expression profiles.

5.3 Combinatorial construct design languages

While "forward-engineering" was considered viable for the future, combinatorial optimization (Fig. 8) was said to have great utility in SB (Naseri and Koffas 2020). For example, Proto Biocompiler could select parts and optimize circuit design based on specifications (Myers et al. 2017) as a language for genetic regulatory network generation (Beal et al. 2011). Such technologies can be coupled to other automation categories, notably assembly design. For example, JBEI developed Device Editor for combinatorial part-based DNA constructs with visualization through VectorEditor, while using J5 for automated DNA assembly design (Myers et al. 2017). As GDA was being pursued, design rules and standardization were being promoted, with cloning the focus of software development rather than function design (Lux et al. 2011), which would need to be addressed. BDA and GDA could utilize DSLs not dissimilar to the "Hardware

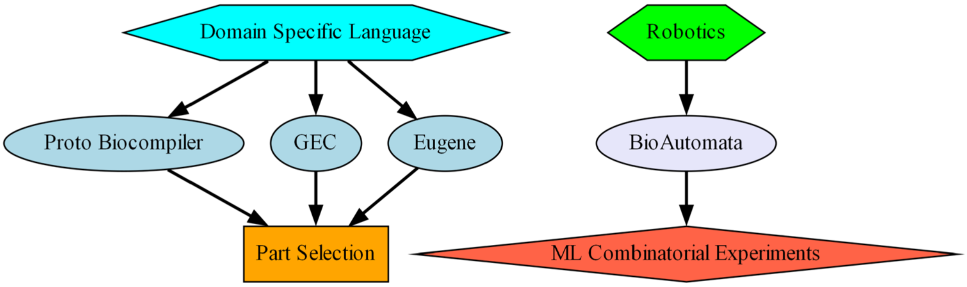


Fig. 8 Technologies related to combinatorial construct design languages and laboratory implementations of combinatorial approaches as discussed in this section. Domain specific languages, e.g. GEC, can be used for genetic part selection. The robotics solution BioAutomata was applied to ML combinatorial automation

Description Languages" of EDA (Bilitchenko et al. 2011; Konur et al. 2021; Smith et al. 2009; Pedersen and Phillips 2009).

GEC was a formal language, with interface implementations, designed for simulation and modelling cycles to select for idealized SB genetic constructs (Pedersen and Phillips 2009) for combinatorial part automation (Pedersen and Andrew;. GEC Manual. 2016) using constraint-based programmatic syntaxes at the part level. Multiple compilations could result, allowing for rapid generation of operon variants (Pedersen and Phillips 2009). Selection capabilities were limited by the lack of well described parts registries containing detailed molecular properties. With Visual GEC discontinued by Microsoft, Lattice Automation and Asimov were approaching the industry with custom tailored software designs (Buecherl and Myers 2022). Similarly, Eugene was a human-readable "ecosystem" of languages for SB, inspired by EDA netlists of connected components (Bilitchenko et al. 2011).

A laboratory combinatorial implementation involved the iBioFAB automated robotics platform integrated with ML and Spearmint source code (HamediRad et al. 2019), with the resulting platform named BioAutomata. Golden Gate assembly was performed by iBioFAB with the iScheduler software. Lycopene production (HamediRad et al. 2019; Exley et al. 2019) would be the output variable, whilst inputs would be via part selection. A T7 promoter region was mutated for strength, generating 12 promoters, and an RBS calculator was used to generate two RBSs of different strengths. The combination of promoters and RBSs yielded 24 unique expression levels, judged via eGFP fluorescence bound to the three expressed genes in the pathway. This project hence demonstrated the potential for ML to predict expression levels, i.e. phenotypic behaviour, from parts selection. Ultimately such design processes benefit from quantifiable dependent outputs relative to input independent variables, where the input variables of the experimental system can be given combinatorial treatment, and outputs can be of varying dimensionality, although in the above case would represent a univariate expression output.

5.4 Circuit design

Circuit design was encountered in relation to Synthetic Biology Suites (Sect. 4.5), and refers primarily to relatively small networks of interactions brought about by small synthetic genetic constructs, unlike genome scale reconstructions. Circuit design is also closely related to the aforementioned "Combinatorial Construct Design Languages", as genetic constructs possess regulatory characteristics that control the behaviour of bioregulatory circuits. This discipline is expanded upon here (Fig. 9).

SB first considered simple genetic circuits before their modular usage (Naseri and Koffas 2020), which would naturally increase the complexity of models. Genetic circuits can include disease marker detection designs, e.g. in lung cancer, and drug delivery (Buecherl and Myers 2022). However, wet-lab testing was still considered necessary since prediction tools had limited accuracy (Naseri and Koffas 2020) and required significant data input from high-throughput experimental transcriptomics, proteomics and metabolomics. For example, Tn-Core could use Tn-seq (transposon insertion sequencing) and RNA-seq data to generate models. Note that

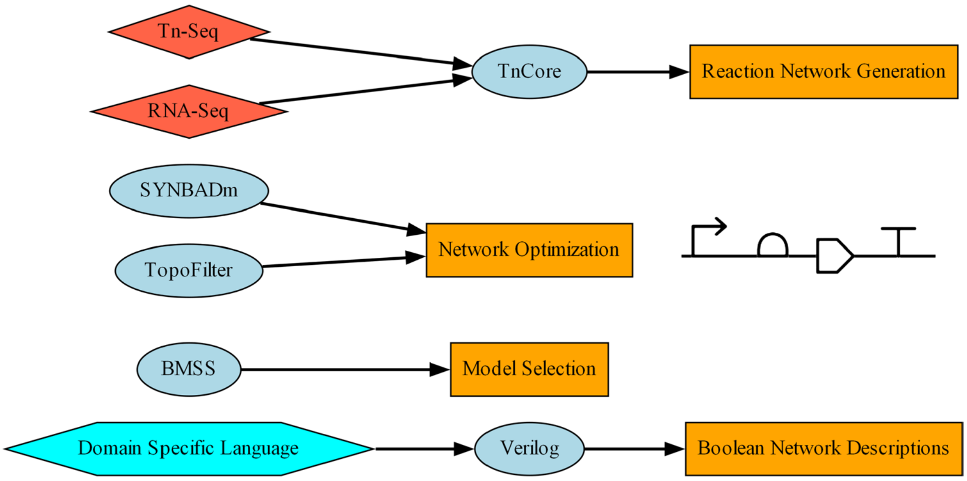


Fig. 9 In silico bioregulatory circuit design related technologies discussed in this section. Categories (orange) include reaction network generation, Boolean network descriptions, model selection and network optimization. Software are in light blue ovals, genetic sequencing data is in tomato coloured diamonds and once again domain specific languages (cyan) have relevance to biological modelling (as also seen in relation to SB suites) this time in the form of Verilog. The illustrative symbols to the right are vSBOL for a typical manually designed circuit involving a promoter, RBS, coding sequence and terminator

Tn-seq can be used to study functional disruptions of genes by transposon introduction.

Logic gates with switching capabilities allow for decision making circuits (Yeoh et al. 2019). Gates can be perceived as nodes in the interactome of a genetic circuit, and potentially controllable in Boolean fashion (Nielsen, et al. 2016). NOT gates can operate via repressors (Cui et al. 2021). AND gates require the presence of multiple signals to allow for expression. OR gates require only the activation of one of multiple pathways. Complex (composite) logic gates include NAND, NOR and XOR. A deoxyribozyme-based circuit of 23 logic gates was reportedly able to play noughts and crosses (Miyamoto et al. 2013). Circuits include logic gates, toggle switches, oscillators (e.g. circadian), repressilators, clocks, French flag, pulse width modulators, memory, counters, decoders, encoders, multiplexers, perceptrons and biosensors (Chakraborty et al. 2022). One model used oscillator-driven DNA tweezers operating alongside an RNA aptamer. An automated biomodel selection platform (BMSS) was created in Python 3 and tested with models containing NOT, AND and OR gates along with inducible and constitutive expression, providing SBOL circuit design and SBML output of the best matched models contrasted to experiment (Yeoh et al. 2019). The BMSS system utilized fluorescence data from microplate readers, along with system perturbation evaluations.

Verilog "Hardware Description Language" was repurposed for genetic circuit design (Nielsen, et al. 2016) and was parsed by Cello into a DNA sequence (Taketani et al. 2020). Genetic circuit generation from Verilog involved the formation of a netlist Boolean gate network description (Jones, et al. 2022). The user constraints file provided restrictions for the selection of part alternatives (Chakraborty et al. 2022), arranged into a DNA sequence according to Eugene language rules (Jones, et al. 2022). Combinatorial construct design algorithms were used for part alternatives or part order (Nielsen, et al. 2016) with subsequent simulation and possible identification of regulatory defects with comparisons made to experimental flow cytometry. The Cello workflow was applied to smart therapeutics (Taketani et al. 2020).

SYNBADm was a Matlab implementation for automated optimization of genetic circuit design (Otero-Muras et al. 2016) utilizing multi-objective optimization for pareto optimality, an approach also mentioned in relation to TopoFilter for 3 enzyme networks (Chakraborty et al. 2022). TopoFilter was considered to have limited scalability due to its brute force approach. SYNBADm supported mass action and Hill kinetics upon construction of biological components/parts, as well as providing time-course simulations (Otero-Muras et al. 2016). This would require libraries of "components" and objective functions based around features such as production costs and circuit behaviours. SYNBADm was scalable to 9 nodes (Chakraborty et al. 2022). It was put forward that bioregulatory networks resemble neural networks, and hence ML has a suitable role to play in relation to them.

5.5 Genetic optimization

Once a genetic construct has been initially designed, it is prudent to consider genetic optimization, not least due to the redundancy of the triplet code for encoding amino acids in codons. Subsequently, the required sequences may be synthesized de novo and/or stitched together through restriction and ligation. Genetic optimization alters the features of a genetic sequence, such as codon optimization and RBS translation initiation rates (Swainston et al. 2018), as well as exotic exercises such as optimizing riboswitches (Wu et al. 2019). Codon optimization may prevent ribosome stalling, ensure correct translation termination, modulate gene expression, prevent growth impairment, prevent frameshifts and prevent the misincorporation of AAs. It allows genes to be recycled between organisms (heterologous expression) (Villalobos et al. 2006; Gaspar et al. 2016).

EuGene (not Eugene language) was a DNA optimization program that exploited online databases for codon usage, context tables and orthologs for sequence alignment (Gaspar et al. 2016). EuGene used data extraction from FASTA and GenBank, combined with homolog searches using BLAST. The PDB and KEGG databases provided EuGene more information on homologs, as well as protein structure and genomic expression levels. EuGene performed alignment using the MUSCLE algorithm. CAI (Codon Adaptation Index) was calculated through highly expressed genes. However, CAI use was advised against (Villalobos et al. 2006). The heterologous gene redesign algorithm used genetic algorithms (slow) or simulated annealing (fast) (Gaspar et al. 2016).

Gene Designer could edit and annotate in silico DNA constructs with functions including the addition of polyhistidine-tags or sequencing primers into a DNA sequence, the identification of restriction sites, and flagging for methylation sensitive restriction enzymes (Villalobos et al. 2006). Gene Designer could search for Open Reading Frames by their start and stop codons; as well as a search capability for RBSs and sequence motifs. It allowed manual codon triplet code manipulations, and could simulate cloning in silico via restriction sites, with cut plasmids selected for ligation considering overhangs. An alternative to CAI involved Codon Usage Tables. Gene Designer's Codon Optimizer used a probabilistic Monte Carlo based algorithm able to find different, but essentially equivalent, outcomes. In-built vector types (Dixon 2023) included an E. coli plasmid (pT7-SNAP), and a mammalian plasmid (pMCPm™).

Available via web application (Berkeley_Lab. BOOST Build 2022), JAR format and REST API, BOOST was a suite of software tools intended for the SB design-build transition (Oberortner et al. 2017), emphasizing automated DNA construct design for vendor synthesis. Consideration could be made regarding GC (strongly hydrogen bonding) content, repeats, secondary structures and restriction sites. BOOST commenced with codon usage optimization via Codon Tables. Violations could undergo "codon juggling" by translation to a polypeptide with codon modification via reverse-translation. "Relaxed Weight" or complete randomization could even out codon usages and reduce excessively used codons. With DNA length a factor for genetic construct assembly success, excessively short sequences were flagged and long sequences partitioned according to success probability. BOOST, for its three tools (Juggler, Polisher, Partitioner), accepted DNA sequences in various formats.

RiboLogic was developed in Python to design Riboswitch sequences (Wu et al. 2019). Input involved ligand-binding aptamer sequences along with estimated dissociation constants and perhaps secondary structures of the activated state. RiboLogic optimized surrounding sequences for ligand binding simulations and utilized simulated annealing optimization with temperature reduction for possible sequences, along with random mutations and scoring mechanisms.

5.6 Automating genetic construct assembly protocols

DNA assembly generates constructs from DNA components/parts, and assembly standardization has been pursued by the SB community (Walsh et al. 2019), despite continued variability. DNA assembly involves vector design, assembly planning and liquid handling (Appleton et al. 2017). Traditionally, such techniques were manual, with restriction and ligation in separate steps. However, high-throughput DNA assembly was sought using assembly planning tools such as DNALD and Raven. Algorithms for joining two DNA fragments per assembly step were developed (Densmore et al. 2010). As DNA assembly evolved, one-pot restriction ligation toolkits were released (Exley et al. 2019). To generate variations of genetic constructs, the assembly of a "goal part" could be sought algorithmically, with each step represented on an "assembly graph" (Densmore et al. 2010), with time and financial costs estimated from resulting graph steps and levels. Algorithms for these purposes were implemented through the Clotho framework.

A liquid-handling platform (Freedom EVO 150) was compared to manual DNA assembly using the MoClo methodology (Walsh et al. 2019) using variations of 5 part constructs. Transformation efficiency was measured in colony forming units (CFU) per volume, as observed by coloration. GenBank files were read by software called Puppeteer to create combinatorial variants with a fixed sequence of part types, and subsequent generation of a DNA assembly plan and protocols for humans and robots. Pipetting commands for a Tecan system were generated more rapidly with Puppeteer than if programmed with EvoWare. Manual versus automated CFU percentage outcomes demonstrated no difference. Thus a single assembly may be more suitable for a human, whilst larger numbers would suit robotics.

J5 was a web-based tool for design automation in scarless DNA assembly (Hillson et al. 2012) across multiple assembly methods. In a case study, GFP was tagged for localization and degradation, with combinatorial design potential. In such experiments, variants could number in the thousands and J5's combinatorial assembly planning could save time. Constraints were applied to parts for combinatorial selection via Eugene-based rules, similarly to tools like Cello (Jones, et al. 2022). J5 could perform BLAST to check for flanking sequence similarity and potential incompatibilities (Hillson et al. 2012). Endonuclease generated overhangs must not combine with the wrong targets, which J5 could manage. As many as 2.4 billion overhang combinations were assessed. J5 performed simulated annealing, and could generate a PCR setup control file for the eXeTek liquid-handling robot, with future intent to apply such methods to the Tecan EvoLab.

DNA Constructor software was used to design DNA combinatorial library construction protocols for a microfluidics platform (Linshiz et al. 2016). J5 and Device Editor were used to construct a combinatorial library. Assembly protocol outputs from DNA Constructor took the form of an"interactive assembly tree" via the DOT language of Graphviz (used for Figs. 3, 4, 5, 8, 9 in this review). Isothermal Hierarchical DNA Construction was automated on a 16 input and output well microfluidic chip. One pot Gibson assembly was used with the pETBlue-1 plasmid expression vector. Automated transformation of the plasmid into E. coli utilized the microfluidic chip, with subsequent plating of the cells. On-chip assays assessed cell growth, protein expression and colorimetry. Hence, combinatorial genetic sequence methods and library construction were combined with assembly protocols for microfluidics assays of transformed cells.

6 Discussion and conclusions

This review elucidated SB automation across the DBTL cycle to inform wet and dry laboratories regarding available technological opportunities. Standards were ubiquitous and provide for numerous benefits and capabilities (Matzko et al. 2023; Myers et al. 2017; Keating et al. 2020; Beal and Rogers 2020). DSLs (Konur et al. 2021; Smith et al. 2009) provide for syntactic translation, human readability, model construction, genetic designs, constraints and combinatorial capabilities (Bilitchenko et al. 2011; Czar et al. 2009; Pedersen and Phillips 2009). Libraries and APIs exist for in silico manipulations (Myers et al. 2017), including web services for data acquisition (Dräger et al. 2015). For design, modelling and ML, data is vital (Rampasek and Goldenberg 2016; Perrakis and Sixma 2021), and resources were outlined to the extent of whole cell modelling (Reactome. Reactome Pathway Browser. 2022; Brunk et al. 2018; Weaver et al. 2014) and minimal genomes (Sleator 2016) via mutagenesis and knockouts (Rees-Garbutt et al. 2020). However, the argument was made that modelling can occur during the test to learn transition (Gurdo et al. 2023). The use of ontologies allowed for functional descriptions (Rees-Garbutt et al. 2020) and cataloguing (Golebiewski et al. 2007), while datamining offers opportunities for data extraction (Büchel et al. 2013; Baltoumas, et al. 2021; Luo, et al. 2022). Kinetics solvers provide for dynamic simulations with consideration for concentrations and perturbations (Matzko et al. 2023; Konur et al. 2021; Choi et al. 2018; Sanassy et al. 2015), which can be analysed in a variety of ways (Konur and Gheorghe 2015; Riva et al. 2022; Hoops et al. 2006), while Boolean models provide a simplification (Karagöz et al. 2021). FBA simulation is suitable for metabolic engineering (Sekiguchi et al. 2021) and does not require kinetic rate parameterization. Parameter estimation is achievable algorithmically through maximal experimental data characterization (Choi et al. 2018; Hoops et al. 2006). Meanwhile, high performance computing speeds up computations (Konur et al. 2021; Riva et al. 2022; Rees-Garbutt et al. 2020) and ML has been used to make SB associated predictions (Rampasek and Goldenberg 2016; HamediRad et al. 2019). Protein structure prediction associated with docking computations has potential in drug design (Callaway 2022; Huang et al. 2016). Genetic optimization allows genes to be used effectively between organisms (Villalobos et al. 2006) and to enhance genetic devices (Wu et al. 2019) with potential for biomedical sensor design (Wang et al. 2016). Automated genetic editing allows for assembly planning (Villalobos et al. 2006) for genetic constructs (Densmore et al. 2010) with combinatorial design potential (Walsh et al. 2019). Databases can be used to generate reaction networks (Büchel et al. 2013; Dräger et al. 2015), and model reduction algorithms exist (Rosmalen et al. 2021). Tissue engineering automation holds promise for multicellular organoid models (Kriegman et al. 2020) and tissue function predictions (Hunsberger et al. 2020). Robotics (Storch et al. 2020) have been available, including from enabling organizations (Buecherl and Myers 2022). However, microfluidics and “Lab-on-a-Chip” (Linshiz et al. 2016) may represent the future alongside ML.

In conclusion, Synthetic Biology is a complex field that artificially recombines and optimizes bioregulatory genetic sequences fit to purpose, with software/DSLs/hardware and data acquisition across its workflow. Data provisions the capacity to design interaction networks for functional elucidation, practical applications, DOE and ML opportunities. Combinatorial approaches and evolutionary methods with high throughput have been industry preferred methods and should not be underestimated. For example, emerging combinatorial strategies based on CRISPR-Cas9 for eukaryotic DBTL, where manual learning took the form of genotype–phenotype mapping using synthetic yeast chromosomes, including defect assessment from behavioural phenomics and Gene Ontology mapping for differential gene expression (Foo et al. 2023). In this case chromosomal design via BioStudio was based on the Sc2.0 project of Saccharomyces cerevisiae, with assembly from chemically synthesized DNA chunks via mitotic and meiotic recombination. Genetic locus-to-locus comparisons could be made between experimental and control strains as a means of manual learning, emphasizing the importance of perturbation and modification of not only model organisms, but for debugging genetic constructs and synthetic chromosomes against a standard. Presumably, a broader challenge may be in replicating such experimental strategies to reflect medical physiological conditions, such as perturbations of histological scenarios for medicine, e.g. cancer mutagenesis.

A range of ML options are available and undoubtedly inbound, which may be explored through frameworks, databases of results, or pretrained models, which could be applied to high-throughput and high dimensional automated Synthetic Biology studies. Indeed, because supervised learning requires prior labelling, a process that is essentially an approximated interpolation, reinforcement learning would be a more fruitful option for directing machines towards objectives with unknown state requirements and for experimental design optimization. Supervised learning would be suited more towards classification predictions based on large amounts of pre-existing data (Perrakis and Sixma 2021). As the amount of data from reinforcement strategies might grow, the larger the dataset for supervised learning, where supervised learning might map the experimental parameter inputs to the outputs, hence constituting a closing of the experimental DBTL loop through model parameterization. A careful evaluation of the human research/development cycle along with objectives and acceleration through ML high-throughput automation might prove worthwhile to minimize trial and error costs. The likely strategy would be a systematic exploration of a constrained parameter space.

A range of test options, including assays, are available within the DBTL loop. These, or pre-existing data, are considered essential for allowing the closing of the loop by transitioning to the learn stage. Deeper, comprehensive analysis of the individual loop phases can be advised. Indeed, such studies exist, for instance emphasizing the criteria in the test phase (Helleckes et al. 2023). The prospect of a community-driven open-source platform could be considered to map DOE and DBTL through the stages of computational design, high-throughput machine automated combinatorial design, and maximally automated analysis of the products. Given the likely utilization of commercial products, such an academic platform might be of interest to industry as a marketing device, and as a possible driver of standardization and competition for efficient, cost-effective, accessible automation.

There is a notable contrast between genome scale reconstructions and the design of, potentially orthogonal, small circuit designs. The latter can be used for orthogonal operations such as biochemical sensor design. However, the more complex a design, the more likely disruptions due to a lack of orthogonality might be. In silico modelling and predictions require considerable work to achieve realistic outcomes compared to in vivo or in vitro models, particularly in terms of spatiotemporal dynamics, a domain of particular interest to us. Thus such modelling involving time-course and dynamic spatial characteristics have CAD implications, likely most suited to hypothesis generation in the short term given the challenges regarding kinetics data (Gurdo et al. 2023) and in our experience the translation between biochemical and physical modelling (Matzko et al. 2023). The effectiveness of such CAD systems depend on the quality of data and the quality of processing operations, which may finally culminate in increasingly accurate digital replicas of Synthetic Biology scenarios through the exploration and expansion of existing software and services, with many benefits ranging from costs, to ethics and logistics.

References

Akdel M et al (2022) A structural biology community assessment of AlphaFold2 applications. Nat Struct Mol Biol 29(11):1056–1067

American_Chemical_Society. CAS SciFinder^{n}. 2023 [cited 2023 24/01/2023]; Available from: https://www.cas.org/solutions/cas-scifinder-discovery-platform/cas-scifinder

Appleton E et al (2017) Design automation in synthetic biology. Cold Spring Harb Perspect Biol 9(4):a023978

libRoadRunner. libRoadRunner. 2022 [cited 2022 16/12/2022]; Available from: https://www.libroadrunner.org/

Baig H et al (2020) Synthetic biology open language (SBOL) version 300. J Integr Bioinf. https://doi.org/10.1515/jib-2020-0017

Baltoumas FA et al (2021) OnTheFly2.0: a text-mining web application for automated biomedical entity recognition, document annotation, network and functional enrichment analysis. NAR Genom Bioinf 3(4):lqab090

Beal J, Rogers M (2020) Levels of autonomy in synthetic biology engineering. Mol Syst Biol 16(12):10019

Beal J, Lu T, Weiss R (2011) Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks. PLoS ONE 6(8):e22490–e22490

Berkeley_Lab (2022) BOOST Build Optimization Software Tools for DNA Synthesis. [cited 2022 20/12/2022]; Available from: https://boost.jgi.doe.gov/

Bilitchenko L et al (2011) Eugene–a domain specific language for specifying and constraining synthetic biological parts, devices, and systems. PLoS ONE 6(4):e18882–e18882

EMBL-EBI. Biomodels Repository. 2022 [cited 2022 27/11/2022]; Available from: https://www.ebi.ac.uk/biomodels/.

Biosciencetoday. Defining the future of experiment design. 2022 [cited 2022 13/12/2022]; Available from: https://www.biosciencetoday.co.uk/synthace-defining-the-future-of-experiment-design/

Blakes J et al (2014) Infobiotics workbench—a P systems based tool for systems and synthetic biology. Springer, Cham

Broad_Institute. Connectivity Map. 2022 [cited 2022 16/12/2022]; Available from: https://clue.io/

Brown, T.B., et al., Language Models are Few-Shot Learners. ArXiv, 2020. **abs/2005.14165**

Brunk E et al (2018) Recon3D enables a three-dimensional view of gene variation in human metabolism. Nat Biotechnol 36(3):272–281

Büchel F et al (2013) Path2Models: large-scale generation of computational models from biochemical pathway maps. BMC Syst Biol 7(1):116–116

Buecherl L, Myers CJ (2022) Engineering genetic circuits: advancements in genetic design automation tools and standards for synthetic biology. Curr Opin Microbiol 68:102155–102155

Cahan P, Treutlein B (2023) A conversation with ChatGPT on the role of computational systems biology in stem cell research. Stem Cell Reports 18(1):1–2

Callaway E (2022) What's next for AlphaFold and the AI protein-folding revolution. Nature (london) 604(7905):234–238

Chakraborty D, Rengaswamy R, Raman K (2022) Designing biological circuits: from principles to applications. ACS Synth Biol 11(4):1377–1388

Chandran D, Bergmann FT, Sauro HM (2010) Computer-aided design of biological circuits using tinkercell. Bioeng Bugs 1(4):276–283

Choi K et al (2018) Tellurium: An extensible python-based modeling environment for systems and synthetic biology. BioSystems 171:74–79

COPASI. COPASI: Biochemical System Simulator. 2022 [cited 2022 16/12/2022]; Available from: http://copasi.org/

Cui S et al (2021) Multilayer genetic circuits for dynamic regulation of metabolic pathways. ACS Synth Biol 10(7):1587–1597

Czar MJ, Cai Y, Peccoud J (2009) Writing DNA with GenoCAD. Nucl Acids Res 37(suppl_2):W40–W47

David A et al (2022) The AlphaFold database of protein structures: a biologist's guide. J Mol Biol 434(2):167336–167336

Delile J et al (2017) A cell-based computational model of early embryogenesis coupling mechanical behaviour and gene regulation. Nat Commun 8(1):13929–13929

Densmore DM, Bhatia S (2013) Bio-design automation: software + biology + robots. Trends Biotechnol (regular Ed) 32(3):111–113

Densmore D et al (2010) Algorithms for automated DNA assembly. Nucleic Acids Res 38(8):2607–2616

Diagenode. Megaruptor® 3. 2024 [cited 2024; Available from: https://www.diagenode.com/en/p/megaruptor-3.

Diagenode. Shearing technologies Bioruptor. 2024 [cited 2024; Available from: https://www.diagenode.com/en/categories/bioruptor-shearing-device

Dixon A (2023) Gene Designer by DNA 2.0 Tutorial. [cited 2023 19/01/2023]

Dotmatics (2023) *Geneious by Dotmatics*. [cited 2023 30/01/2023]; Available from: https://www.geneious.com/

Dotmatics (2022) *Snapgene: The Future of Cloning is Smarter and Faster* [cited 2022 23/12/2022]; Available from: https://www.snapgene.com/

Dräger A et al (2015) SBMLsqueezer 2: context-sensitive creation of kinetic equations in biochemical networks. BMC Syst Biol 9(1):68

Dukovski I et al (2021) A metabolic modeling platform for the computation of microbial ecosystems in time and space (COMETS). Nat Protoc 16(11):5030–5082

EBML_EBI. Expression Atlas. 2022 [cited 2022 23/12/2022]; Available from: https://www.ebi.ac.uk/gxa/home

Elixir. Rfam: The RNA families database. 2024 [cited 2024; Available from: https://rfam.org/

Else H (2023) Abstracts written by ChatGPT fool scientists. Nature (london) 613(7944):423–423

EMBL-EBI. AlphaFold Protein Structure Database. 2023 [cited 2023 19/01/2023]; Available from: https://alphafold.ebi.ac.uk/

Etemadi M, Ghobaei-Arani M, Shahidinejad A (2021) A cost-efficient auto-scaling mechanism for IoT applications in fog computing environment: a deep learning-based approach. Clust Comput 24(4):3277–3292

Exley K et al (2019) Utilising datasheets for the informed automated design and build of a synthetic metabolic pathway. J Biol Eng 13(1):8–8

Fawzi A et al (2022) Discovering faster matrix multiplication algorithms with reinforcement learning. Nature 610(7930):47–53

Foo JL et al (2023) Establishing chromosomal design-build-test-learn through a synthetic chromosome and its combinatorial reconfiguration. Cell Genomics 3(11):100435

Gaspar P et al (2016) EuGene: maximizing synthetic gene design for heterologous expression. Bioinformatics (oxford, England) 32(7):1120–1120

Gillespie M et al (2022) The reactome pathway knowledgebase 2022. Nucleic Acids Res 50(D1):D687–D692

Golebiewski M et al (2007) Integration of SABIO-RK in workbenches for kinetic model design. BMC Syst Biol 1(S1):P4–P4

Gurdo N et al (2023) Automating the design-build-test-learn cycle towards next-generation bacterial cell factories. New Biotechnol 74:1–15

Ham TS et al (2012) Design, implementation and practice of JBEI-ICE: an open source biological part registry platform and tools. Nucleic Acids Res 40(18):e141–e141

HamediRad M et al (2019) Towards a fully automated algorithm driven platform for biosystems design. Nat Commun 10(1):5150–5210

Helleckes LM et al (2023) From frozen cell bank to product assay: high-throughput strain characterisation for autonomous design-build-test-learn cycles. Microb Cell Fact 22(1):130

Hillson NJ et al (2012) j5 DNA assembly design automation software. ACS Synth Biol 1(1):14–21

Hillson N et al (2019) Building a global alliance of biofoundries. Nat Commun 10(1):2040–2040

Hoops S et al (2006) COPASI—a COmplex PAthway SImulator. Bioinformatics 22(24):3067–3074

Huang P-S, Boyken SE, Baker D (2016) The coming of age of de novo protein design. Nature (london) 537(7620):320–327

Human_Protein_Atlas. The Human Protein Atlas. 2022 [cited 2022 23/12/2022]; Available from: https://www.proteinatlas.org/

Hunsberger J et al (2020) Improving patient outcomes with regenerative medicine: how the Regenerative Medicine Manufacturing Society plans to move the needle forward in cell manufacturing, standards, 3D bioprinting, artificial intelligence-enabled automation, education, and training. Stem Cells Transl Med 9(7):728–733

Illumina_Inc. Illumina sequencing platforms. 2024 [cited 2024; Available from: https://www.illumina.com/systems/sequencing-platforms.html

Jones TS et al (2022) Genetic circuit design automation with Cello 2.0. Nat Protocols 17(4):1097–1113

Kahl LJ, Endy D (2013) A survey of enabling technologies in synthetic biology. J Biol Eng 7(1):13–13

Karagöz Z et al (2021) Towards understanding the messengers of extracellular space: computational models of outside-in integrin reaction networks. Comput Struct Biotechnol J 19:303–314

Karr JB., Brandon (2015) Mycoplasma genitalium whole-cell model GitHub. 2015 [cited 2023 19/01/2023]; Available from: https://github.com/CovertLab/WholeCell

Keating SM et al (2020) SBML Level 3: an extensible format for the exchange and reuse of biological models. Mol Syst Biol 16(8):1–21

Konur S, Gheorghe M (2015) A property-driven methodology for formal analysis of synthetic biology systems. IEEE/ACM Trans Comput Biol Bioinf 12(2):360–371

Konur S et al (2014) Conventional verification for unconventional computing: a genetic XOR gate example. Fundam Inf. https://doi.org/10.3233/FI-2014-1093

Konur S et al (2021) Toward full-stack in silico synthetic biology: integrating model specification, simulation, verification, and biological compilation. ACS Synth Biol 10(8):1931–1945

Kriegman S et al (2020) A scalable pipeline for designing reconfigurable organisms. Proc Natl Acad Sci 117(4):1853–1859

Larsen NY et al (2021) Cellular 3D-reconstruction and analysis in the human cerebral cortex using automatic serial sections. Commun Biol 4(1):1030–1030

Letort G et al (2019) PhysiBoSS: a multi-scale agent-based modelling framework integrating physical dimension and cell signalling. Bioinformatics 35(7):1188–1196

Li B et al (2019) NUFEB: a massively parallel simulator for individual-based modelling of microbial communities. PLoS Comput Biol 15(12):e1007125–e1007125

Liimatainen K et al (2021) Virtual reality for 3D histology: multi-scale visualization of organs with interactive feature exploration. BMC Cancer 21(1):1133–1133

Linshiz G et al (2016) End-to-end automated microfluidic platform for synthetic biology: from design to functional analysis. J Biol Eng 10(1):3–3

Luo R et al (2022) BioGPT: generative pre-trained transformer for biomedical text generation and mining. Brief Bioinf 23(6):409

Lux MW et al (2011) Genetic design automation: engineering fantasy or scientific renewal? Trends in Biotechnology (regular Ed) 30(2):120–126

Matzko RO, Mierla L, Konur S (2022) A 3D multicellular simulation layer for the synthetic biology CAD infobiotics workbench suite. In: bioinformatics and biomedical engineering. Springer International Publishing, Cham

Matzko RO, Mierla L, Konur S (2023) Novel ground-up 3D multicellular simulators for synthetic biology CAD integrating stochastic gillespie simulations benchmarked with topologically variable SBML models. Genes 14(1):154

Matzko RO (2023) *BioNexusSentinel*. [cited 2023 23/12/2023]; Available from: https://github.com/RichardMatzko/BioNexusSentinel

McLaughlin JA et al (2018) SynBioHub: a standards-enabled design repository for synthetic biology. ACS Synth Biol 7(2):682–688

Miles B, Lee PL (2018) Achieving Reproducibility and Closed-Loop Automation in Biological Experimentation with an IoT-Enabled Lab of the Future. SLAS Technol 23(5):432–439

Misirli G et al (2011) Model annotation for synthetic biology: automating model to nucleotide sequence conversion. Bioinformatics 27(7):973–979

Misirli GK et al (2019) A computational workflow for the automated generation of models of genetic designs. ACS Synth Biol 8(7):1548–1559

Miyamoto T et al (2013) Synthesizing biomolecule-based boolean logic gates. ACS Synth Biol 2(2):72–82

Motamedian E et al (2017) TRFBA: an algorithm to integrate genome-scale metabolic and transcriptional regulatory networks with incorporation of expression data. Bioinformatics 33(7):1057–1063

Myers, C.J.N.B.S.G.K.J.H.K.C.M.N.N.T.N.T.P.N.R.J.S.L.W. iBioSim Version 2.8 User's Manual. 2015 [cited 2022 18/12/2022]; Available from: https://myersresearchgroup.github.io/ibiosim.github.io/docs/iBioSim.html

Myers CJ et al (2017) A standard-enabled workflow for synthetic biology. Biochem Soc Trans 45(3):793–803

Naseri G, Koffas MAG (2020) Application of combinatorial optimization strategies in synthetic biology. Nat Commun 11(1):2446–2446

Nielsen AAK et al (2016) Genetic circuit design automation. Science (american Association for the Advancement of Science) 352(6281):aac7341–aac7341

NIST. 'Microfluidic Palette' May Paint Clearer Picture of Biological Processes. 2009 03/05/2023]; Available from: https://www.nist.gov/news-events/news/2009/07/microfluidic-palette-may-paint-clearer-picture-biological-processes

Oberortner E et al (2017) Streamlining the design-to-build transition with build-optimization software tools. ACS Synth Biol 6(3):485–496

Orth JD, Thiele I, Palsson BØ (2010) What is flux balance analysis? Nat Biotechnol 28(3):245–248

Otero-Muras I, Henriques D, Banga JR (2016) SYNBADm: a tool for optimization-based automated design of synthetic gene circuits. Bioinformatics (oxford, England) 32(21):3360–3362

Oxford_Nanopore_Technologies_plc. PromethION 24/48. 2024 [cited 2024; Available from: https://nanoporetech.com/products/sequence/promethion-24-48

Palaniappan, A. RiboswitchClassifier. 2022 [cited 2023 26/01/2023]; Available from: https://github.com/RiboswitchClassifier.

Pan M et al (2021) Modular assembly of dynamic models in systems biology. PLoS Comput Biol 17(10):e1009513–e1009513

Pattisapu N, et al. (2020) Medical Concept Normalization by Encoding Target Knowledge, in Proceedings of the Machine Learning for Health NeurIPS Workshop, V.D. Adrian, et al. (ed) 2020, PMLR: Proceedings of Machine Learning Research. p 246–259

Pedersen MP, Andrew. GEC Manual. 2016 [cited 2022 19/12/2022]; Available from: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/gec-manual.pdf

Pedersen M, Phillips A (2009) Towards programming languages for genetic engineering of living cells. J R Soc Interface 6(Suppl 4):S437–S450

Perrakis A, Sixma TK (2021) AI revolutions in biology: the joys and perils of AlphaFold. EMBO Rep 22(11):e54046–e54046

Poole W et al (2022) BioCRNpyler: compiling chemical reaction networks from biomolecular parts in diverse contexts. PLoS Comput Biol 18(4):e1009987–e1009987

Premkumar KAR, Bharanikumar R, Palaniappan A (2020) Riboflow: using deep learning to classify riboswitches with ~99% accuracy. Front Bioeng Biotechnol 8:808–808

Promega_UK. Benchtop Automated DNA/RNA Extraction From Any Sample. 2024 [cited 2024; Available from: https://www.promega.co.uk/products/lab-automation/automated-dna-rna-extraction-purification-maxwell/

QIAGEN. *QIAGEN Ingenuity Pathway Analysis* (QIAGEN IPA). 2024 [cited 2024; Available from: https://digitalinsights.qiagen.com/products-overview/discovery-insights-portfolio/analysis-and-visualization/qiagen-ipa/#

Rampasek L, Goldenberg A (2016) TensorFlow: biology's gateway to deep learning? Cell Syst 2(1):12–14

Reactome. Reactome Pathway Browser. 2022 [cited 2022 23/12/2022]; Available from: https://reactome.org/PathwayBrowser/

Rees-Garbutt J et al (2020) Designing minimal genomes using whole-cell models. Nat Commun 11(1):836–836

Riva SG et al (2022) SMGen: a generator of synthetic models of biochemical reaction networks. Symmetry (basel) 14(1):119

Rojas I et al (2007) SABIO-RK: a database for biochemical reactions and their kinetics. BMC Syst Biol 1(S1):S6–S6

Rosenhahn E et al (2022) Bi-allelic loss-of-function variants in PPFIBP1 cause a neurodevelopmental disorder with microcephaly, epilepsy, and periventricular calcifications. Am J Hum Genet 109(8):1421–1435

Rowe E, Palsson BO, King ZA (2018) Escher-FBA: a web application for interactive flux balance analysis. BMC Syst Biol 12(1):84–84

Rubinacci S et al (2015) CoGNaC: a chaste plugin for the multiscale simulation of gene regulatory networks driving the spatial dynamics of tissues and cancer. Cancer Informatics 2015(Suppl. 4):53–65

Sanassy D, Widera P, Krasnogor N (2015) Meta-stochastic simulation of biochemical models for systems and synthetic biology. ACS Synth Biol 4(1):39–47

SBML. Systems Biology Markup Language Website. 2022 [cited 2022 14/12/2022]; Available from: https://sbml.org/

SCALEBIO. SINGLE CELL RNA SEQUENCING KIT. 2024 [cited 2024; Available from: https://scale.bio/single-cell-rna-sequencing-kit/

Sekiguchi T, Hamada H, Okamoto M (2021) WinBEST-KIT: biochemical reaction simulator for analyzing multi-layered metabolic pathways. Bioengineering (basel) 8(8):114

Sleator RD (2016) JCVI-syn3.0—a synthetic genome stripped bare! Bioengineered 7(2):53–56

Smith LP et al (2009) Antimony: a modular model definition language. Bioinformatics (oxford, England) 25(18):2452–2454

Stephanopoulos G (2012) Synthetic biology and metabolic engineering. ACS Synth Biol 1(11):514–525

Storch M, Haines MC, Baldwin GS (2020) DNA-BOT: a low-cost, automated DNA assembly platform for synthetic biology. Synth Biol (oxford University Press) 5(1):ysaa010–ysaa010

Swainston N et al (2018) PartsGenie: an integrated tool for optimizing and sharing synthetic biology parts. Bioinformatics 34(13):2327–2329

Swat MH et al (2009) Multi-cell simulations of development and disease using the compucell 3D simulation environment. Methods Mol Biol 500:361–428

Synthace. Synthace website. 2022 [cited 2022 21/11/2022]; Available from: https://www.synthace.com/

Systems_Biology_Research_Group. BiGG Models. 2023 [cited 2023 18/01/2023]; Available from: http://bigg.ucsd.edu/

Taketani M et al (2020) Genetic circuit design automation for the gut resident species Bacteroides thetaiotaomicron. Nat Biotechnol 38(8):962–969

Tecan_Trading_AG. Uno Single Cell Dispenser. 2024 [cited 2024; Available from: https://lifesciences.tecan.com/products/liquid_handling_and_automation/uno-single-cell-dispenser

The_CellML_Project. CellML Model Repository. 2022 [cited 2022 14/12/2022]; Available from: https://models.cellml.org/cellml

The_Pan_Cancer_Atlas. Welcome to the Pan-Cancer Atlas. 2022 [cited 2022 23/12/2022]; Available from: https://www.cell.com/pb-assets/consortium/pancanceratlas/pancani3/index.html

Thornton JM, Laskowski RA, Borkakoti N (2021) AlphaFold heralds a data-driven revolution in biology and medicine. Nat Med 27(10):1666–1669

TinkerCell_Website. *TinkerCell*. 2022 [cited 2022 19/12/2022]; Available from: http://www.tinkercell.com

Tong AB et al (2021) Could AlphaFold revolutionize chemical therapeutics? Nat Struct Mol Biol 28(10):771–772

Torabi E, Ghobaei-Arani M, Shahidinejad A (2022) Data replica placement approaches in fog computing: a review. Clust Comput 25(5):3561–3589

Torres MDT, de la Fuente-Nunez C (2019) Toward computer-made artificial antibiotics. Curr Opin Microbiol 51:30–38

UCSD_SBRG. BiGG Models. 2019 [cited 2022 16/12/2022]; Available from: http://bigg.ucsd.edu/

Unchained_Labs. Lipid Nanoparticles. 2024 [cited 2024; Available from: https://www.unchainedlabs.com/lipid-nanoparticles/

van Lent P, Schmitz J, Abeel T (2023) Simulated design–build–test–learn cycles for consistent comparison of machine learning methods in metabolic engineering. ACS Synth Biol 12(9):2588–2599

van Rosmalen RP et al (2021) Model reduction of genome-scale metabolic models as a basis for targeted kinetic models. Metab Eng 64:74–84

Varadi M et al (2022) AlphaFold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. Nucleic Acids Res 50(D1):D439–D444

Vaswani A, et al. (2017) Attention is All You Need. Nips'17. p 6000–6010

Villalobos A et al (2006) Gene designer: a synthetic biology tool for constructing artificial DNA segments. BMC Bioinf 7(1):285–285

VMH. Recon Map 3. 2022 [cited 2022 16/12/2022]; Available from: https://www.vmh.life/minerva/index.xhtml?id=ReconMap-3

VMH. Virtual Metabolic Human. 2022 [cited 2022 16/12/2022]; Available from: https://www.vmh.life/

Walsh DI et al (2019) Standardizing automated DNA assembly: best practices, metrics, and protocols using robots. SLAS Technology 24(3):282–290

Wang X et al (2016) Genetic circuit for the early warning of lung cancer using iBioSim. ITM Web of Conf 7:9019

Watanabe L et al (2019) iBioSim 3: a tool for model-based genetic circuit design. ACS Synth Biol 8(7):1560–1563

Watson J et al (2022) SubcellulaRVis: a web-based tool to simplify and visualise subcellular compartment enrichment. Nucleic Acids Res 50(W1):W718–W725

Weaver DS et al (2014) A genome-scale metabolic flux model of *Escherichia coli* K-12 derived from the EcoCyc database. BMC Syst Biol 8(1):79–79

Whitmore LS et al (2019) RetSynth: determining all optimal and sub-optimal synthetic pathways that facilitate synthesis of target compounds in chassis organisms. BMC Bioinformatics 20(1):461–461

Wishart DS et al (2018) HMDB 40: the human metabolome database for 2018. Nucl Acids Res 46(D1):608-D617

Wu MJ et al (2019) Automated design of diverse stand-alone riboswitches. ACS Synth Biol 8(8):1838–1846

Xia B et al (2011) Developer's and user's guide to Clotho v2.0 A software platform for the creation of synthetic biological systems. Methods Enzymol 498:97–135

Yaman F et al (2012) Automated selection of synthetic biology parts for genetic regulatory networks. ACS Synth Biol 1(8):332–344

Yeoh JW et al (2019) An automated biomodel selection system (BMSS) for gene circuit designs. ACS Synth Biol 8(7):1484–1497

Zhang M et al (2017) SBOLDesigner 2: an intuitive tool for structural genetic design. ACS Synth Biol 6(7):1150–1160

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.