EasyChair Preprint

№ 10074

Quantum Generators: Chip Design for Processing Protein Structures Using AI and Geometric Patterns in Cell Synthesizer Unit

Poondru Prithvinath Reddy

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

May 12, 2023

Quantum Generators: Chip Design for Processing Protein Structures using AI and Geometric Patterns in Cell Synthesizer Unit

Poondru Prithvinath Reddy

ABSTRACT

Quantum Generators is a means of achieving mass food production with short production cycles and when and where required by means of machines rather than land based farming which has serious limitations. The process for agricultural practices for plant growth in different stages is simulated in a machine with a capacity to produce multiple seeds from one seed input using computational models of multiplication (generating multiple copies of kernel in repetition). In this respect, we present a modular platform for automating cell synthesis which embodies synthesis abstraction with complex pathways of protein synthesis therefore, altogether different processor/'computing power' is required to address cell synthesis. Firstly, the automated synthesis could make use of combination of starting materials for planning the synthesis routes to achieve the target molecules and accordingly, neural networks are required to be trained on all possible reactions in cell synthesis for a particular crop. The trained AI system (machine learning) allows for autonomous exploration of synthesis space allowing for discovery of new synthesis transformations and these are automatically interpreted for cell structural patterns also and are then used to update the respective machine learning models. Secondly, an AI agent is designed to learn to optimize the final circuit generation from the cell synthesis requirement/environment. We designed an RL agent to add or to remove the circuits to maintain a correct computation and high-performance computing/ 'computer graphics' , and to build through a series of steps(adding or removing circuits) for improving the synthesis performance & efficiency of cell structural patterns. For this we used fully convolutional neural network the Q-learning algorithm (an RL algorithm) for cell synthesis and the algorithm trained the circuit design agent using a matrix representation for synthesis requirement. Since we have two learning models(Composition and Geometric patterns) along with a learning agent for circuit design, we show an implementation of

combining two of them with small model in obscene of real-world model of CellSynputer for autonomous protein folding/synthesis. In this way, it is possible to script and run desired synthesis with reconfigurable system for diverse protein folding outcomes. Although the platform model given us a method of automating/ optimizing cellular assemblies however, this need to be tested using natural crop cells for quantum generation.

INTRODUCTION

A **Quantum** (plural quanta) is the minimum amount of any physical entity (physical property) involved in an interaction. On the other hand, **Generators** don't actually create anything instead, they generate quantity prescribed by physical property through multiplication to produce high quality products on a mass scale. The aim of Quantum Generators is to produce multiple seeds from one seed at high seed rate to produce a particular class of food grains from specific class of **seed** on mass scale by means of machine rather than land farming.

The process for agricultural practices include preparation of soil, seed sowing, watering, adding manure and fertilizers, irrigation and harvesting. However, if we create same conditions as soil germination, special watering, fertilizers addition and plant growth in different stages in a machine with a capacity to produce multiple seeds from one seed input using computational models of multiplication(generating multiple copies of kernel in repetition) then we will be closure to achieving mass food production by means of quantum generators(machine generated) rather than traditional land based farming which has very serious limitations such as large space requirements, uncontrolled contaminants, etc. The development of Quantum Generators requires specialized knowledge in many fields including Cell Biology, Nanotechnology, 3D Cellprinting, Computing, Soil germination and initially they may be big occupying significantly large space and subsequently small enough to be placed on roof-tops.

The Quantum Generators help world meet the food needs of a growing population while simultaneously providing opportunities and revenue streams for farmers. This is crucial in order to grow enough food for growing populations without needing to expand farmland into wetlands, forests, or other important natural ecosystems. The Quantum Generators use significantly less space compared to farmland and also results in increased yield per square foot with short production cycles, reduced cost of cultivation besides easing storage and transportation requirements.

In addition, Quantum Generators Could Eliminate Agricultural Losses arising out of Cyclones, Floods, Insects, Pests, Droughts, Poor Harvest, Soil Contamination, Land Degradation, Wild Animals, Hailstorms, etc.

Quantum generators could be used to produce most important *food* crop *like* rice, wheat and maize on a mass scale and on-demand when and where required.

Computers and Smartphones have become part of our lives and Quantum Generators could also become very much part of our routine due to its potential benefits in enhancing food production and generating food on-demand wherever required.

METHODOLOGY

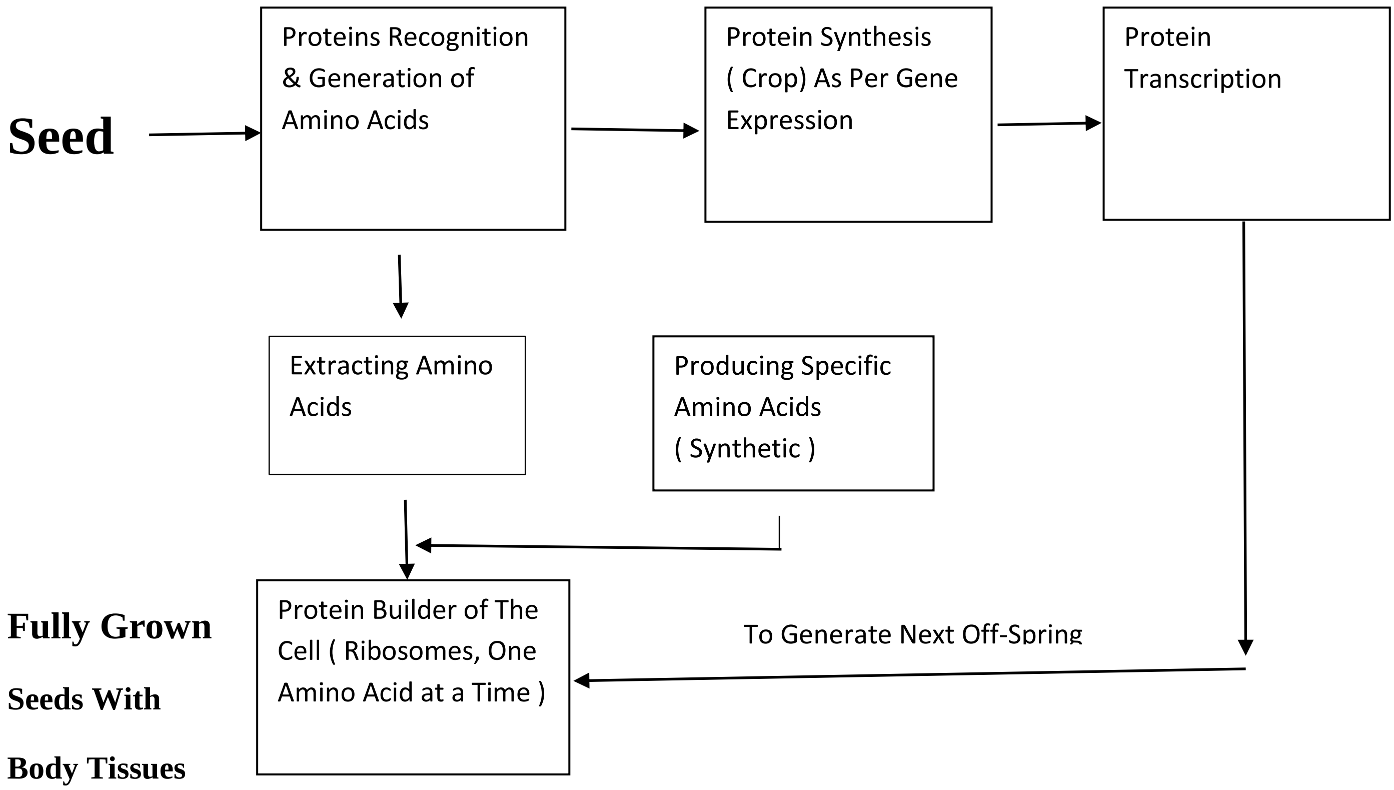


Fig 1. Process Flow Diagram of Seed Builder

Protein from input seeds is broken down into individual amino acids which are reassembled by Quantum Generating ribosomes into proteins that Crop cells need to be generated. The information to produce a protein is encoded in the **cell's** DNA. When a protein is produced, a copy of the DNA is made (called mRNA) and this copy is transported to a ribosome.

Protein **synthesis** is the process used by the QG(Quantum Generator) to make proteins. The first step of protein **synthesis** is called Transcription. It occurs in the nucleus. During transcription, mRNA transcribes (copies) DNA.

Body tissues **grow** by increasing the number of cells that make them up. Every **cell** in the crop body contains protein. The basic structure of protein is a chain of amino acids. We need protein in our diet to help human body repair cells and make new ones.

The major steps in protein synthesis are:

- DNA unzips in the nucleus.
- mRNA nucleotides transcribe the complementary DNA message.
- mRNA leaves nucleus and goes to ribosome.
- mRNA attaches to ribosome and first codon is read.
- tRNA brings in proper amino acid from cytoplasm.
- a second tRNA brings in new amino acid.

The journey from gene to **protein** is complex and tightly controlled within each cell. It consists of two major **steps**: transcription and translation. Together, transcription and translation are known as gene expression. Transcription is the transfer of genetic instructions in DNA to mRNA in the nucleus. Translation occurs at the ribosome, which consists of rRNA and proteins**.**

Ribosomes are the sites in a **cell** in which **protein** synthesis takes place. Cells have many ribosomes, and the exact number depends on how active a particular cell is in synthesizing proteins. **Ribosomes** are the protein builders or the protein synthesizers of the cell. They are like construction guys who connect one amino acid at a time and build long chains.

Ribosomes, large complexes of **protein** and ribonucleic acid (RNA), are the cellular organelles responsible for protein synthesis. They receive their “orders” for protein synthesis from the nucleus where the DNA is transcribed into messenger RNA (mRNA).

During the **process** of transcription, the information stored in a gene's DNA is passed to a similar molecule called RNA (ribonucleic acid) in the cell nucleus. A type of RNA called transfer RNA (tRNA) assembles the protein, one amino acid at a time.

Amino acids can be produced by breaking down proteins, known as the extraction method. However, the amount of amino acids in the source protein limits the amount of amino acids made. Extraction is not good for making mass quantities of specific amino acids**.** So Synthetic Methods of making amino acids is necessary in protein synthesis.

The Quantum Generator contains pre-programmed Protein Synthesizer relevant to specific Crop/Tissue which essentially reassembles ribosomes (Sites in a Cell) into proteins that your crop cells need. The sequence and information to produce a protein is encoded in the synthesizer of Quantum Generator.

Robotics & Machine Learning towards Biological Space Exploration

Machine learning approaches are fundamental to scientific investigation in many disciplines. In biological studies, many of these methods are widely applicable and robotics/automation is helping to progress cell synthesis through biological space exploration. Scientists have begun to embrace the power of machine learning coupled with statistically driven design in their research to predict the performance of synthetic reactions. For our study, the yield of a synthetic reaction can be predicted **machine learning** in the multidimensional space obtained from robotic automation to map the yield landscape of intricate synthesis following synthesis code. Meanwhile, our emphasis is on automation of synthesis, which is controlled by robots/computers rather than by humans. Synthesis through automation offers far better efficiency and accuracy. In addition, the machine learning algorithms explore a wider range of biological space that would need to be performed purely automated random search and it is observed that self-driven laboratories/robots lead the way forward to fast-track synthesis. This brings the development of automation, optimization, and molecular synthesis very close.

Figure 2 shows a graphical representation of workflow for joining automated retrosynthesis with a synthesis robot and reaction optimization. The retrosynthetic module will generate a valid synthesis of the target that can then be transferred into synthesis code that can be executed in a robotic platform. The optimization module can optimize the whole sequence, getting the feedback from the robot.

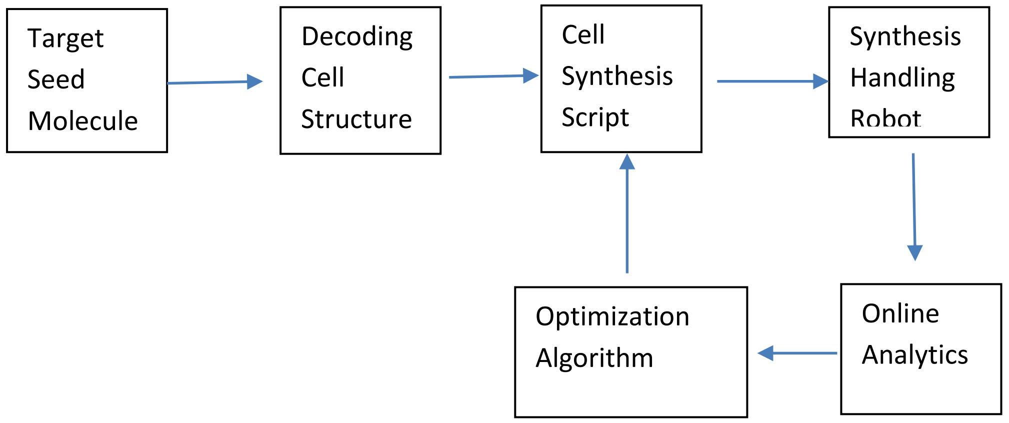


Fig. 2 Architecture of Robotic Synthesis of Crop Cells in a Quantum Generator

The methodology is essentially fundamental for getting the quantum generators as autonomous as possible and also as fast & optimized and the aim is to design processors both CPU and GPU to represent computations and their structural patterns from generator in realizing the desired quantity. Therefore, we use circuit extraction process from the CPU to speed up the synthesis generation and also desired IC's required in GPU for the structural formation. The CPU and GPU are required to be trained separately using reinforcement learning algorithm to arrive at the circuit designs that can easily be adopted and customized from the environment in quantum generators and these are used to localize the requirement.

The methodology primarily consists of following parts:-

1. Designing neural networks on the composition of raw materials (Extracted & Synthetic, enzymes, etc.).
2. Designing machine learning systems for Extracting structural patterns from CellSynputer at each generation step.
3. Introducing learning agent in the processor(CPU & GPU) that uses deep neural network with learning algorithm
4. The neural network used by the learning agent will be trained with learning algorithm by using different methods
5. Measuring the outcome with generator loss or optimization steps
6. Based on generation requirements, get the circuit requirements of the CPU on the basis of computational data.

7. Similarly get the circuit requirements of GPU on the basis of structural pattern in the generation unit
8. Carryout computational data association with the graphical data of the Cellsynputer by matching with the desired data in the crop database.

ARCHITECTURE

Platform Design in Cell Synthesis

Methodologies for the automation of cell synthesis, optimization, and crop yields have not generally been designed for the realities of crop-based yields, instead focussed on engineering solutions to practical problems. We argue that the potential of rapidly developing technologies (e.g., machine learning and robotics) are more fully realized by operating seamlessly with the way that synthetic biologists currently work. This is because the researchers often work by thinking backwards as much as they do forwards when planning a synthetic procedure. To reproduce this fundamental mode of operation, a new universal approach to the automated exploration of cell synthesis space is needed that combines an abstraction of cell synthesis with robotic hardware and closed-loop programming.

Automation Approach

There are different automation approaches for cell synthesis these include block based, iterative, multistep however, we considered CellSynputer which is integration of abstraction, programming and hardware interface, which is given below depicted as in Fig 3.

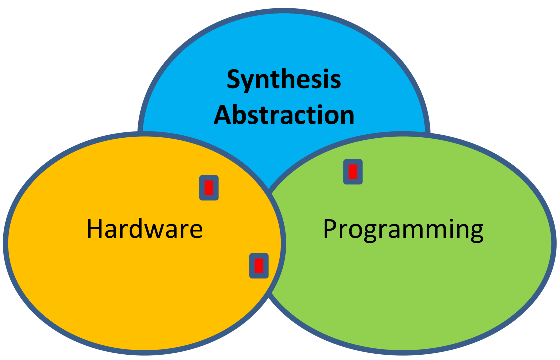


Fig. 3 Approach – Cell Synthesis Automation

Synthetic biologists already benefit from algorithms in the field of cell synthesis and, therefore, automation is one step forward that might help biologists and chemists to plan and develop biological space more quickly, efficiently, and importantly, CellSynputer is a platform that employs a broad range of algorithms interfacing hardware and abstraction to solve synthesis-related problems and surely can very well be established for quantum generation.

Synthesis via Programmable Modular System: 'The CellSynputer'

We presented a modular platform for automating cell synthesis, which embodies our synthesis abstraction in 'the CellSynputer'. Our abstraction of cell synthesis contains the key four stages of synthetic protocols: recognition, gene expression, transcription, and protein builder that can be linked to the physical operations of an automated robotic platform. Software control over hardware allowed combination of individual unit operations into multistep cell synthesis. A CellSynputer was created to program the platform; the system creates low-level instructions for the hardware taking graph representation of the platform and abstraction representing cell synthesis. In this way, it is possible to script and run published syntheses without reconfiguration of the platform, providing that necessary modules are present in the system.

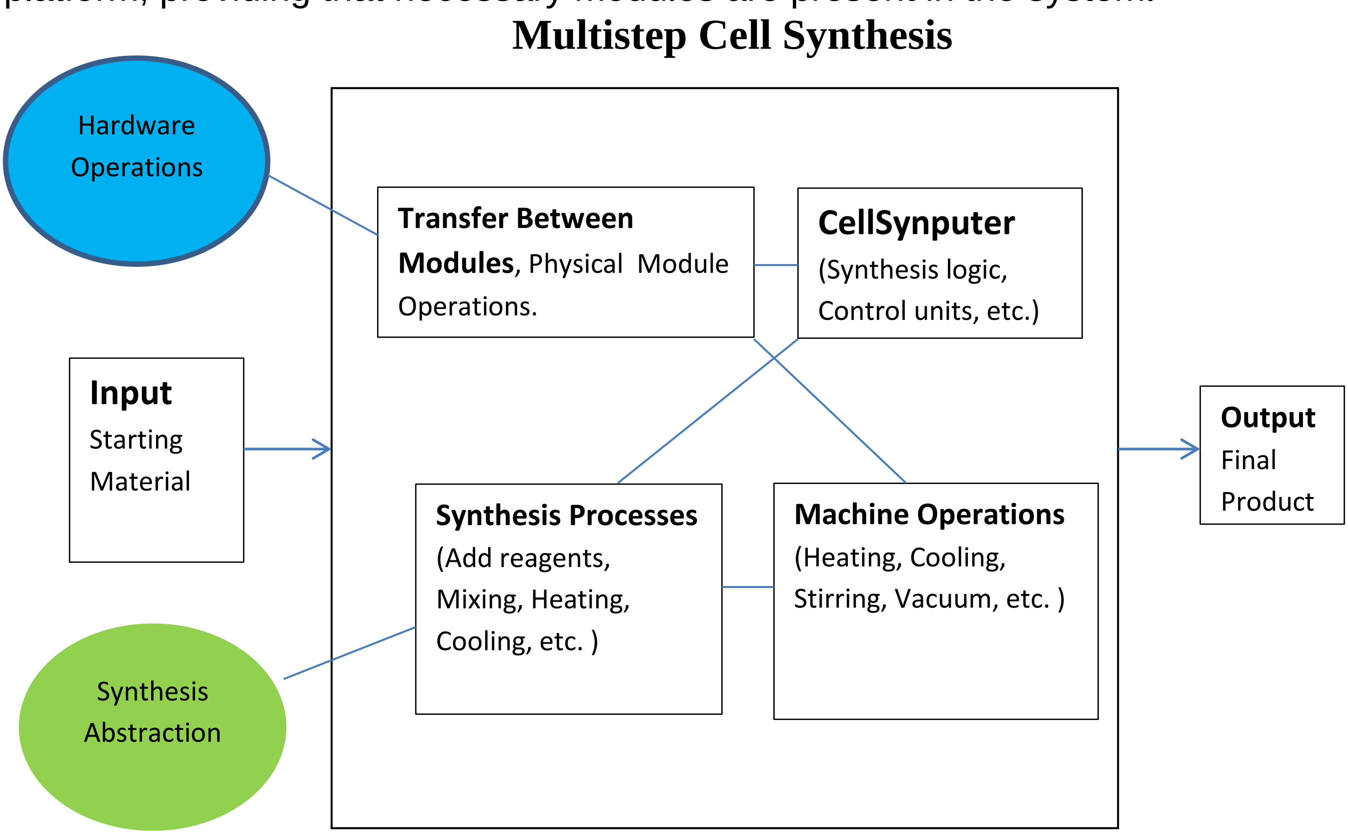


Figure 4. CellSynputer Operational Architecture

Finally, by combining CellSynputer platform and robotic systems with AI, it is possible to build autonomous systems working in closed loop, making decisions based on prior experiments. We already presented a flow system for navigating a network of synthesis reactions utilizing an infrared spectrometer for on-line analysis and as the sensor for data feedback. The system will be able to select the suitable starting materials autonomously on the basis of change in the infrared spectra.

Parallel Synthesizers

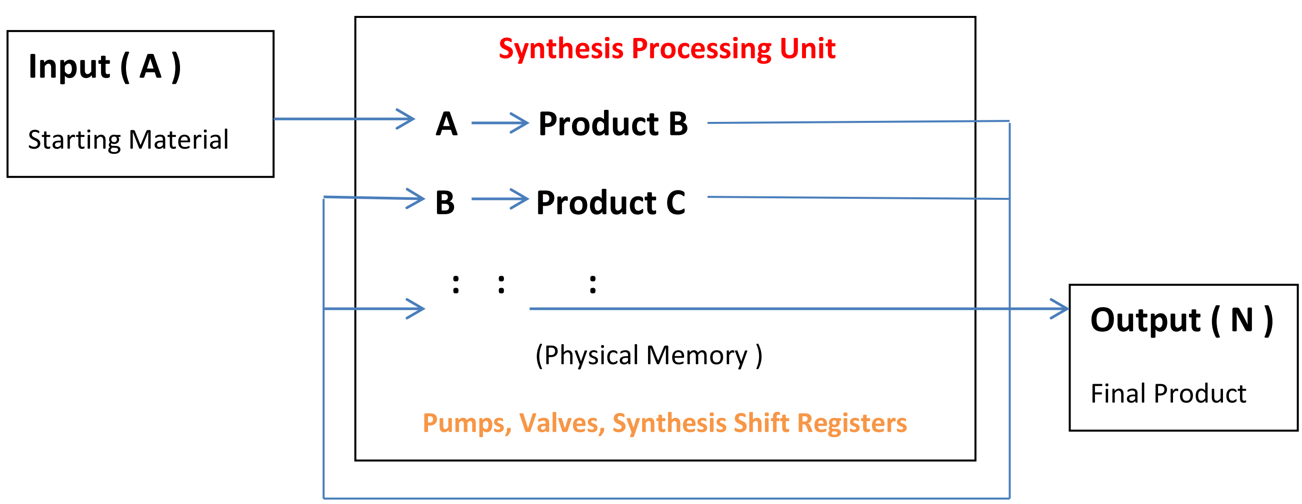
Parallel Synthesizer is a high yielding multiple synthesis systems consisting of parallel processing units & multiple synthesizers and these automated multistep units are used as parallel synthesizers for high yield applications. Parallel synthesis with cell synthesis processes is a way to use the advantages of combinatorial synthesis in a manner that provides a more focused approach to the target molecules. This results in a smaller, more concentrated set of molecules, making the process of unit level synthesis easier.

The following are the attributes of parallel synthesizer:

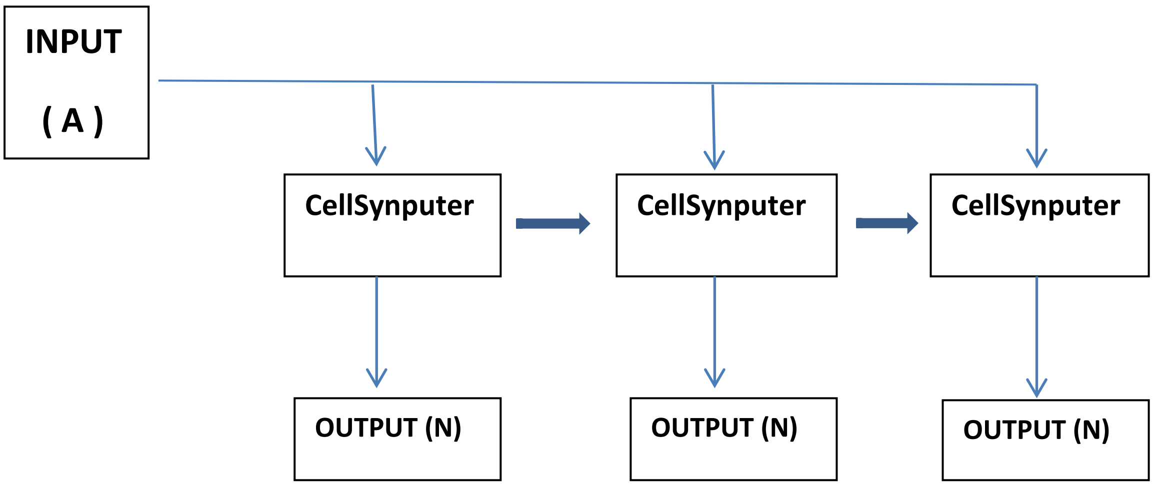
- **Based on multi-unit concept**
- **Configurable at unit level**
- **High throughput**
- **Small scale at unit level**
- **Limited to individual synthesis scope**
- **Embodies multistep procedure**

We give below automated cell synthesis using parallel synthesizer in pictorial format:

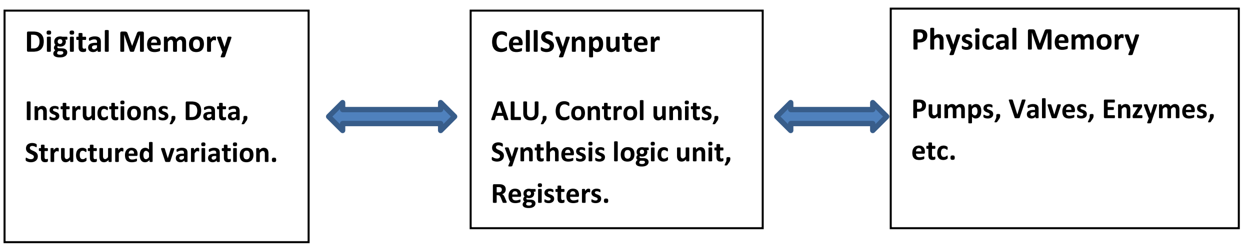
A) N-Step Cell Synthesis



B) Multi-unit Synthesis



C) CellSynputer Architecture



Neural Networks in Exploring Synthesis Space

The automated synthesis could make also use of analysis and combination of starting materials for planning the synthesis routes to achieve the target molecules. There are many approaches to automated cell synthesis, and the one seems to be particularly promising as it employs neural networks and AI and it uses Monte Carlo tree search and symbolic AI to discover target molecule via different synthesis routes. The neural networks are required to be trained on all possible reactions in cell synthesis for a particular crop. The trained AI system allows cracking for many target molecules, faster than the traditional computer-aided search method, which is based on extracted rules and heuristics. In general, this approach allows for faster and more efficient synthesis combination and analysis than any other well-known method. Figure 5 shows a workflow for joining automated synthesis of a target molecule of a desired crop with a synthesis robot and reaction optimization. The synthetic process module will generate a valid synthesis of the target that can then be transferred into synthesis code that can be executed in a CellSynputer/robotic platform. The optimization module can optimize the whole sequence, getting the feedback from the robot.

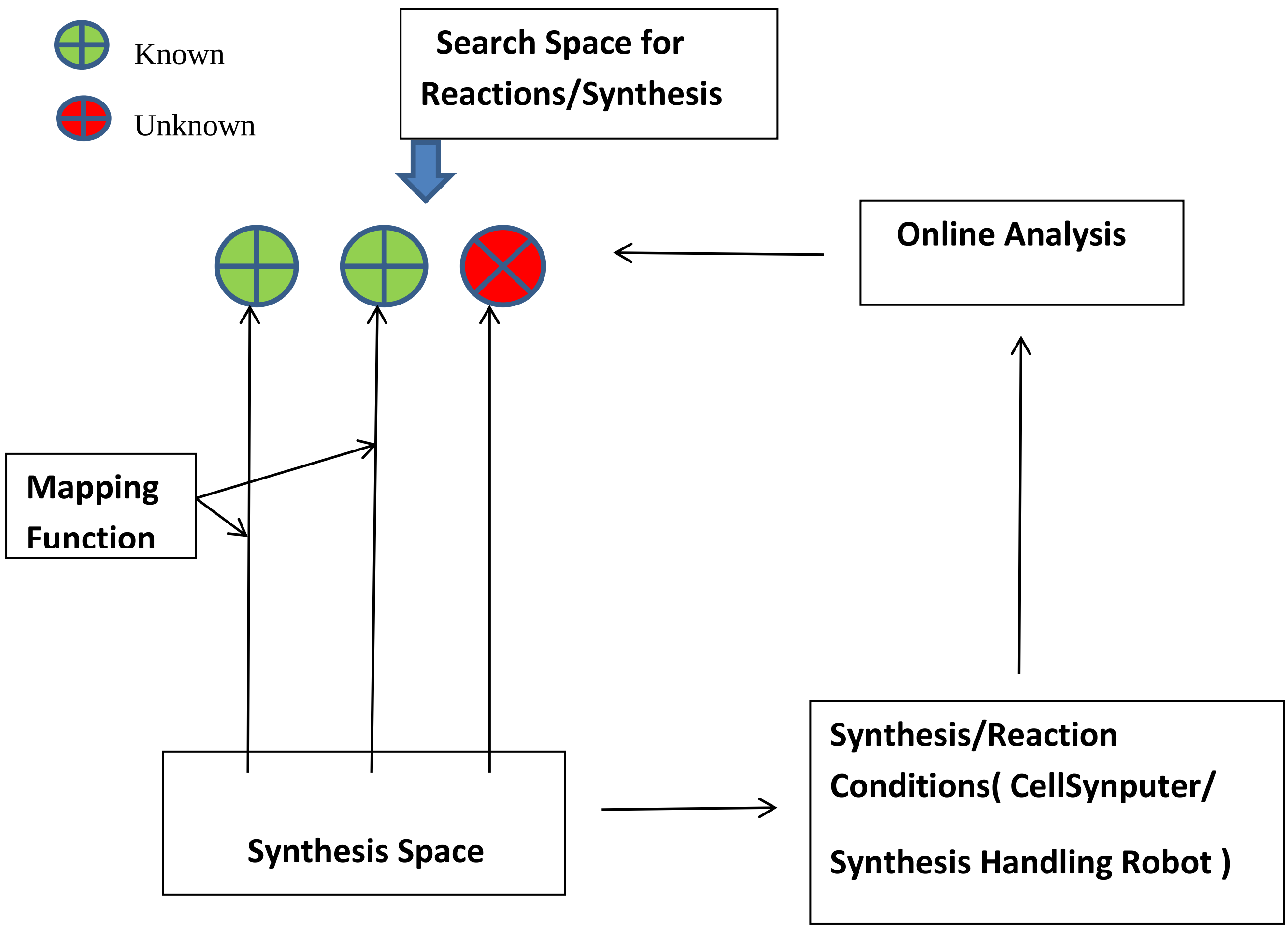


Figure 5 Exploring the Synthesis Space of Experiments with Neural Networks.

The platform operates in a closed loop with a machine learning algorithm; the machine learning algorithm suggest the most promising combinations and reactions that were then conducted and analysed automatically within the platform. The results of each experiment are automatically interpreted and the data are then used to update the machine learning model. The use of machine learning allows for autonomous exploration of synthesis space allowing for discovery of new synthesis transformations.
A standard crop grain composition parameters (like fibre, protein, carbohydrates, etc.) dataset is the first step and the data need to be

collected from different subjects of variety. And also the dataset need to split into training(70%) and test (30%) sets based on data for subjects.

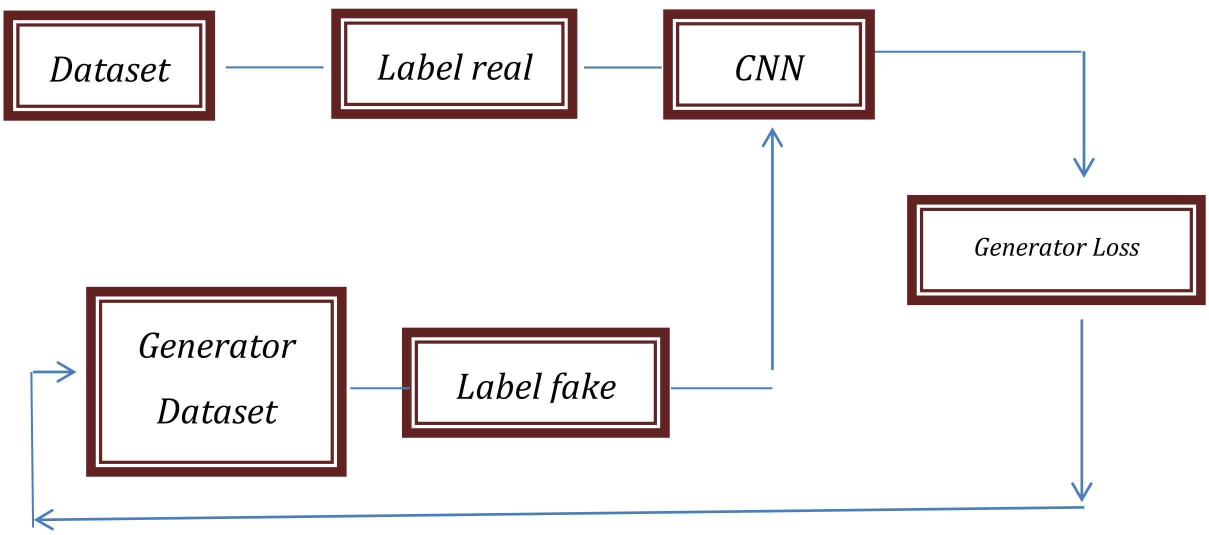
First we must define the CNN model using the deep learning library. We will define the model as having CNN layers, and it is common to define CNN layers in groups of two in order to give the model a good chance of learning features from the input data. CNNs learn very quickly, so the dropout layer is intended to help slow down the learning process and hopefully result in a better final model. The pooling layer reduces the learned features to 1/4 their size, consolidating them to only the most essential elements.

After the CNN and pooling, the learned features are flattened to one long vector and pass through a fully connected layer before the output layer used to make a prediction. The fully connected layer ideally provides a buffer between the learned features and the output with the intent of interpreting the learned features before making a prediction.

The efficient Adam version of stochastic gradient descent will be used to optimize the network, and the categorical cross entropy loss function will be used given that we are learning a multi-class classification problem.

Protein Structures Prediction

We have used slightly different & simplified version of GAN(Generative Adversarial Network) and the following steps are executed back and forth allowing simplified GAN to tackle otherwise difficult generative related predictive problems.



1. Select real images from the training data set
2. Generate a number of fake images(in reality the images are related to synthesized crop tissues in quantum generators) using the generator
3. Train the network(CNN) for one or more epochs using the real images
4. Train the network(CNN) model for one or more epochs using only fake images
5. Compare with real images by calculating the generator loss
6. Finally the backpropagation is performed on the Generator of input images. Here the network weights are not updated but only the Generator is tuned to make it to learn the real requirement**.**

Convolutional Neural Network (CNN) functional model was used for the image processing as it uses multilayer perceptions, and we have used MNIST dataset(dataset of handwritten images) in absence of any real data on protein's chemical contents and its structure.

For this task , the system with different layer configurations for the hidden structures of the networks is as below:

- 2 hidden layers: the first with 28 neurons and a *tanh* activation function; the second with 10 neurons and a *linear* activation fucntion. Dropout rate of 0.5.

We calculated the generator loss, then backpropagation to reduce the loss and to improve the prediction accuracy.

Chip Design Architecture

A **chip** is a carrier of an integrated circuit, which is divided from a wafer and is generally an important part of a computer or other electronic equipment. Simply put, the chip integrates the electronic components such as resistors and capacitors that we can see everywhere and the circuits composed of them into a small particle.

Chip System Specification

1. **Microcontroller** (μC or MCU): An IC containing a processor, memory (RAM, ROM, etc.), and other peripherals is called a microcontroller. This is a general-purpose device and needs to be programmed for the application. Microcontrollers can be used in various industrial products and the approximate components of the hardware are core, storage, peripheral interfaces (high-speed peripherals and low-speed peripherals), bus, interrupt module, clock module, etc.

2. **Processor/Microprocessor** (μP or MPU): An IC that only contains a processor is called a microprocessor. It does not contain memory (RAM, ROM, etc.) or any other peripheral devices.

Here we look at the design aspects of microprocessor. This is nothing but designing a learning agent for improving the cell synthesis performance along with the desired cell structural patterns for autonomous protein folding/synthesis in the form of Circuit Design or operational functionality.

The revolution of modern computing has been largely enabled by remarkable advances in computer systems and hardware. However, majority of today's chips designed are not suitable for high-end computing, resulting in the need to speculate about how to optimize the next generation of chips for the machine learning (ML) models with high end graphics applications. Further, dramatically shortening the chip design requirement would allow hardware to adapt to the rapidly advancing field of ML. The ML itself could provide the means to the chip design requirement , creating a more integrated relationship between hardware and ML, and a deep-learning approach that leverages existing data like blueprints and metrics around power and latency to create accelerator designs that are faster and smaller than chips designed using traditional tools.

Vast arrays of arithmetic circuits have powered CPU/ GPUs thus, improving the design of these arithmetic circuits would be critical in improving the performance and efficiency of CPU/GPUs and use of AI to achieve unprecedented acceleration for AI, high-performance computing, and computer graphics with each chip generation.

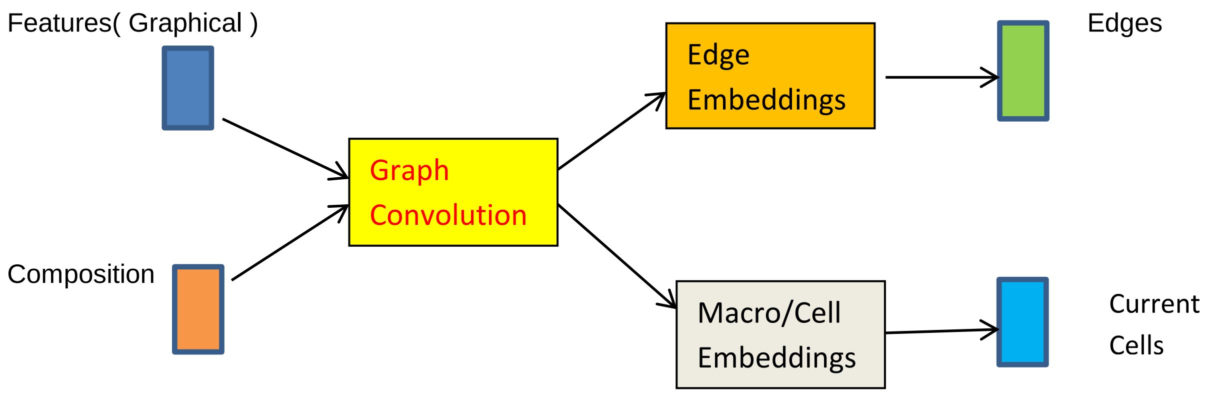
In order for the AI to design with a run at RL agent and the technique proved that AI can not only learn to design circuits from scratch but that those circuits are smaller and faster than circuits designed using the latest design/validation tools. Here an AI agent could design neural graphs and such graph is converted into a circuit with wires and logic gates using a circuit generator. These generated circuits are then further optimized by a physical synthesis tool using synthesis optimizations such as gate sizing, duplication, and buffer insertion.

Arithmetic circuits are built using logic gates like NAND, NOR and XOR and a lot of wires, should be small so more can fit on a chip, fast to reduce any delay that can be a drag on performance and consume as little power as possible, and the 'focus is on the size of the circuit and the speed (for reducing delay), The arithmetic circuit design is represented as a reinforcement learning (RL) task, where we train an agent to optimize the area and delay properties of arithmetic circuits and for this circuits are represented using grid representation with each element in the grid mapping to a graph node, and design an environment where the RL agent can add or remove a node from the circuit graph.

Quantum generator makes the distribution of work across a mix of CPU, and GPU, and at unit level. Networking in this reinforcement learning application is diverse and independent. Quantum generators' ability to switch between CPU and GPU for point-to-point transfer to make transfer model parameters directly from the learner to an inference processor at each processor level.

We propose chip placement as a reinforcement learning (RL) problem, where we train an agent (i.e, an RL policy) to optimize the quality of chip placements. Unlike other methods, this approach has the ability to learn from past experience and improve over time. In particular, as we train over a greater number of chip blocks, the method becomes better at rapidly generating optimized placements for previously unseen chip blocks, and can rapidly generate optimized placements for accelerator chips, and these methods can be applicable to any kind of chip design.

A computer chip is divided into dozens of blocks, each of which is an individual module, such as a memory subsystem, compute unit, or control logic system and these blocks can be described by a graph of circuit components consisting of node types and graph adjacency information. The graph of circuit components representing the composition and structural patterns, are passed through an edge based graph neural networks to encode input state. This generates the embeddings of the placed graph and the candidate nodes.



A graph neural network generates embeddings that are concatenated with the basic crop meta data to form the input to the policy and value requirement of chip design for quantum generation. The policy network generates a probability distribution overall possible grid cells onto which the current node/cells could be placed.

RESULTS

In obscene of graph databases using graph representation for machine learning systems for managing circuit generation data, we build and store the graphs in a simple read format i.e. matrix representations (stored as a node or record with edge list) to perform link prediction.

We have represented this model as matrix with encoded values with possible values for each of the nodes along with the link attributes. We populated the matrix data with randomly generated data and simulated to represent the real world circuit elements/nodes.
The system with different configurations for the hidden structures of the networks:

- 2 hidden layers: the first with 30 neurons and a *tanh* activation function; the second with 15 neurons and a *linear* activation function. No dropout.
- 2 hidden layers: the first with 30 neurons and a *tanh* activation function; the second with 15 neurons and a *linear* activation function. Dropout rate of 0.5.

The dropout rate of 0.5 has been chosen because it seems to be optimal for a wide range of networks.
The results for our CNN based model – RL policy model – The networks that do not use dropout seem to learn well. The percentage of desired generation for the networks (without dropout) is high.
Although we only have partial results, we can make the following observations: the networks that do not use dropout seem to learn well, while the network using dropout does not; it either learns very slowly or just converges to very low level of generation requirements.

CONCLUSION

Quantum Generators (QG) creates new seeds iteratively using the single input seed and the process leads to a phenomenon of generating multiple copies of kernels in repetition. We presented a robotic synthesis equipped with AI-driven learning that can effectively explore unknown and complex phenomenon of protein folding in cell synthesizer and is also designed an AI chip with RL agent (Q-learning) to add or to remove the circuits to maintain a correct computation and to build through a series of steps(adding or removing circuits) for improving the performance & efficiency of cell structural patterns in an open-ended way. In this way, an automation assisted synthesizer with reconfigurable system that is part of CellSynputer is feasible for automated experimentation of diverse protein folding outcomes depending on the crop tissues and in that respect an implementation of Reinforcement Learning agent as a part of AI processor based on small model is presented. Although the platform model with learning agents given us a method of automating and optimizing cellular assemblies however, this need to be tested using natural crop cells for quantum generation.

REFERENCE

1. Poondru Prithvinath Reddy: "**Quantum Generators: A Platform for Automated Synthesis in a Modular Robotic System Driven by Cell Programming**", Google Scholar.