Quantum Generators: Application of Machine Learning Solution That Uses Swarm Intelligence and Edge Computing Technology to Enable Peer-to-Peer Collaboration and Collective Behavior

Poondru Prithvinath Reddy

January 6, 2024

Quantum Generators: Application of machine learning solution that uses swarm intelligence and edge computing technology to enable peer-to-peer collaboration and collective behavior

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 neural processing units with 'multi-features extraction' 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. We applied machine learning system based on best fit conditions and accordingly all quantum generating units follow the swarm intelligence concept in generating desired crops by matching the best cell generated in any of the units. We presented a robotic synthesis equipped with swarm intelligence based AI-driven learning that can effectively explore protein folding in cell synthesizer and also designed a differential evolution algorithm with sphere objective function with swarm learning to add or to remove the controls to maintain a correct synthesis and to build through a series of steps(adding or removing controlled materials) for improving the performance & efficiency of cell structural patterns in an open-ended way. For this we used swarm intelligence and evolution strategies, the differential evolution algorithm (a different version) on synthesizer units for capturing, optimizing and for executing diverse cell synthesis. In this way, a group of swarm computing units assisted synthesizer with reconfigurable system consisting of edge computing that is part of CellSynputer is feasible for automated experimentation and in that respect an implementation of Differential Evolution algorithm for managing swarm of process controls as a part of microcontroller unit

based on small model is presented. Although the platform model with swarm intelligence as microcontroller unit given us a method of automating and 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

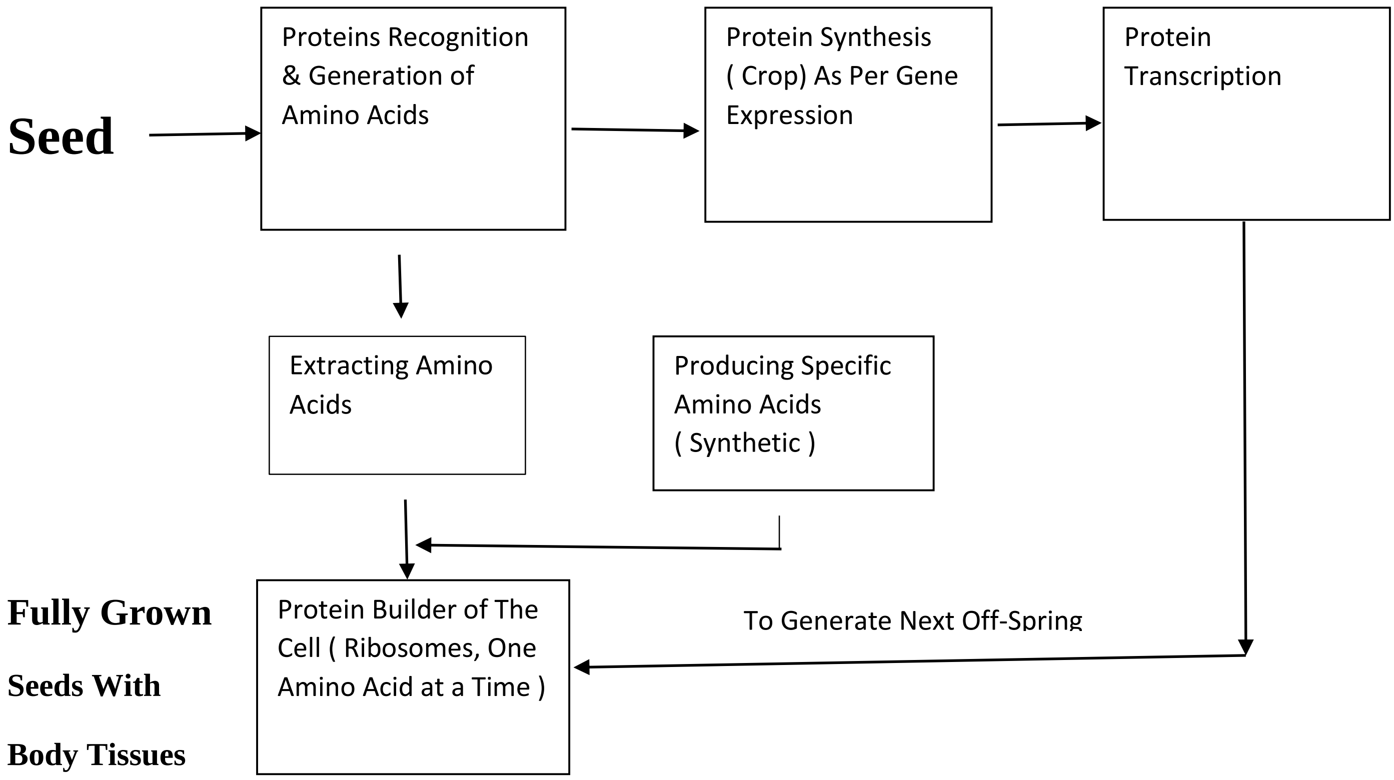


Fig 1. Process Flow Diagram of Seed Builder

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 for Automation and Optimization in Cell Synthesis

We believe 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. To reproduce this fundamental mode of operation, a new approach to the automated exploration of biological space is needed that combines an abstraction of biological synthesis with robotic hardware and closed-loop programming.

As there is a growing drive to exploit rapidly growing robotic technologies along with artificial intelligence-based approaches and applying this to cell synthesis requires a holistic approach. Here, we outline an approach to this problem beginning with an abstract representation of the practice of cell synthesis that then informs the programming and automation required for its practical realization. Using this foundation to construct closed-loop robotic synthesis engine, we can generate new synthesises that may be optimized, and repeated entirely automatically. These robots can perform synthesis reactions and analyses much faster than that can be done by other means. As such, this leads to a road map whereby molecules can be synthesized, optimized, and made on demand from a digital code.

The ability to make small molecules autonomously and automatically will be fundamental to many applications, including quantum generators. Additionally, automated synthesis requires (in many cases) optimization of reaction yields; following optimization, the best conditions can be fed to the synthesis robot to increase the overall yield. There are different approaches to automated yield optimization, and as optimization of reaction conditions requires live feedback from the robotic system, many different detectors are required to monitor progress of the reactions, including benchtop nuclear magnetic resonance spectroscopy, Raman spectroscopy, UV-Vis spectroscopy, etc. Harvested data are then fed to optimization algorithms to explore often the multidimensional parameter space. The platform could be easily reconfigured to the desired task in a plug-and-play fashion, by attaching different modules to the platform core and Robotic approaches also promise to speed up biological space exploration and realization.

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. For our study, the yield of a synthetic reaction can be predicted using **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 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.

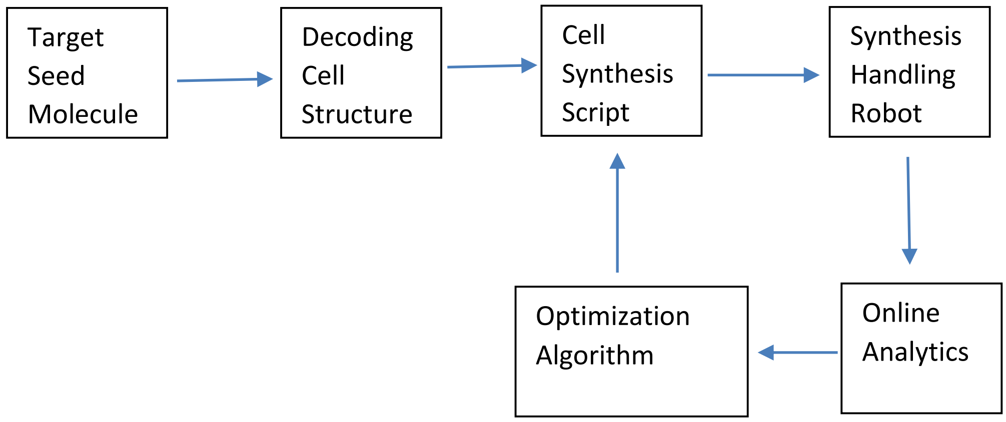


Fig. 2 Architecture of Robotic Synthesis of Crop Cells in a Quantum Generator

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 and also controls required for the microcontroller in synthesizer unit from generator in realizing the desired quantity. Therefore, we use circuit extraction process from the CPU and desired IC's required in GPU and also final control generation required for microcontroller for the structural formation. The CPU and GPU are required to be trained separately and also microcontroller is to trained independently using reinforcement learning algorithm to arrive at the designs that can easily be adopted and customized from the environment in quantum generators and these are used to localize the requirement.

Swarm Intelligence

Swarm intelligence is **a form of collective learning and decision-making based on decentralized, self-organized systems**. Natural examples are commonplace—flocks of birds and schools of fish act and react as groups, without instruction or direction from any single leader.

Swarm learning is **a decentralized machine learning solution that uses edge computing and blockchain technology to enable peer-to-peer collaboration**.

The algorithm used in swarm intelligence is

- Evolutionary algorithms (EA), particle swarm optimization (PSO), differential evolution (DE), ant colony optimization (ACO) and their variants dominate the field of nature-inspired metaheuristics.

The methodology primarily consists of following parts :-

1. Swarm intelligence is a branch of Artificial Intelligence where we observe computing units and try to learn how different cell process phenomenon can be imitated in a CellSynputer to optimize the cell generation
2. In swarm intelligence, we focus on the collective behavior of computing units and their interactions with the CellSynputer environment
3. In Quantum Generators, the focus is on a group of compute units and this group is referred to as 'swarm'

4. The compute units on a Quantum Generators can be correlated with the tasks in a CellSynputer which are awaiting for resources with best suitable environmental conditions. On careful observation of Swarm, the compute units do not know where the resources are located, but they do know what is to be achieved.
5. Thus, the best approach for finding the target cell synthesis, is to follow the compute units which are generating nearest to the target molecule
6. This behavior of compute units is simulated in the computation environment and the algorithm so designed is variant version of Differential Evolution

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, and 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.

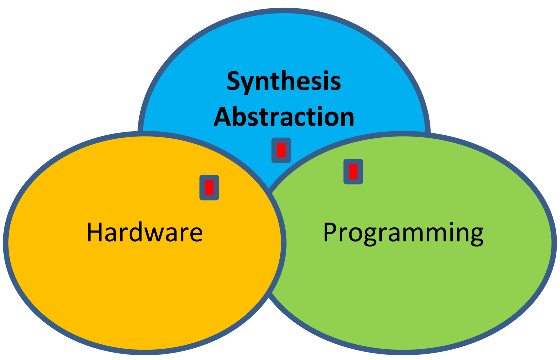


Fig. 3 Approach – Cell Synthesis Automation

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.

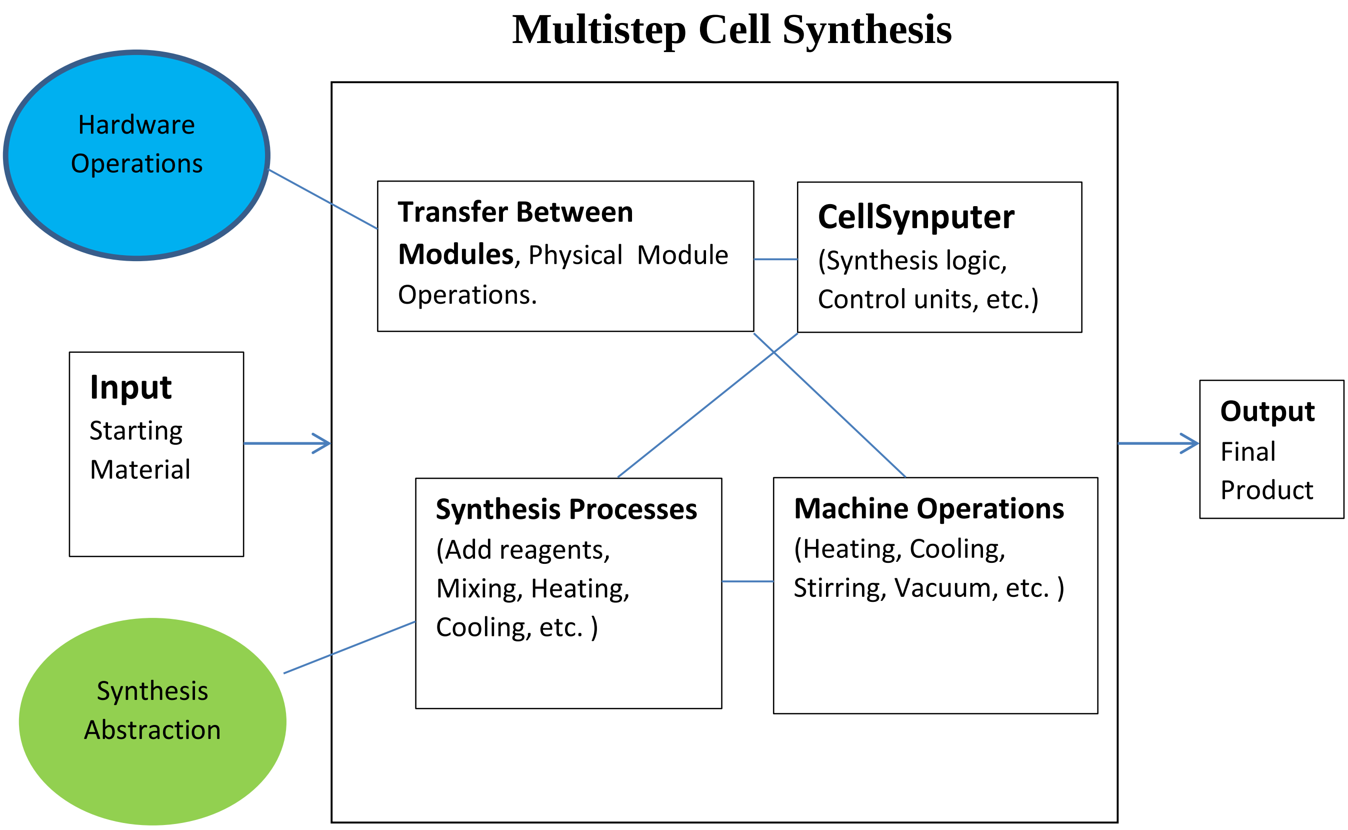


Figure 4. CellSynputer Operational Architecture

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

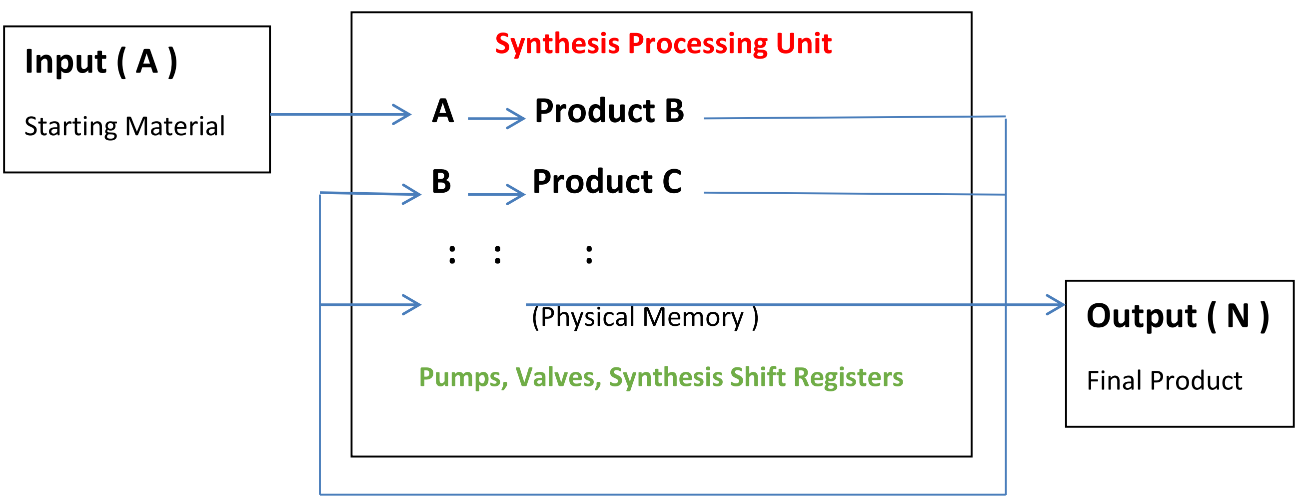
Parallel Synthesizer is a high yielding multiple synthesis systems consisting of parallel processing units & multiple synthesizers and these automated multistep units are used as parallel synthesizers for high yield applications. Parallel synthesis with cell synthesis processes is a way to use the advantages of combinatorial synthesis and 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:

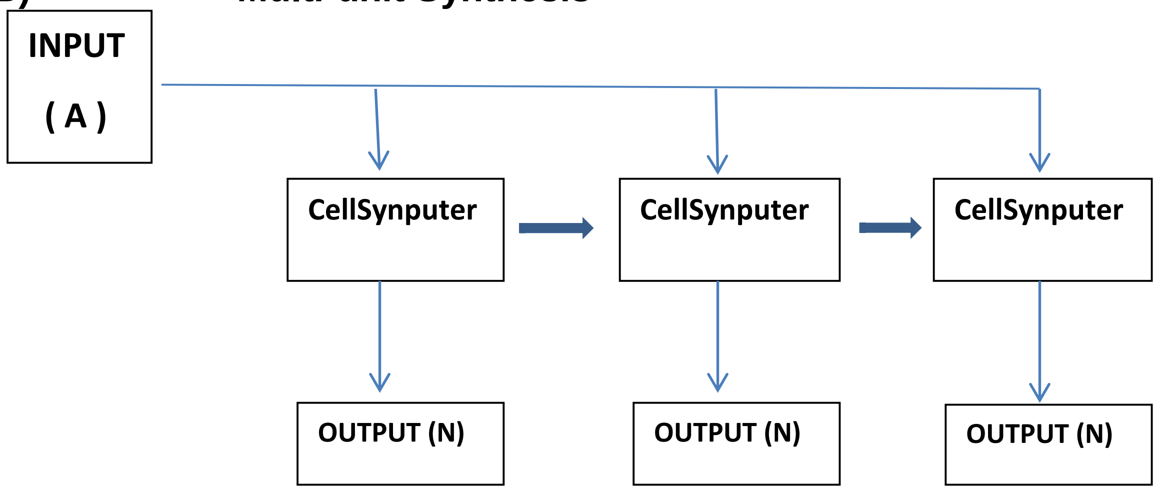
- **Based on multi-unit concept**
- **Configurable at unit level**
- **High throughput**
- **Small scale at unit level**
- **Limited to individual synthesis scope**
- **Embodies multistep procedure**

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

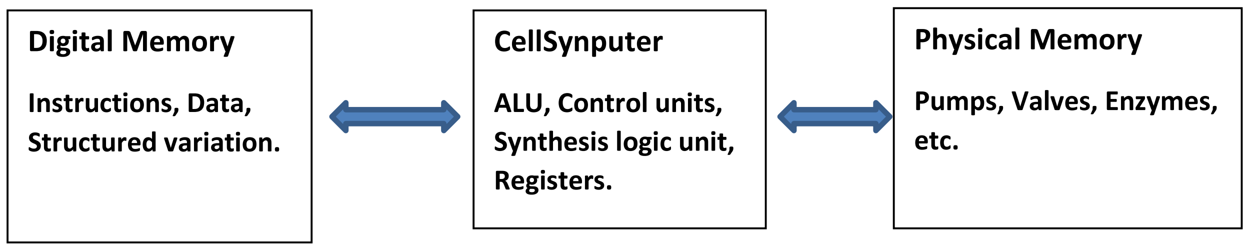
A) N-Step Cell Synthesis



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, and 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.

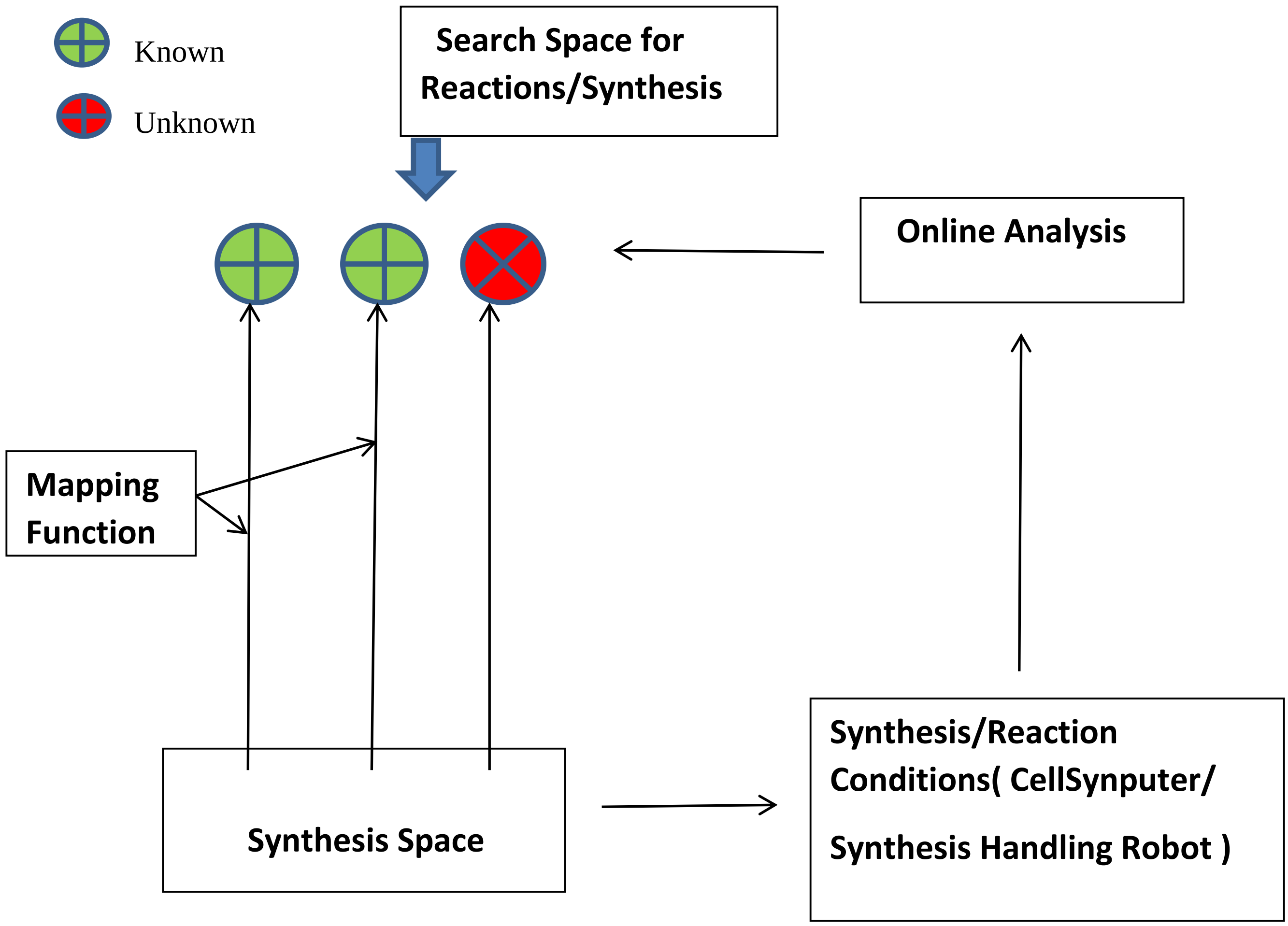


Figure 5. Exploring the Synthesis Space of Experiments with Neural Networks.

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.

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.

Robotic Microcontroller

A **microcontroller** is a compact integrated circuit designed to govern a specific operation in an embedded system. A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip.

A robot microcontroller is basically the brain of the robot. It is used to collect the information from various input devices such as sensors, switches and others. Then it executes a program and in accordance with it controls the output devices such as motors, lights and others

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, and other embedded systems and one of the main application of Microcontroller is sensing and controlling (process control) devices and this feature will be used in automatically controlled flow in CellSynputer.

Differential Evolution Algorithm

The differential evolution algorithm belongs to a broader family of evolutionary computing algorithms. Similar to other popular direct search approaches, such as genetic algorithms and evolution strategies, the differential evolution algorithm starts with an initial population of candidate solutions. These candidate solutions are iteratively improved by introducing mutations into the population, and retaining the fittest candidate solutions that yield a lower objective function value.

The algorithm begins by randomly initiating a population of real-valued crop parameter vectors, also known as genomes or chromosomes. These represent the candidate's solutions to the multi- dimensional optimisation problem.

At each iteration, the algorithm introduces mutations (change in genetic structure) into the population to generate new candidate solutions. The mutation process adds the difference in population vectors to produce a mutated vector. The parameters of the mutated vector are again compared for differential evaluation with the parameters of another predetermined vector, the target vector, during a process known as crossover that aims to increase the diversity of the perturbed parameter vectors. The resulting vector is known as the trial vector.

A final selection operation replaces the target vector, or the parent, by the trial vector, its offspring, if the latter yields a lower objective function value. Hence, the fitter offspring now becomes a member of the newly generated population, and subsequently participates in the mutation of further population members. These iterations continue until a termination criterion is reached.

Implementation

The differential evolution algorithm begins by generating an initial population of candidate solution. Here, we specify crop parameters/composition with the bounds (lower, upper bounds), where the bounds are specified in the form of 2D array with each dimension corresponding to each input variable.

We used Objective Function – the Sphere function with bounds on each input variable and evaluated our initial population of candidate solutions by passing to the objective function as input argument.

We combined all steps together into a differential_evolution() function that takes as input arguments the population size, the bounds of each input variable, the total number of iterations, the mutation scale factor and the crossover rate, and returns the best solution found.

A final selection step replaces the target vector by the trial vector if the latter yields a lower objective function value. For this purpose, we evaluated both vectors on the objective function and subsequently performed selection, then storing the new objective function value if the trial vector is found to be the fittest of the two.

We replaced the values in objective function with better ones as the population evolves and converges towards an optimal solution.

RESULTS

In order to do implementation, we must define values for population size, the number of iterations, the mutation scale factor and the crossover rate. We set their values randomly to 8, 30, 0.5 and 0.6 respectively.

We also defined the bounds of each input variable.

Next, we carry out the search and reported results are somewhat close to the desired composition.

Running the algorithm and checking the progress of the search including the iteration number, the response from the objective function each time an improvement is detected.

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 swarm intelligence based AI-driven learning that can effectively explore unknown and complex phenomenon of protein folding in cell synthesizer and is also designed a differential evolution algorithm with sphere objective function with swarm learning to add or to remove the controls to maintain a correct synthesis and to build through a series of steps(adding or removing controlled materials) for improving the performance & efficiency of cell structural patterns in an open-ended way. In this way, a group of swarm computing units assisted synthesizer with reconfigurable system consisting of edge computing 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 Differential Evolution algorithm for managing swarm of process controls as a part of microcontroller unit based on small model is presented. Although the platform model with swarm intelligence as microcontroller unit 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.