Quantum Generators: Designing Simulator Assisted Algorithms to Aid in Multi-Step Cell Synthesis for On-Demand Crops

Poondru Prithvinath Reddy

November 26, 2022

Quantum Generators: Designing Simulator Assisted Algorithms to Aid in Multi-step Cell Synthesis for On-demand Crops

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). Biological systems contain complex metabolic pathways with many synergies that make them difficult to predict from first principles and Protein synthesis is an example of such a pathway. Here we show how protein synthesis may be improved through a design of simulator for automated synthesizer with a series of iterated high-throughput experiments guided by a machine-learning algorithm implementing a form of optimization of experiments. The algorithm predicts fruitful experiments from statistical models of the previous experimental results, combined with stochastic exploration of the experimental space and with synthesis abstraction in-built into the experimental space. The desired experimental response, or optimized fitness, was defined as the yield of the target product, and new experimental conditions were synergistically combined with automation in CellSynputer (where the unit level computer creates low-level instructions for the hardware taking interface representation of the platform and abstraction representing cell synthesis) and discovered to have improved yield when graphically interpreted. In this way, it is possible to script and run desired synthesis for assessing outcome for multiple crop tissues simultaneously. Although the platform model given us a method of automating cellular assemblies in an intelligent framework embodied in multi-unit system however, this need to be tested using natural crop cells and it could be promising for us in achieving 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 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 by generating food on-demand wherever required by bringing critical advanced technologies into the farmland practices.

METHODOLOGY

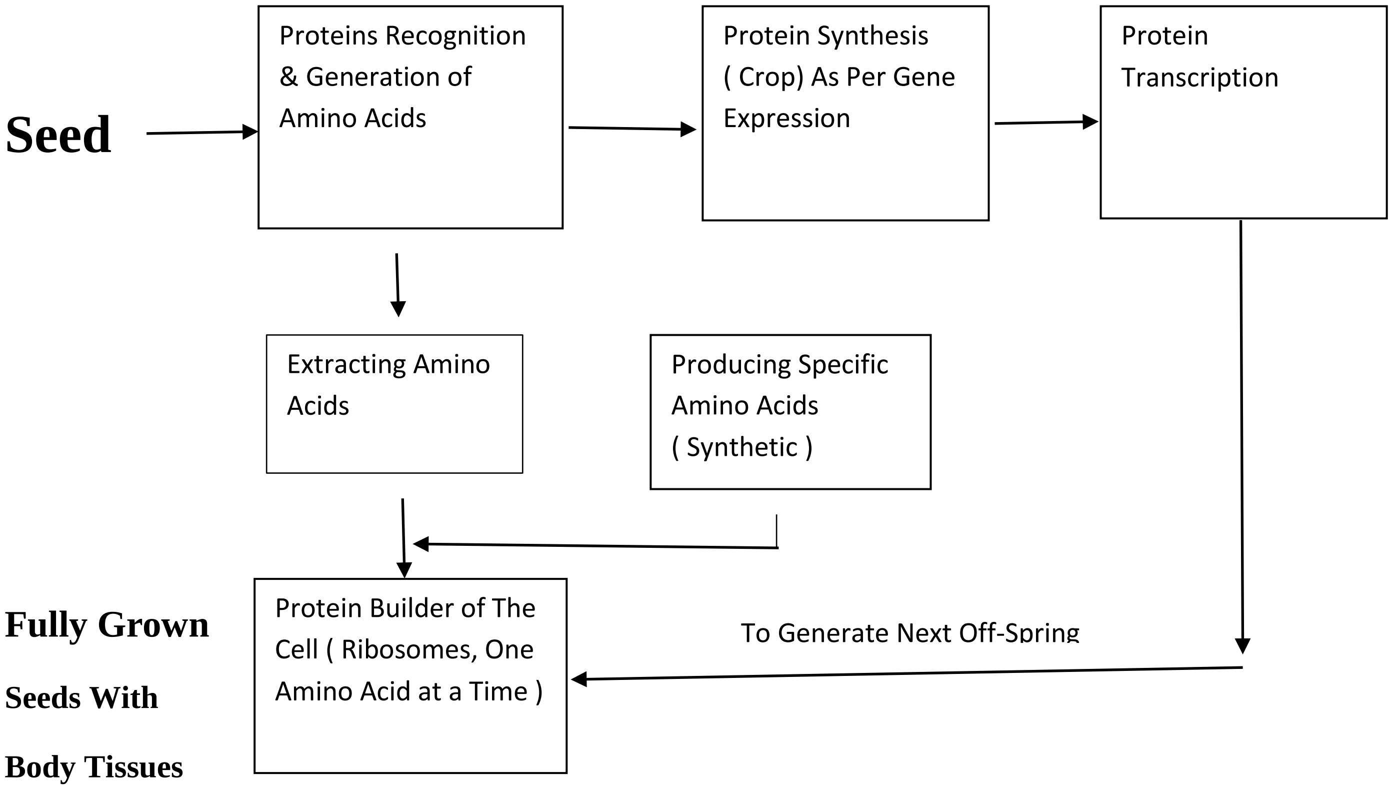

Seed
Proteins Recognition & Generation of Amino Acids
Protein Synthesis (Crop) As Per Gene Expression
Protein Transcription
Extracting Amino Acids
Producing Specific Amino Acids (Synthetic)
Fully Grown Seeds With Body Tissues
Protein Builder of The Cell (Ribosomes, One Amino Acid at a Time)
To Generate Next Off-Spring

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.

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.

Protein synthesis is the process in which **cells make proteins**. It occurs in two stages: transcription and translation. 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 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 are special because they are found in both prokaryotes and eukaryotes.

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).

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 and beyond. 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 allowing improved prediction of high-yielding conditions and replication mechanisms. Meanwhile, our emphasis is on automation of synthesis, which is controlled by robots/computers rather than by humans. In addition, the machine learning algorithms explore a wider range of biological space that would need to be performed purely automated random search and lead the way forward to fast-track synthesis.

In general, this approach allows for faster and more efficient retrosynthetic analysis than any other well-known method. 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.

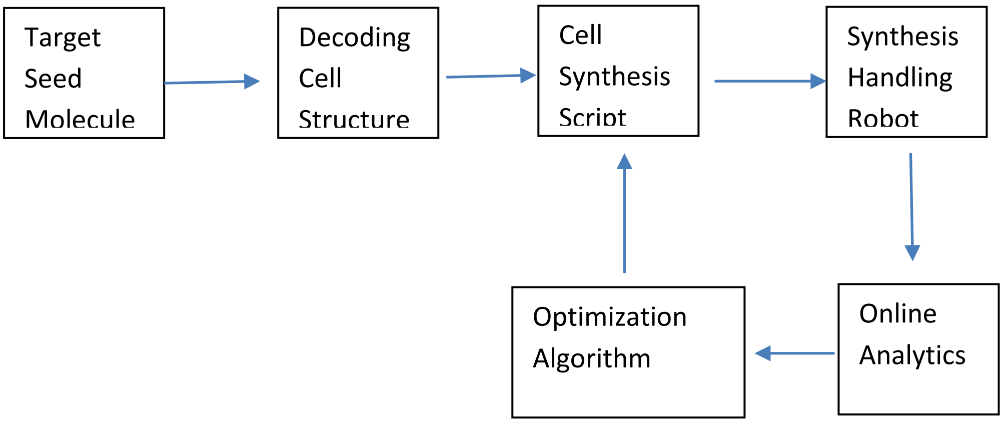


Fig. 2 Architecture of Robotic Synthesis of Crop Cells in a Quantum Generator

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.

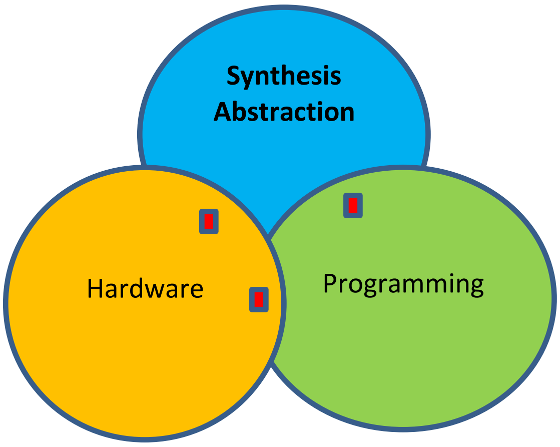


Fig. 3 Approach – Cell Synthesis Automation

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. The synthesis of different small crop molecules on the system can be successfully scripted and performed automatically with yields comparable to traditional methods.

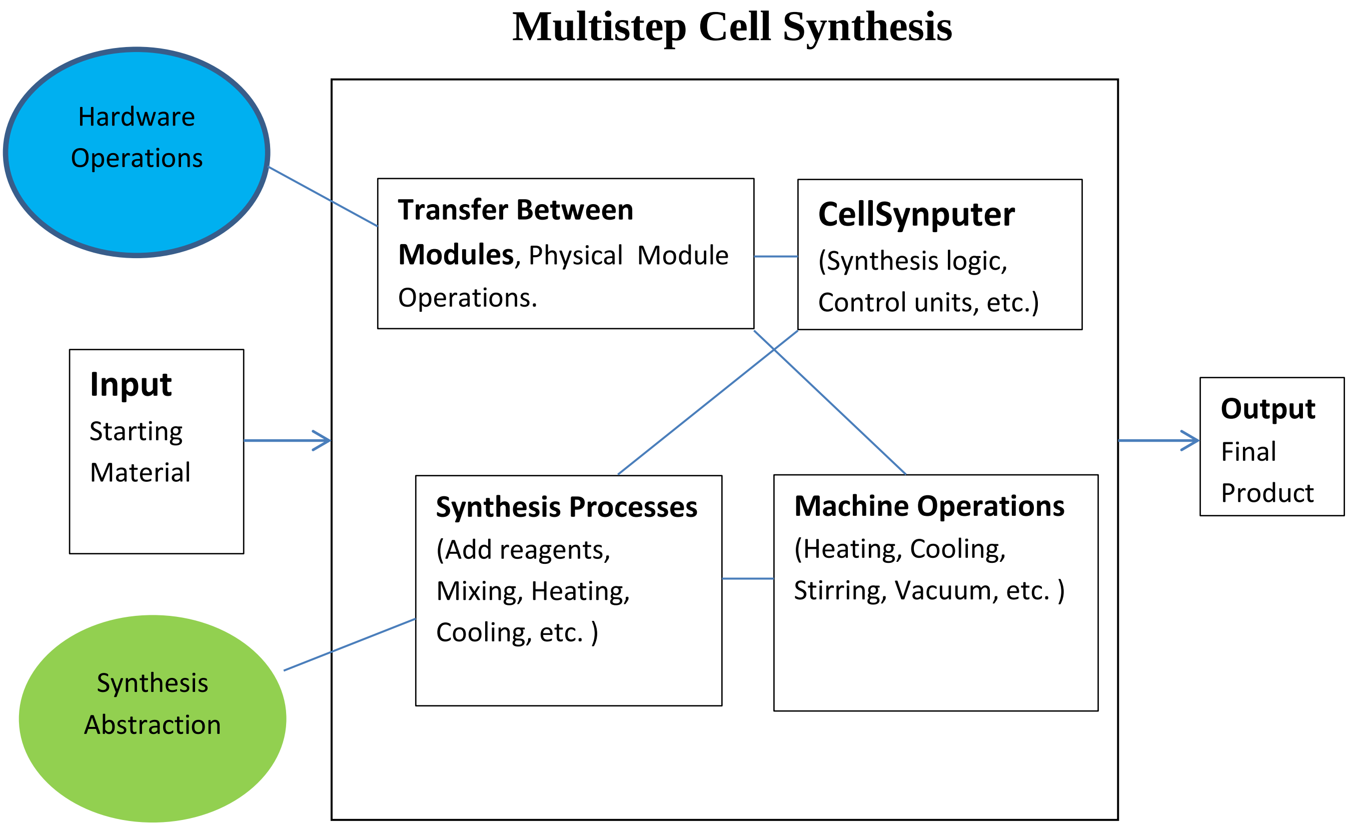

Multistep Cell Synthesis
Hardware Operations
Input Starting Material
Synthesis Abstraction
Transfer Between Modules, Physical Module Operations.
CellSynputer (Synthesis logic, Control units, etc.)
Synthesis Processes (Add reagents, Mixing, Heating, Cooling, etc.)
Machine Operations (Heating, Cooling, Stirring, Vacuum, etc.)
Output Final Product

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 and reactive conditions. 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 most reactive or suitable starting materials autonomously.

Parallel Synthesizers

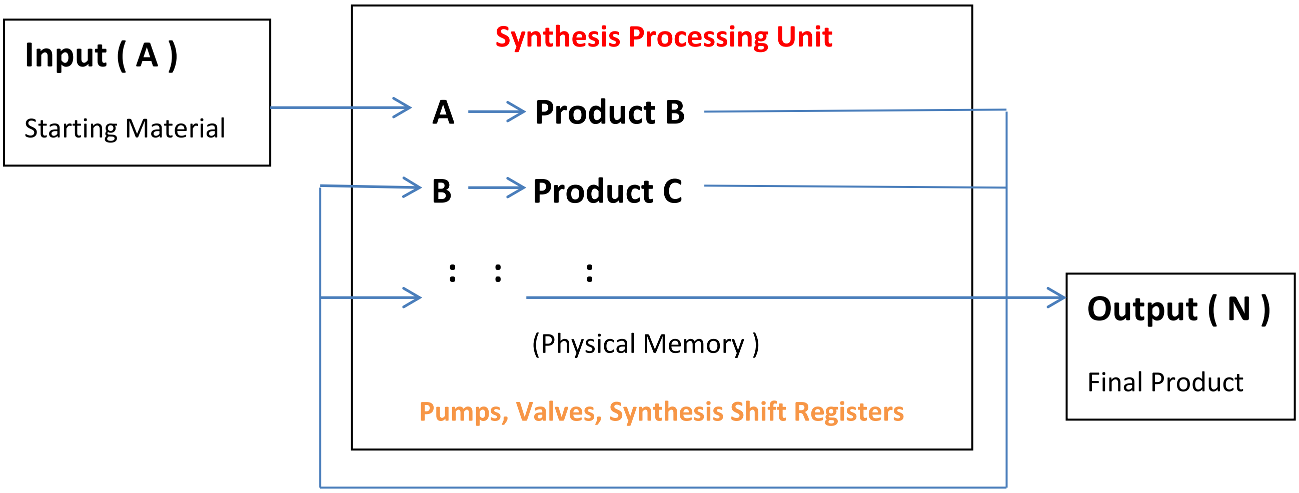
Parallel Synthesizer is a high yielding multiple synthesis systems consisting of parallel processing units & multiple synthesizers, in parallel. These automated multistep units are used as parallel synthesizers for high yield applications. Parallel synthesis with cell synthesis processes is a way to 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:

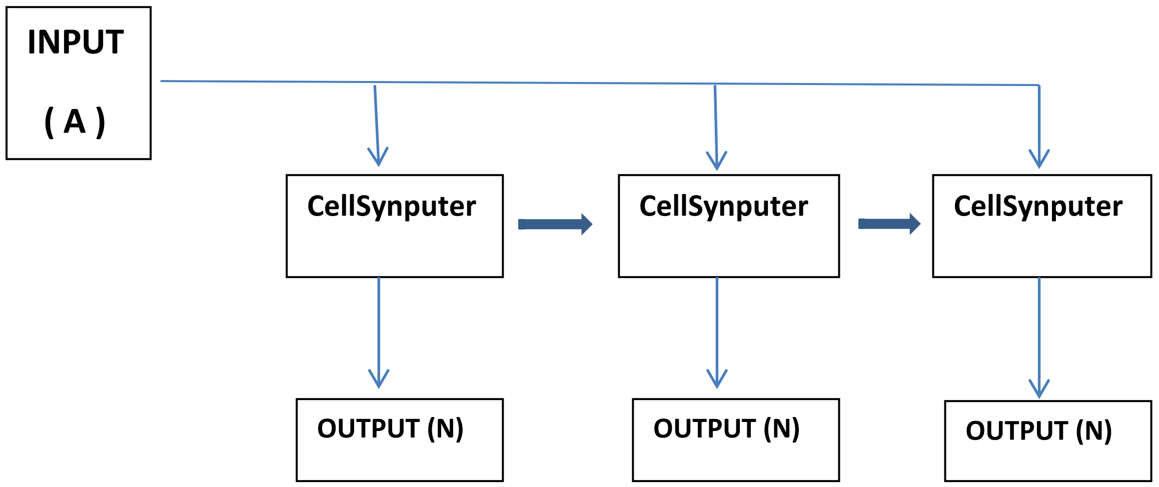
- **Based on multi-unit concept**
- **Configurable at unit level**
- **High throughput**
- **Small scale at unit level**
- **Limited to individual synthesis scope**
- **Embodies multistep procedure**

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

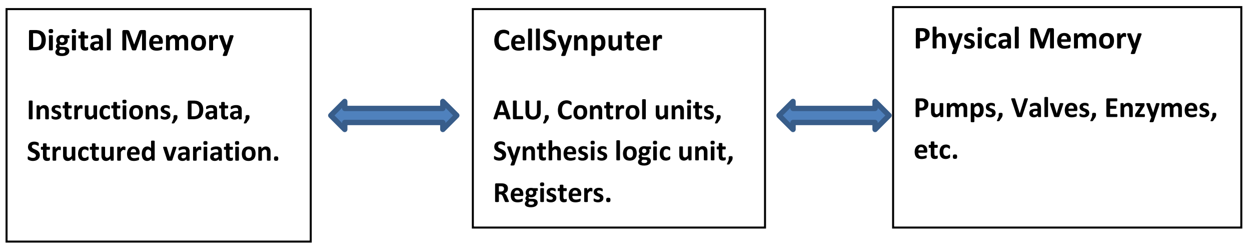
A) N-Step Cell Synthesis



B) Multi-unit Synthesis

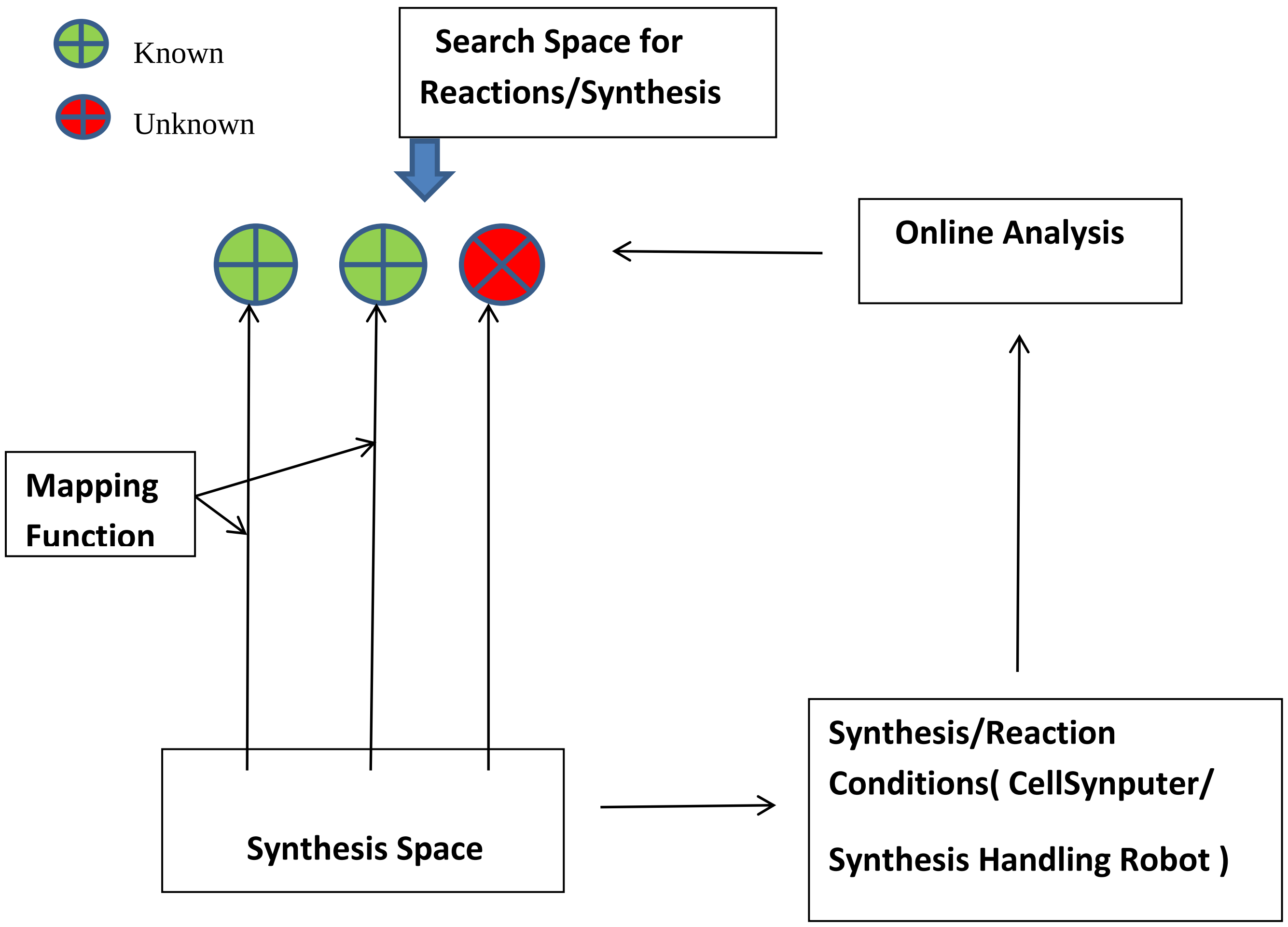


C) CellSynputer Architecture



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. 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.



Known
Unknown
Search Space for Reactions/Synthesis
Online Analysis
Mapping Function
Synthesis Space
Synthesis/Reaction Conditions(CellSynputer/ Synthesis Handling 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 analyzed 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.

Synthesis Framework

The exploration of biological space by autonomous platform requires it to assess the difference or change of the obtained results in cell space. To achieve this, we proposed a framework for assessing the difference in originality and change of the synthesis results as shown in Figure 6. First, the synthesis process must be repeatable to be valid and exclude any unobserved values in measurement and by the system. Following confirmation of result repeatability, the next step is to check if this result has a precedent. This can be achieved simply by querying a given database containing knowledge of a given subject in a platform memory. If the search confirms that similar observation has been reported, the synthesis can be classified as not new, not contributing added information to our knowledge base. However, if the result has not been observed previously, we need to consider if it could be predicted using all the current knowledge. The predictability implies that this result is not unusual but new to some extent. Unpredictability implies that result obtained is offbeat, for example, a synthesis mechanism that cannot be predicted can be classified as unusual, opening a new set of flow parameters to execute for the platform. Therefore, this framework will enable automatic assessment of the synthesis results by autonomous and digitised robotic platform.

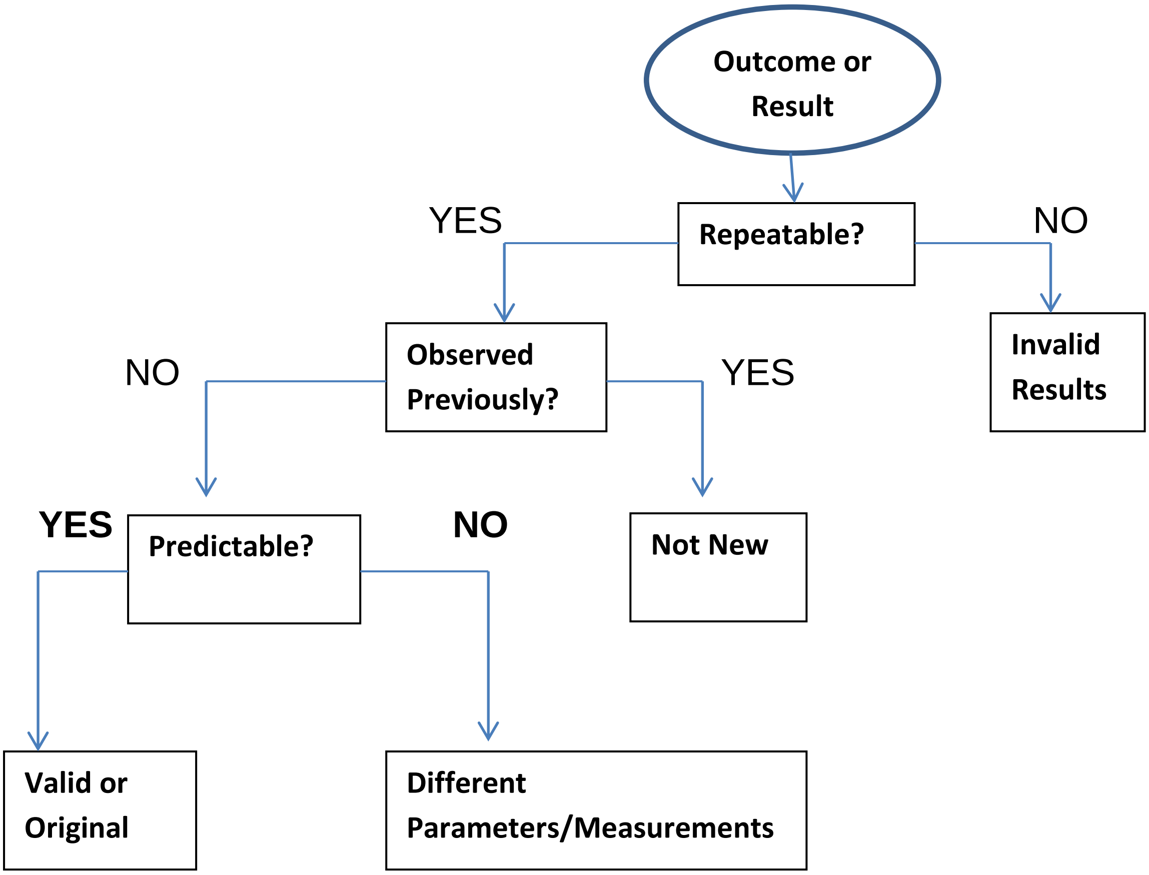


Figure 6. Synthesis Validity Diagram.

Synthesis Simulator

Based on the basic components of our CellSynputer environment, we can use them to form a real synthesis simulator. The first thing a synthesis simulator needs is actual flow, i.e., amino acids(Synthetic and Extracted). Hence, we need to generate this material flow. Once this is done, we can run our simulator to see graphically how it performs.

Generating Flow

We have shown how the CellSynputer platform is defined as well as the dynamics in the CellSynputer. We now need to generate actual flow of materials. In other words, we need to define how input materials arrive in the CellSynputer and the attributes of these materials.

How a material arrives actually refers to when – which time step – and where – which inlet – they arrive in the environment. To do so, we use a new parameter for the CellSynputer, the *flow density*, denoted τ , that serves as the probability, at each time step and for each inlet, that a new material is arriving in this inlet.

This new material's attributes, their rate, desired rate and quantity, and composition, are chosen randomly. Their pressure and temperature is also chosen randomly within a range that depends on the maximum flow rate. Note that the initial rate of flow of a material does not depend on the size of inlet in which they arrive; we do not make assumptions about materials before entering the CellSynputer.

The flow generation process is summarized in below Algorithm.

Algorithm: Generating Flow

```
For all inlet do
    if inlet's initial position is active then
        if random < $\tau$ , with probability $\tau$ then
            materials ← randomly initialized
            materials enters CellSynputer
        end
    end
end
```

CellSynputer Steps

We mentioned earlier that the CellSynputer is updated regularly from the sequence of flow. In other words, materials that are already in the CellSynputer enable to make decisions for the materials flow behind them. Said decisions can then be observed by the other materials that are waiting behind and consequently impact their decisions to flow based on instructions from flow control units. To implement this, we divide a CellSynputer time step into two “sub-steps”, that we call the ***observation step*** where the materials flow observe the environment and choose an action, and the ***update step*** where we update the CellSynputer’s state by executing the materials’ flow as per control unit actions. During the observation step, the materials choose their flow action and accordingly, their flow lines are turned on. After the observation step, all materials flow have chosen their action, and we can now update the CellSynputer: this is the *update step*, and this is how a CellSynputer time step is implemented.

Simulation Parameters

From the above, we can identify the parameters, whether of the CellSynputer or of the materials, that constitute the simulation parameters:

- Number of inlets $X > 0$,
- Size of the inlets $Y > 0$,
- Number of outlets $E > 0$,
- Size of the outlets $S_e > 0$,
- Mix duration $T_m > 0$,
- Flow density τ [0, 1]
- Materials’ maximum rate $v^{max} > 0$,
- Materials’ maximum temperature $t^{max} > 0$,
- Materials’ maximum pressure $p^{max} > 0$

Each of these parameters can be defined arbitrarily. The lower bounds on their values guarantee a minimal working environment. There are no upper bounds, though we expect the simulator to be used with reasonable values.

CellSynputer Performance

At this point, we have defined fully functional synthesis simulator and, since we wanted to make a realistic system, we would like to see how well it performs and mirrors real-world synthesis. First, we have to define what a good performance is; we want to maximize the ratio of amino acids that reach their goal and minimize the number of failed synthesises. Second, we have to run some simulations and see what happens. It is clear that the result of a simulation depends on the chosen parametes. Hence, we focus on what we consider to be the most important ones: the flow of the materials, as it impacts their merging behavior, and the flow density.

We run a series of tests to analyze the percentage of each outcome and the throughput of our synthesis simulator depending on the probability of the flow of materials and the environment. To do that, we fixed the other parameters:

- Number of units = 5
- Size of the units= 10
- Number of exits = 5
- Size of the exits = 10
- Mixing duration = 10
- Materials' maximum rate of flow = 5

The possible outcomes are called *goal* if the concatenated materials reaches their goal, *fail* if they *missed goal* if the flow materials does not reach their goal.

For each probability between 0 and 0.20 , we run the simulation for 40 time steps and repeat it 10 times to compute the average. This is done for two different flow densities: 0.25, and 0.50.

Figure 7 shows the percentage of each outcome – goal, fail or missed goal – depending on the probability for two flow densities. We can see that when the flow density increases, it takes a lower probability value to make the simulator behave poorly. With a lower flow density, there should be more time and space between the materials to consolidate and they consequently have more room to make failures.

The amount of materials that enter the CellSynputer is obviously higher when the flow density is high. However, we can see that when the conditional probability increases, this number tends to decrease.

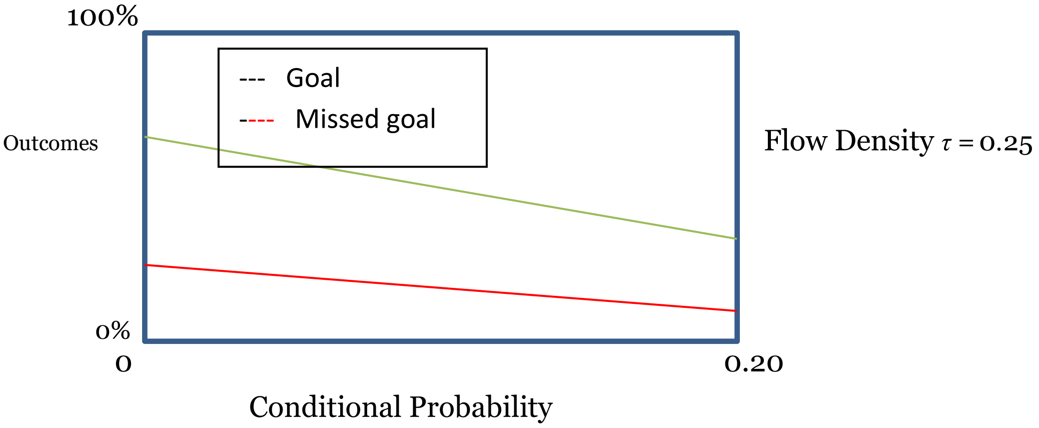


Figure 7: Outcome percentage depending on the conditional probability

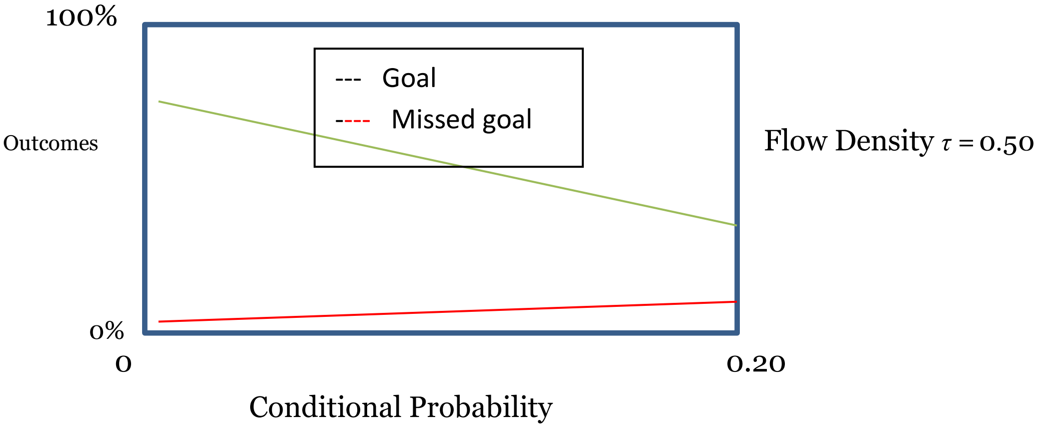


Figure 8: Outcome percentage depending on the conditional probability

ALGORITHM ALONG WITH SIMULATOR

1. **Physical Reality Modelling -** required information
   - The goal of the units/quantum generator
   - What the synthetic robot sees, Finished & Unfinished Synthesis Materials
   - Real Simulation for Synthesis Execution
2. **Task Execution (Simulation)**
   - Generating actual materials(how materials arrive)
   - Initial setup of the CellSynputer environment (control units and goal)
   - Synthesis Execution(Simulation Steps), is updated as the work process progresses in line with the simulator

- Synthesis execution performance, as we have fully functional simulator and to make a realistic system, we would like to see how well it performs and mirrors real world execution(Artificial Intelligence)
- Measuring the outcome with optimization steps after integrating the real world elements with the simulator data.
- Implementation of Graphical Version of the Synthesis Execution

Models for Simulator based Algorithm

Minimum amount of required information

- The current state of the CellSynputer and its environment
- The goal of the synthesis units/quantum generator
- What the synthesis robot sees, Synthesized materials(Finished & Unfinished)

3. Algorithm(Simulator Guided)

```
For  Generating input materials at time step t do
      ( # Encoding observation )
   If  materials in CellSynputer then
       Initialize observation
        If materials in CellSynputer Units then
          Mark the location & update control units
             If materials in synthesis unit  then
                Carry synthesis  as per simulator  process
                  If carried synthesis is as per the goal  then
                     Exit
                  Else  repeat the synthesis task
                  End
             End
        End
     End
   End
```

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. Biological systems contain complex metabolic pathways that make them difficult to predict and Protein synthesis is an example of such a pathway. Here we show how protein synthesis may be improved through a design of simulator for automated synthesizer with a series of iterated high-throughput experiments guided by a machine-learning algorithm. The algorithm predicts fruitful experiments from statistical models of the previous experimental results, combined with stochastic exploration of the experimental space and with synthesis abstraction in-built into the experimental space. The desired experimental conditions were synergistically combined with automation in CellSynputer and discovered to have improved yield when graphically interpreted. In this way, it is possible to script and run desired syntheses in simulator-driven mode where the computer creates low-level instructions for the hardware taking interface representation of the platform and abstraction representing cell synthesis. Although the platform model given us a method of automating cellular assemblies in a framework embodied multi-unit & algorithmic simulator driven system however, this need to be tested using natural crop cells and it could be promising for us in achieving 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.