Quantum Generators: Building a Model that Generate Crop Molecules using Molecular Synthesis and Generative AI

Poondru Prithvinath Reddy

August 22, 2024

Quantum Generators: Building a Model that Generate Crop Molecules using Molecular Synthesis and Generative AI

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 or Protein synthesis contain complex metabolic pathways with many synergies that make them difficult to predict from first principles. Here we show how protein synthesis may be improved by capturing protein structures in quantum generator(a complex process involving various stages like - protein sequencing to generate different crop molecules)from a protein sequence i.e. the amino acids character concealed within protein sequences. It typically consists of stages like target molecules, sequence design, robotic testing, and rigorous trials before final buildup of molecules. However, it is essential to generate correct molecules with close monitoring at each step in a controlled – intensive CellSynputer(a modular platform automating cell synthesis which embodies synthesis abstraction) platform with real-time monitoring and AI is emerging as a game – changer in optimizing quantum generation to individualise molecular profiles. Over here, we try to take advantage of generative AI to produce novel protein molecules with the aim to optimizing the quantum generation process from input to final generation using GenAI and robotic synthesis. For this, we used a different class of AI models – a variant of Autoencoder for more stable training and improved generation quality. With this background, we build a Variational Autoencoder that use simulated structures to learn the structural patterns of molecules and generate new ones as neural networks are used for data reconstruction, and generate new molecular samples by sampling from a learned latent space distribution of input structures. The model converts input structures used as a numerical representation to process data to decode or generate new molecules that comes from the same embedding pattern as the input data, enabling generative molecular design in a quantum generator. and in that respect an implementation of Variational Autoencoder algorithm for managing cell type structures as a part of robotic system based on small model is presented. Although the platform model with cell type intelligence as modular multi-unit based CellSynputer 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

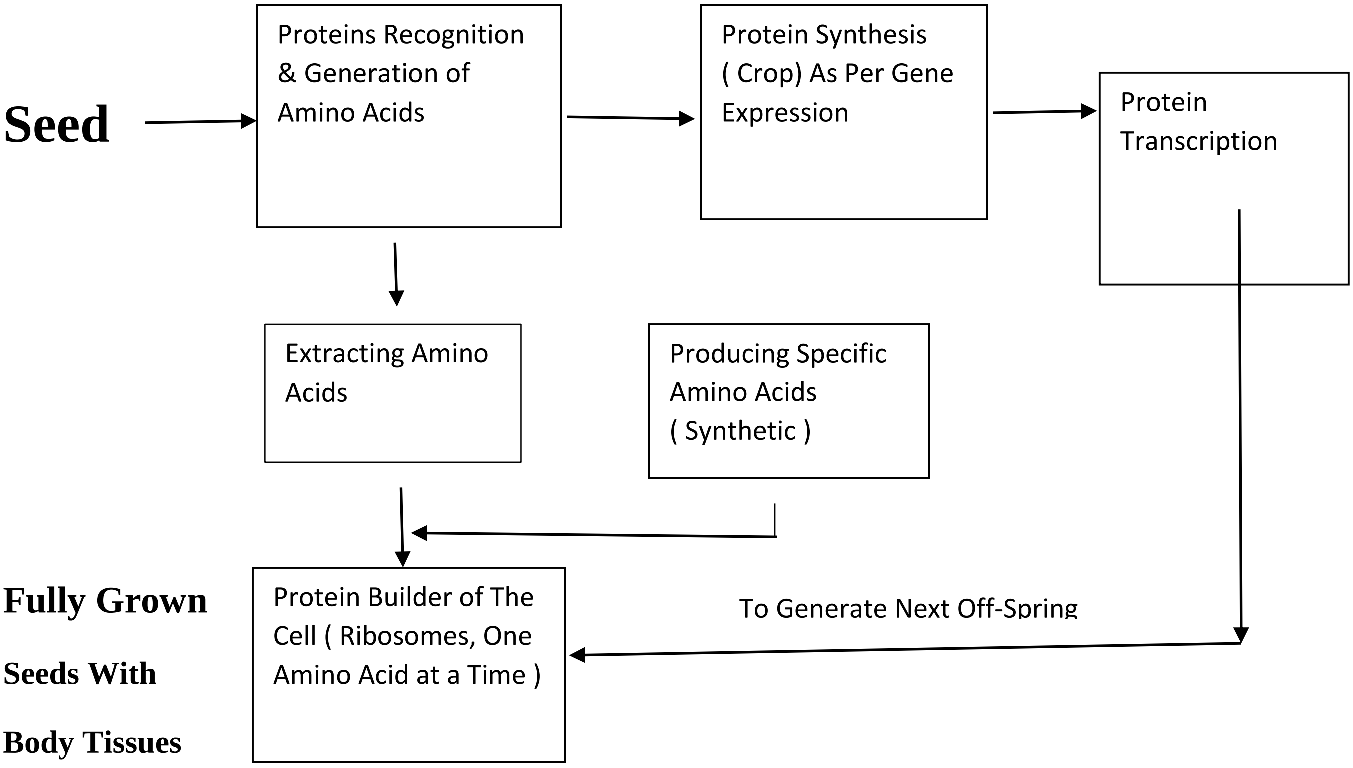

Seed
Proteins Recognition & Generation of Amino Acids
Protein Synthesis (Crop) As Per Gene Expression
Protein Transcription
Extracting Amino Acids
Producing Specific Amino Acids (Synthetic)
Fully Grown
Seeds With
Body Tissues
Protein Builder of The Cell (Ribosomes, One Amino Acid at a Time)
To Generate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

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.

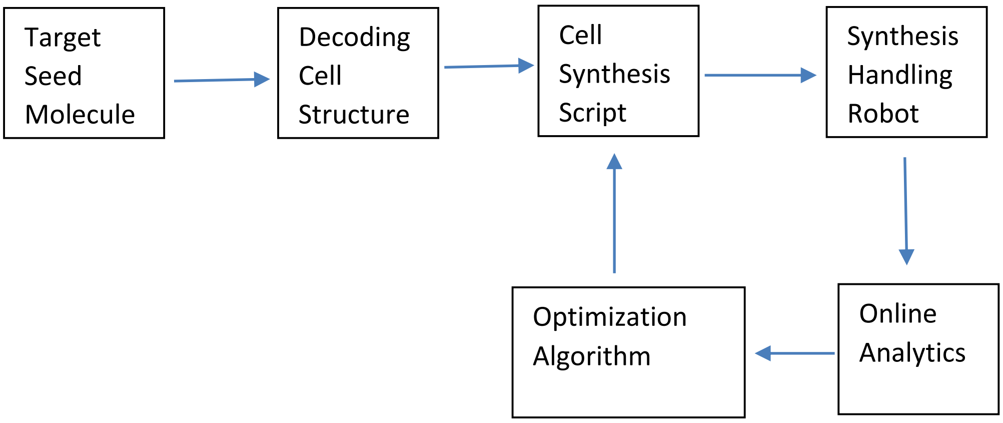


Fig. 2 Architecture of Robotic Synthesis of Crop Cells in a Quantum Generator

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.

AI in Quantum Generation

Quantum Generation is a complex and well-controlled process involving the synthesis, testing, and generation of new molecular structures to generate various crop molecules. It typically consists of stages like target identification, synthesis design, robotic testing, and rigorous optimization trials before final generation. It's a time-controlled and resource-intensive endeavour, with most potential molecular candidates close monitoring to progress at each step.

In recent times, AI emerged as a game-changer in optimizing molecular development. At the initial stages, AI has the ability to sift through vast datasets to identify promising molecular candidates and even designs new molecules, expediting the generation phase. During online testing, AI could predict potential structures, reducing reliance on traditional molecular testing. In the initial trial phase, AI aids in molecular requirement by analyzing structures in real-time monitoring system and optimizes trial molecular design for increased success rates.

Generative AI for Molecular Synthesis

Generative AI: **GenAI generates new knowledge or content that is similar to its training data**. Algorithms are trained on large datasets of existing crop molecules that learn to identify patterns and relationships common to these molecules.

Generative AI models can take inputs such as **text, image, audio, video, and code** and generate new content into any of the modalities mentioned.

Generative models can be **used to generate images of biological structures or processes based on textual descriptions**, which can be helpful in visualizing complex phenomena or generating data for hypothesis testing. **The model converts chemical structures into useful numerical representations to process data more efficiently with AI**. A molecule's numerical representation can be used as input to “decode,” or generate, molecules with desired properties, enabling generative molecular design.

There are two main types of generative AI models. The most used generative models are **Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs)**, and also autoregressive models. Each of these models has advantages and disadvantages, depending on the complexity and quality of the data.

Generative AI utilizes **deep learning, neural networks, and machine learning** techniques to enable computers to produce content that closely resembles human-created output autonomously. These algorithms learn from patterns, trends, and relationships within the training data to generate coherent and meaningful content

Variational AutoEncoders

Autoencoders have emerged as an architecture for data representation and generation. Among them, Variational Autoencoders (VAEs) stand out, introducing probabilistic encoding and opening new avenues for diverse applications.

Autoencoders are neural network architectures that are intended for the compression and reconstruction of data. It consists of an encoder and a decoder; these networks are learning a simple representation of the input data. Reconstruction loss ensures a close match of output with input, which is the basis for understanding more advanced architectures such as VAEs. The encoder aims to learn efficient data encoding from the dataset and pass it into a latent space. The other part of the autoencoder is a decoder that uses latent space to regenerate images similar to the dataset. These results backpropagate the neural network in the form of the loss function.

A variational autoencoder (VAE) provides a probabilistic manner for describing an observation in latent space. Thus, rather than building an encoder that outputs a single value to describe each latent state attribute, the encoder is formulated to describe a probability distribution for each latent attribute. It has many applications, such as data compression, synthetic data creation, etc. Variational autoencoder is different from an autoencoder in a way that it provides a statistical manner for describing the samples of the dataset in latent space. Therefore, in the variational autoencoder, the encoder outputs a probability distribution in the latent space layer instead of a single output value.

Architecture of Variational Autoencoder

- The encoder-decoder architecture lies at the heart of Variational Autoencoders (VAEs). The encoder network takes raw input data and transforms it into a probability distribution within the latent space.
- The latent code generated by the encoder is a probabilistic encoding, allowing the VAE to express not just a single point in the latent space but a distribution.
- The decoder network takes a sampled point from the latent distribution and reconstructs it back into output data space. During training, the model refines both the encoder and decoder parameters to minimize the reconstruction loss – the disparity between the input data and the decoded output.
- The process involves a delicate balance between two essential components: the reconstruction loss and the regularization term, often represented by the KL divergence. The reconstruction loss makes the model to accurately reconstruct the input, while the regularization term encourages the latent space to adhere to the chosen distribution.
- By iteratively adjusting these parameters during training, the VAE learns to encode input data into a meaningful latent space representation. This optimized latent code encapsulates the underlying features and structures of the data, facilitating precise reconstruction. The probabilistic nature of the latent space also enables the generation of novel samples by drawing random points from the distribution.

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 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 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.

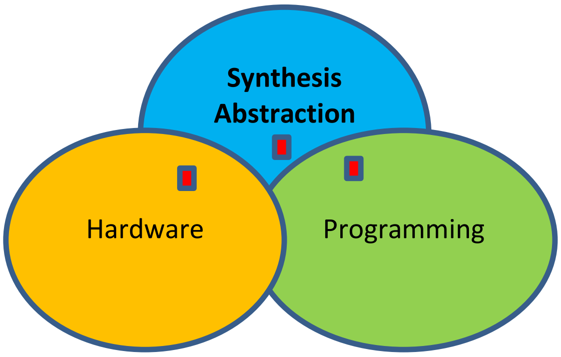


Fig. 3 Approach – Cell Synthesis Automation

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.

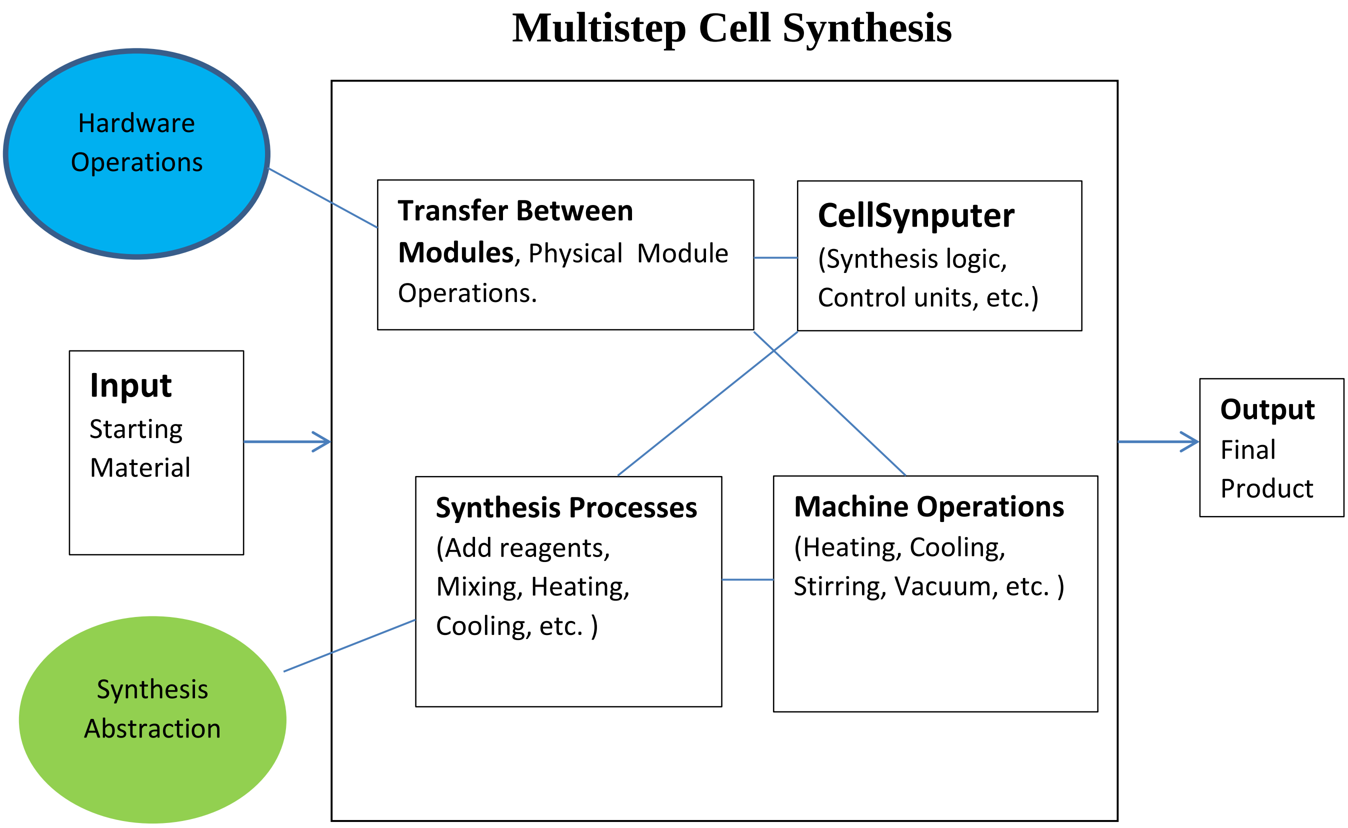

Multistep Cell Synthesis
Hardware Operations
Input
Starting Material
Synthesis Abstraction
Transfer Between Modules, Physical Module Operations.
CellSynputer
(Synthesis logic, Control units, etc.)
Synthesis Processes
(Add reagents, Mixing, Heating, Cooling, etc.)
Machine Operations
(Heating, Cooling,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

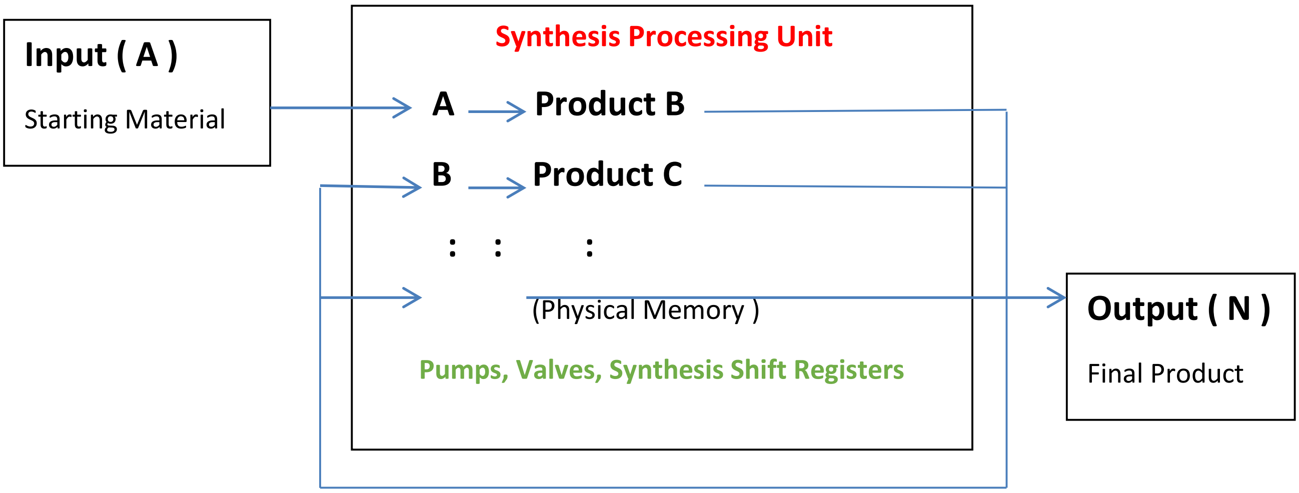
The following are the attributes of parallel synthesizer:

- **Based on multi-unit concept**
- **Configurable at unit level**

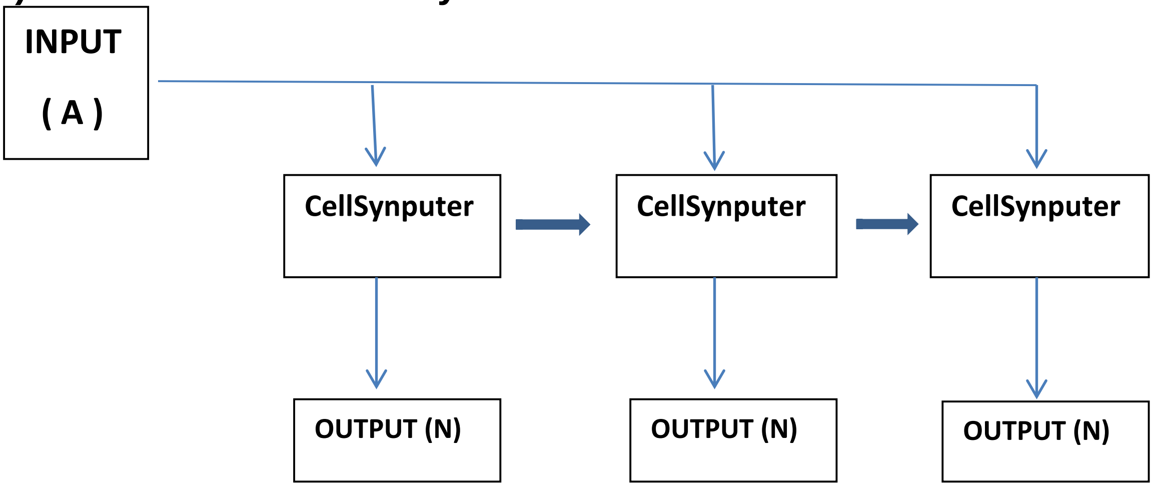
- High throughput
- Small scale at unit level
- Limited to individual synthesis scope
- Embodies multistep procedure

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

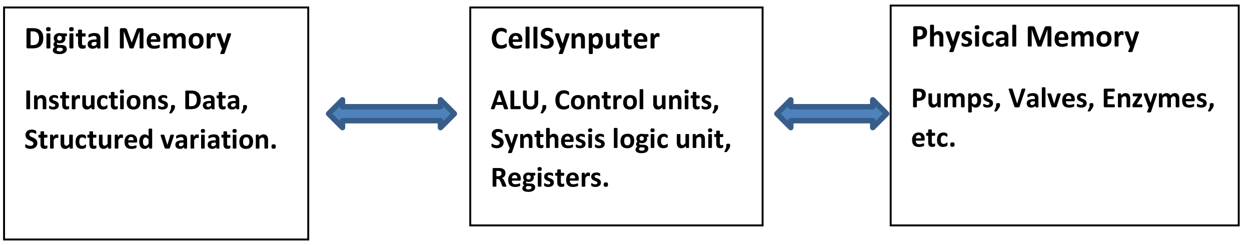
A) N-Step Cell Synthesis



B) Multi-unit Synthesis



C) CellSynputer Architecture


Digital Memory
Instructions, Data, Structured variation.
CellSynputer
ALU, Control units, Synthesis logic unit, Registers.
Physical Memory
Pumps, Valves, Enzymes, etc.

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.

Implementation of Variational Autoencoder

Variational Autoencoders (VAEs) have one unique property that separates them from usual autoencoders, and this property that makes them useful for generative modeling: their latent spaces are continuous, allowing easy random sampling and interpolation. It achieves this by making its encoder not output an encoding vector rather, outputting two vectors of differing parameters. This generation of encoding vectors means, that even for the same input, the actual encoding will somewhat vary on every single pass.

Out of the two, one vector controls where the encoding of an input should be centered around, while the other controls the area, how much from the centered the encoding can vary. As encodings are generated at random from anywhere inside the distribution, the decoder learns that not only is a single point in latent space referring to a sample of that class, but all nearby points refer to the same as well. This allows the decoder to not just decode single, specific encodings in the latent space, but ones that slightly vary.

The model is exposed to a certain degree of local variation by varying the encoding of one sample, resulting in smooth latent spaces on a local scale, that is, for similar samples. However, since there are no limits on what values vectors can take on, the encoder can learn to generate very different vector for different classes, clustering them apart, and minimize deviations, while making sure the encodings themselves don’t vary much. This allows the decoder to efficiently reconstruct the training data.

Therefore, the encodings, all of which are as close as possible to each other while still being distinct, allowing smooth interpolation, and enabling the construction of new samples.

In order to force this, we introduce the loss function i.e the measure between two probability distributions simply measures how much they depart from each other. Minimizing the reconstruction loss here means optimizing the probability distribution of parameters to closely resemble that of the target distribution. So, this loss encourages the encoder to distribute all encodings (for all inputs), evenly around the center of the latent space.

Now, using purely the measured loss results in a latent space encodings densely placed randomly, near the center of the latent space, with little regard for similarity among nearby encodings. The decoder finds it impossible to decode anything meaningful from this space.

In all, optimizing using reconstruction loss results in the equilibrium reached by the cluster-forming nature of the reconstruction loss, forming distinct clusters the decoder can decode. This means when randomly generating, if you sample a vector from the same prior distribution of the encoded vectors, the decoder will successfully decode it. And in interpolating, there are no sudden gaps between clusters, but a smooth mix of features.

In this implementation, we will be using the simulated dataset of images, in absence of any real molecular dataset and this dataset is need to add or upload manually for training the model.

1. First, we need to import the necessary packages to our computing environment.

2. **Creating a Latent Layer (** For variational autoencoders, we need to define the architecture of two parts encoder and decoder but first, we need to define the latent layer - it uses the output from two dense layers as input and convert them into normal distribution and pass them to the decoder layer)
3. **Define Encoder Block (**We define the architecture of encoder part of our autoencoder, this part takes images as input and encodes their representation in the Latent layer.)
4. **Define Decoder Block(**We define the architecture of decoder part of our autoencoder, this part takes the output of the Latent layer as input and output an image)
5. Define the VAE Model (In this step, we combine the model and define the training procedure with loss functions.)
6. Train the VAE (Now it’s the right time to train our variational autoencoder model,. But first we need to import the simulated dataset of images.)

7. Output - Display Sampled Images (In this step, we display training results, we will be displaying these results according to their values in latent space vectors.)
8. Display Latent Space Clusters (To get a more clear view of our representational latent vectors values generated from the encoder, we will be plotting the scatter plot of training data on the basis of their values.)

RESULTS

We systematically run Variational Autoencoder(VAE) algorithm showcasing the encoder, decoder and latent layers culminating in the creation of the VAE model using the simulated image dataset utilizing deep learning models. The implementation included autoencoders as neural networks for data compression and reconstruction paving the way for generation of synthetic image structures. After running the algorithm and examining the output in terms of training results, including the loss values and a scatter plot for latent space visualizations providing insights into the models performance.

As encodings are generated at random from anywhere as a distribution, the decoder learns to not just decode single, specific encodings in the latent space, but ones that slightly vary too, as the decoder is exposed to a range of variations of the encoding of the same input during training.

However, the encoder learns to generate very different vector for different classes, verifying the encodings themselves don’t vary much for the same sample. It is observed the encodings, all of which are close to each other while still being distinct, enabling the construction of new samples as we introduced the loss function and minimizing the loss resulted in probability distribution parameters to closely resemble that of the target distribution.

Similarly, optimizing reconstruction loss forming distinct clusters the decoder can decode. This means when randomly generating, a sample vector from the same prior distribution of the encoded vectors, the decoder will successfully decode it and this feature is important from the point of view of quantum generation as it enable generation of crop lookalike molecules.

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 or Protein synthesis contain complex metabolic pathways that make them difficult to predict. Here we have shown how protein synthesis may be improved by capturing protein structures from a protein sequence and specifically to predict the amino acids character concealed within protein sequences. Over here, we try to take advantage of generative AI to produce novel protein molecules with the aim to optimizing the quantum generation process from input to final generation using GenAI and robotic synthesis. For this, we used different class of AI models – a variant of Autoencoder for more stable training and improved generation quality. With this background, we build a Variational Autoencoder that use simulated patterns to learn the structure of molecules and generate new ones. The model converted input structures used as a numerical representation to process data to decode or generate molecules with desired properties, enabling generative molecular design in a quantum generator. and in that respect an implementation of Variational Autoencoder algorithm for managing cell type structures as a part of robotic system based on small model is presented. The Information about the structure being modelled develops within the network of autoencoders by sampling from a learned latent space distribution.as neural networks are used for data reconstruction, and.the protein structure is generated from the distribution pattern of input data activated inside the network and the desired conditions and patterns were synergistically combined with automation in CellSynputer(a modular platform automating cell synthesis which embodies synthesis abstraction) and successfully generated random molecular samples for the same input. The results of which, including loss values and latent space visualization, were reviewed to interpret the model's performance. Although the platform model given us a method of automating cellular assemblies in a framework embodied multi-unit and algorithmic 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.