# BioSyn - A High Level Language For Molecular Synthesis

## Introduction:

While genetic engineering primarily involves the design, modification and synthesis of individual genes, synthetic biology is an emerging discipline, centered on constructing entire systems of genes and gene products. It involves designing and building new biological systems by assembling molecular and genetic parts, with the purpose of adding to or modifying the biological functions of existing organisms or creating new organisms with specific properties.

In a manner similar to the way electrical engineers assemble complex integrated circuits from transistors, synthetic biology envisions assembly of complex biological devices and networks by assembling individual biological parts. And in a manner similar to how electrical engineers use high level languages such as Verilog and VHDL, the development of a high level language BioSyn is proposed, for the purpose of designing biological and genetic assemblies.

Standard biology parts have been catalogued by various parts registries, notable among them being the MIT parts registry (see references below). Assembly standards have been documented by the BioBricks Foundation and made available as RFCs. All parts that are part of an assembly need to comply with one of the assembly standards. Parts compliance data is available in the parts registries. The genetic sequence associated with a part is also published by the parts registries.

It is proposed that the biological assembly is put together using high level programming constructs.

## Language Constructs:

The language constructs needed to construct an assembly from biological parts are informally described below, together with example usage of the same. In addition to the ability to build assemblies from standard parts, it is proposed that the language includes built in functions to print the genetic sequence, print schematic diagrams, as well as generate XML markup for use with external simulation software.

Primitive types: int, float, char, boolean and string as in C, with the exception that character strings are included as built-in types.

Arrays: arrays of the primitive types as in C

Operators :

arithmetic : +, -, *, /
string : + // string concatenation as in Java
boolean : &&, || and !
Comparison : >, <, ==, >=, <=, !=
assignment : = // assignment to variables representing primitives, objects and object properties

Complex types: These include types representing biological parts, attributes of parts eg. part sequence, as well as composites derived from parts (assemblies ). Prototypes for each of these are defined prior to instantiation as illustrated in the examples below.

```
Attribute Sequence string;          // The sequence property is declared as a string attribute
Attribute Name string;              // The BioBricks name property
Attribute Compatibility string;     // Represents BioBricks RFC compatibility
Attribute Strength float;

// define the prototype for promoters
Part Promoter(Name, Sequence, Compatibility, Strength);

// define the prototype for an RBS
Part RBS(Name, Sequence, Compatibility);

// instantiate the BioBricks promoter BBa_I14018
Promoter Bba_I14018("BBa_I14018", "tgtaagtttatacataggcgagtactctgttatgg", "RFC11", 0.5);

// instantiate the BioBricks RBS Bba_J63003
RBS Bba_J63003("BBa_J63003", "cccgccgccaccatggag", "RFC21");

// create an assembly represented by a composite; composites may contain other composites
Composite Assembly1( Bba_I14018,  Bba_J63003);
```

Conditional statements: if-else as in C

Loops: for and while, as in C, can be used for iterating through arrays

Constraints: Composites can be parsed and validated against constraints

```
// examples of constraints
Start → <PlasmidBackbone><Prefix><Cassette><Suffix>
Cassette → <Promoter><Cistron><Terminator>
Cistron → <RBS><Gene>
```

User defined functions: Global functions as in C

Library functions: Functions to validate assemblies, print gene sequences, print diagrams and generate markup

```
// examples of function usage
if (Assembly1.validate()) {
        Assembly1.printSequence();
        Assembly1.printDiagram();
        Assembly1.generateMarkup("/usr/local/home/user1/assembly1markup.xml");
}
```

**Sample program:**

```
// declare all attributes that can be properties of standard parts

Attribute Sequence string;          // The sequence property is declared as a string attribute
Attribute Name string;              // The BioBricks name property
Attribute Compatibility string;     // Represents BioBricks RFC compatibility
Attribute Strength float;

// part prototypes are declared prior to instantiation, each part uses one or more attributes
// technical definitions for the parts can be found in the glossary

// define the prototype for promoters
Part Promoter(Name, Sequence, Compatibility, Strength);

// define the prototype for an RBS
Part RBS(Name, Sequence, Compatibility);

// define the prototype for a terminator
Part Terminator(Name, Sequence, Compatibility);

// define the prototype for a cistron
Part Cistron(Name, Sequence, Strength);

// define the prototype for a cassette
Part Cassette(Name, Sequence, Compatibility);

// define the prototype for a gene
Part Gene(Name, Sequence);

// define the prototype for a prefix
Part Prefix(Name);

// define the prototype for a suffix
Part Suffix(Name);

// instantiate the BioBricks promoter BBa_I14018
Promoter Bba_I14018("BBa_I14018", "tgtaagtttatacataggcgagtactctgttatgg", "RFC21", 0.5);

// instantiate the BioBricks RBS Bba_J63003
RBS Bba_J63003("BBa_J63003", "cccgccgccaccatggag", "RFC21");

// instantiate the BioBricks terminator BBa_B1002
Terminator Bba_B1002("BBa_B1002", "cgcaaaaaaccccgcttcggcggggttttttcgc", "RFC21");

// declare the constraints against which the assembly is parsed and validated
Start → <PlasmidBackbone><Prefix><Cassette><Suffix>;
Cassette → <Promoter><Cistron><Terminator>;
```

```
Cistron → <RBS><Gene>;
Cistron → <Cistron><Cistron>;
Terminator → <Terminator><Terminator>;
Gene → <Gene><Gene>;

// create an assembly,  represented by a composite; composites may contain other composites
Composite Assembly1( Bba_I14018,  Bba_J63003, Bba_B1002);

// validate the assembly against the declared constraints and then print sequence, print diagram and
// generate markup

if (Assembly1.validate()) {
        Assembly1.printSequence();
        Assembly1.printDiagram();
        Assembly1.generateMarkup("/usr/local/home/user1/assembly1markup.xml");
}
```

**Glossary:**

(adapted from definitions at the MIT Biological Parts Registry and the medical dictionary at freedictionary.com as well as http://openwetware.org/wiki/Synthetic_Biology_and_the_High_School_Curriculum:_Glossary)

- **Cloning** -- recombinant DNA molecules inserted into a plasmid or virus "vector." The vector must then be introduced into a host cell without killing it
- **Device** -- an engineered genetic object that produces a human-defined function under specified conditions. Devices are produced by combining one or more standard biological parts
- **DNA Synthesis** -- chemical assembly of nucleotides in a specified order
- **Gel electrophoresis** -- the use of current to draw a polymer (like DNA or proteins) through a sieving matrix, separating the polymers by size. Most often agarose is the matrix used for DNA electrophoresis, and polyacrylamide is the matrix used for proteins
- **Gene** - A hereditary unit consisting of a sequence of DNA that occupies a specific location on a chromosome and determines a particular characteristic in an organism. Genes undergo mutation when their DNA sequence changes.
- **iGEM** -- the international Genetically Engineered Machine competition in which teams of undergraduates build living systems from standardized, biological parts
- **Inverter** -- takes an input signal and produces the opposite output signal, e.g., HIGH input produces LOW output and vice versa. An inverter functions like a Boolean NOT
- **Measurement** -- the quantitative assessment of a biological function. Measurements can be made of a part, device or system
- **Open Reading Frame** -- the DNA pattern of triplet sequences that encode a protein
- **Part** -- a nucleic acid-encoded biological function
- **PCR** -- a technique for amplifying DNA of known or unknown sequence. The reactions require only 4 components: DNA to be amplified, oligonucleotide primers to bind sequences flanking the target, dNTPs to polymerize into new DNA chains, and a heat stable polymerase in a buffered solution to carry out the synthesis reaction
- **Plasmid** -- a circular, double-stranded DNA molecule typically containing a few thousand base

pairs that replicates within a cell independently of the chromosomal DNA. Plasmid DNA is easily purified from cells, manipulated using common lab techniques and incorporated into cells

- **Plasmid backbones**: A plasmid is a circular, double-stranded DNA molecules typically containing a few thousand base pairs that replicate within the cell independently of the chromosomal DNA. A plasmid backbone is defined as the plasmid sequence beginning with the BioBrick suffix, including the replication origin and antibiotic resistance marker, and ending with the BioBrick prefix.
- **Promoter** -- sequence of DNA to which RNA polymerase binds for initiation of transcription
- **Protein domains**: Protein domains are portions of proteins cloned in frame with other proteins domains to make up a protein coding sequence. Some protein domains might change the protein's location, alter its degradation rate, target the protein for cleavage, or enable it to be readily purified.
- **Protein coding sequences**: Protein coding sequences encode the amino acid sequence of a particular protein. Note that some protein coding sequences only encode a protein domain or half a protein. Others encode a full-length protein from start codon to stop codon. Coding sequences for gene expression reporters such as LacZ and GFP are also included here.
- **Restriction Enzyme** -- an enzyme that recognizes and cleaves a specific DNA sequence
- **Ribosome Binding Site** -- the sequence of RNA to which ribosome binds for initiation of translation
- **Standardization** -- a series of assembly and characterization rules. In time, these standards may allow the reliable physical and functional assembly of genetic parts into devices, and devices into systems
- **Transcription** -- the reaction that converts of DNA-templated information to RNA. This reaction is catalyzed by one of several RNA polymerases
- **Transcriptional Terminator** -- a sequence of DNA that signals the RNA polymerase to cease the synthesis of RNA. Terminator sequences are often inverted repeats in the DNA that fold into stem-loop structures, leading the RNA polymerase to pause and leave the DNA it is transcribing
- **Translation** -- the reaction that converts RNA-templated information to protein. This reaction is catalyzed by ribosomes

**References:**


1) Wikipedia on Artificial Gene Synthesis - http://en.wikipedia.org/wiki/Gene_synthesis

2) Chapter on high level programming languages for biomolecular systems by Jacob Beal et al. in Design and Analysis of Biomolecular Circuits – Springer 2011

3) Targeted development of registries of biological parts – Jean Peccoud et al. PLoS ONE, 3, e2671 2008

4) A syntactic model to design and verify synthetic genetic constructs derived from standard biological parts. Yizhi Cai et al. Bioinformatics, 23, 2008

5) MIT Registry of Standard Biological Parts - http://partsregistry.org/Main_Page

6) COPASI Biochemical Network Simulator - http://www.copasi.org/tiki-view_articles.php

7) COPASI file format (CopasiML) - http://www.copasi.org/tiki-index.php?page_ref_id=128

8) The Systems Biology Markup Language - http://sbml.org/Main_Page

9) BioBricks RFC's - http://biobricks.org/programs/technical-standards-framework/

10) Berkeley introduction to Synthetic Biology - Bio Building Basics: A Conceptual Instruction Manual for Synthetic Biology