Automatic restriction strategy finder for Synthetic Biology Constructs

Jorge Gomes[1,2]

[1] University of Minho, School of Engineering
Jorge.gomes12@gmail.com
[2]Campus of Gualtar, 4710-057, Braga, Portugal

1 Introduction

DNA, as any other form of data, can be both read and written. For DNA, the process of reading is something known as DNA sequencing, whereas writing is achieved by gene synthesis. Over the last decade the main concern of molecular biology has been set on reading and analyzing naturally occurring DNA sequences, as revealed by massive sequencing efforts worldwide, on a very wide range of organisms. In contrast, the emerging field of Synthetic Biology aims to write new genetic informatic, thereby creating designed, non-natural genes, proteins, biological processes, and even organisms [17].

Gene synthesis was conceived as a means of gene acquisition in the 1970s and early 1980s [1,11], meant to be applied on living systems, as recombinant DNA technologies allowed biologist to deliberately change the molecular structure of the organisms, and the chemical synthesis of DNA became widely available [15]. However, this approach would be soon overtaken by cloning libraries and PCR in the following years. More recently, protein and DNA sequences have become much easier to obtain electronically from databases than physically from library clones. In the meantime, gene synthesis technology, has matured, allowing direct gene synthesis to become the most efficient way of producing functional genetic constructs, enabling a variety of applications from codon optimization to protein engineering [17].

Synthetic Biology has established itself in the recent years, aided by the convergence of molecular biology and the engineering principles, underpinned by the massive advances in biological technologies [7], that have allowed the creation of full length genes, operons and even genomes *de novo* [17]. In the last 40 years, the length of synthetic genes that can be obtained in a laboratory has increased by four orders of magnitude, ranging from 10^2 to 10^6 base pairs [16]. In 2010 it was estimated that at this rate, the synthesis of a genome as complex as the human (10^9 base pairs) could be feasible in 10 years' time [7]. In the same year Gibson and his coworkers managed to create a fully functional bacterial cell controlled only by its 1.08 mega base pair chemically synthetized genome [5]. Further developments among bacterial genomes have been made, however eukaryotic genomes present additional challenges, as the genome is much larger and much more complex. Another 4 years later another landmark would be achieved, the first eukaryotic chromossome was synthetized for *Saccharomyces cerevisiae*, a fully functional copy of its 316,617-base pair chromossome III [2]. In fact, the synthesis of the Human Genome might not be that far away, since scientists have

already announced the next take on the Human Genome Project - HGP-Write – a 10-year project which aims to completely synthetize the human genome [3]. One of the facts that supports this project is the continuous downfall of the price for synthetizing genes, with prices coming as low as 0.01 $ per base pair [7].

Synthetic biology holds now a tremendous potential as both an investigative and therapeutic modality, since it can represent a major breakthrough in some of humanity biggest concerns', namely, the production of environmentally friendly biofuels, or inexpensive pharmaceutical drugs [7, 18].

For those goals, Synthetic Biology makes the best use for a variety of microorganisms and plants, which have evolved to produce a myriad array of complex molecules known as natural products or secondary metabolites that are of biomedical and biotechnological importance [8]. The ability to synthetize said molecules requires prior knowledge on a species' genome and metagenome, which represents a rich source for discovery of novel pathways involved in natural product biosynthesis [13]. This is commonly referred as pathway or metabolic engineering, a process where complete biosynthetic pathways are often transferred from native hosts to heterologous organisms in order to obtain products as referred before, but with higher yields than in a non-engineered organism. Consequently, gene expression needs to be balanced, promoter strength needs to be tuned and the endogenous regulatory network needs to be modified [14]. One of two approaches can be chosen here, the combinatorial one, where genes are rearranged in a construct to generate small molecules, or the synthetic way, whereas complex combination of genetic elements are used to create new circuits, with designed properties. In any of these approaches, the conventional multi-step, sequential-cloning method including primer design, PCR amplification, restriction digestion, *in vitro* assembly and transformation, is typically involved and multiple plasmids are often required. This method however can be time consuming and relies on unique restriction sites that become limited for large recombinant DNA molecules [14]. The lack of facile, highly efficient manipulation techniques for libraries of interchangeable genetic elements has stood as a significant hurdle for biosynthetic constructs. As this necessity arose, efforts have been made to develop revolutionary techniques, to transform arduous constructions into routine tasks [4].

Modern DNA assembly techniques can be classified into two groups: those based on homology, and those based on ligation. Homology based methods require neighboring DNA fragments to share identical sequences, such that splicing can occur either by annealing by either annealing followed by extension of the homologous ends *in vitro* or by homologous recombination *in vivo,* a method known as overlap-directed assembly. The most distinguishable *in vitro* technique is probably the one proposed by Gibson and his coworkers, colloquially know as Gibson Assembly [6], which, as stated above, made possible the assembly of a fully functional bacterial cell. Worth mentioning is also the Shao 'DNA assembler' method which enables design and rapid construction of large biochemical pathways in one-step fashion by exploitation of *in vivo* homologous recombination mechanism in *S. cerevisiae*. Due to its high efficiency and ease to work with, *in vivo* homologous recombination in yeast has been widely used for gene cloning, plasmid construction and library creation in the past, yet yeast *in vivo* homol-

ogous recombination was only successfully used to assemble multiple-gene biochemical pathways in a plasmid, by Shao and his co-workers in 2009 [13]. A more recent example of the same effort is the work published by Mitchell and his colleagues who managed to develop a method for assembling genetic expression pathways for expression in *Saccharomyces cerevisiae*¸ which takes advantage of the organism capacity to perform homologous recombination, efficiently joining sequences with terminal homologies. The versatile genetic assembly system (VEGAS) uses a modified version of the Golden Gate cloning method, in which each Transcription Unit is assigned a pair of VEGAS adapters that assemble up and down stream, providing terminal homology for overlap-directed assembly by homologous recombination *in vivo.* Through this approach, it was possible to assemble the β-carotene and the violacein biosynthetic pathways [10].

While in the past, small DNA constructs incorporating few parts were common, the complexity of new constructs has grown with advancing technology. Techniques like the ones mentioned earlier are commonly referred to as "next generation cloning", since these protocols describe the assembly of ten to twenty PCR fragments into a complex DNA construct. The complexity of these strategies results in high risks of cloning errors and omissions, and without proper documenting and *in silico* simulation these strategies are not fully reproducible by other laboratories [12]. A possible solution to these problems is a strategy description that is both readable by humans and executable by a computer to simulate the individual steps of the protocol as well as the result. To assess this necessity several efforts were driven by the bioinformatic community [12]. Here I will be discussing Pydna, as it is the platform where my project will be developed upon.

Pydna is software tool that provides high level computer simulation of DNA manipulation procedures and aid the design of complex constructs. It allows automated primer design for homologous recombination cloning or Gibson assembly, as well as a simulator of DNA assembly. This software package was implement exclusively in Python and makes uses of the Biopython and NetworkX packages. Most of the Pydna functionalities are implemented as methods for the Dseqrecord class, designed to hold all the sequence information necessary for describing a double-stranded DNA molecule. Dseqrecord objects reflect much of the functionality of SeqRecord objects from Biopython. Even though Pydna works through a command line interface, the code semantically resembles the molecular biology unit operations, making it easy to read even for non-programmers. An example using this package is presented in Figure 1, where Pydna executes the assembly of a circular molecule from 3 PCR products also obtained from Pydna *in silico* simulation. Pydna also supports integration with IPython notebook, a format where figures, code and text can be combined, allowing for the creation of effective workflow description for complex constructs [12].

```
In [19]: asm = pydna.Assembly((yep_bgl,cycl_prd, gfp_prd))
In [20]: cnt = asm.circular_products[0]
In [21]: cnt
Out[21]: Contig(o10681)
In [22]: cnt.figure()
Out[23]:
```

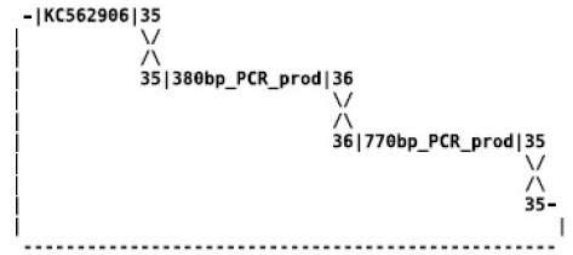

Fig. 1: The above code executes the assembly of a circular molecule of DNA based on previously obtained PCR products (yep_bgl, cycl_prd, gfp_prd), and prints in a text based figure the way the sequences were assembled. The input in [19] assembles the fragments, in several different recombined constructs, both linear and circular, whereas the code input in [20] accesses only the first circular product of said assembly. Assemblies are stored as Contig objects which can be represented with the **figure** method.

1.1 Problem definition and objectives

As we've seen there are many new cloning strategies by which very large and complex genetic constructs can be synthesized. The size and complexity means that verification of the construct becomes a challenge. One of the fastest and most robust ways to analyze DNA molecule structure continues to be restriction analysis with a set of carefully chosen restriction enzymes. Restriction enzymes are incontestably the most fundamental tool used in molecular biology, and over the past years, more than 3500 restriction have been isolated [9]. However, because of the large number of restriction enzymes now available commercially, it is becoming truly difficult to navigate through an ever-increasing number of choices, to find the enzyme or enzymes that we actually need. Restriction enzymes used for analysis of genetic constructs are normally chosen ad-hoc and this choice becomes both more difficult and critical with the increasing number of available enzymes, and the increasing length and complexity of the constructs. This method can be both time and resource consuming, since it's commonly based on a trial and error approach.

As we discussed previously, most constructs are based on homologous overlapping sequences, so one way of analyzing those constructs is to assess where those homologies are located, and for that we need to digest them. That requires a restriction enzyme set that can cut through all the sequences of the construct, and differentiate them based on the digestion products. Many of these DNA molecules are plasmids, whose nucleotides are arranged in a circular-fashion. Therefore, the search for restriction sites within a plasmid constitutes a circular string pattern matching problem, which requires a string to be rotated several times in order to find a match. However, that problem is already

addressed by Pydna Dseqrecord objects' which support circular sequences, and by Biopython restriction algorithms'.

The main goal of this project is to design an algorithm for the automatic selection of the most effective restriction enzymes for DNA analysis based on user defined criteria. The algorithm will then be implemented in Python and embedded within Pydna software package.

2 Methodology

The algorithm implementation will be based on the following pseudo-code:

Inputs: A list of Dseqrecord objects representing each DNA molecule to be analyzed, the minimum fragment size that can result from digestion and a list of enzymes that are available to search within.

Output: A list of enzymes that can cut through all the presented DNA molecules, and respect the user defined criteria.

Procedure:

```
0 Set result = []
1 Set n = 1
2 b = Get the longest contiguous sequence between the list of sequences
3 While result is empty or n < 3:
    3.1 For each enzyme in the list:
        3.2 If enzyme cuts n times in b:
            3.3 If enzyme cuts n+1 or more times in all the elements of the list:
                3.4 If all resulting fragments > defined minimum fragment size:
                    3.5 Store enzyme in result
    3.6 Set n = n + 1
4 If result is empty:
    4.1 Get all possible combinations of two enzymes in enzyme list
        4.2 For each enzyme pair:
            4.3 If enzyme pair cuts once in b:
                4.4 If enzyme pair cuts twice or more in all the elements of the list:
                    4.5 If all resulting fragments > defined minimum fragment size:
                        4.6 Store enzyme pair in result
5 Stop
```

References

1. Agarwal, K. L., Büchi, H., Caruthers, M. H., Gupta, N., Khorana, H. G., Kleppe, K., ... & Sgaramella, V. Total synthesis of the gene for an alanine transfer ribonucleic acid from yeast. Nature, 227, 27-34 (1970).
2. Annaluru, N., Muller, H., Mitchell, L. A., Ramalingam, S., Stracquadanio, G., Richardson, S. M., ... & Cai, Y. Total synthesis of a functional designer eukaryotic chromosome. Science, 344(6179), 55-58 (2014).
3. Boeke, J. D., Church, G., Hessel, A., Kelley, N. J., Arkin, A., Cai, Y., ... & Isaacs, F. J. The genome project-write. Science, 353(6295), 126-127 (2016)
4. Cobb, R. E., Ning, J. C., & Zhao, H. DNA assembly techniques for next-generation combinatorial biosynthesis of natural products. Journal of industrial microbiology & biotechnology, 41(2), 469-477 (2014).
5. Gibson, D. G., Glass, J. I., Lartigue, C., Noskov, V. N., Chuang, R. Y., Algire, M. A., ... & Merryman, C. Creation of a bacterial cell controlled by a chemically synthesized genome. science, 329(5987), 52-56 (2010).
6. Gibson, D. G., Young, L., Chuang, R. Y., Venter, J. C., Hutchison, C. A., & Smith, H. O. Enzymatic assembly of DNA molecules up to several hundred kilobases. Nature methods, 6(5), 343-345 (2009).
7. LaVan, D. A., & Marmon, L. M. Safe and effective synthetic biology. Nature biotechnology, 28(10), 1010-1012 (2010).
8. Li, J. W. H., & Vederas, J. C. Drug discovery and natural products: end of an era or an endless frontier?. Science, 325(5937), 161-165 (2009).
9. Martin, P., Boulukos, K. E., & Pognonec, P. REtools: A laboratory program for restriction enzyme work: enzyme selection and reaction condition assistance. BMC bioinformatics, 7(1), 98 (2006).
10. Mitchell, L. A., Chuang, J., Agmon, N., Khunsriraksakul, C., Phillips, N. A., Cai, Y., ... & Blomquist, P. Versatile genetic assembly system (VEGAS) to assemble pathways for expression in S. cerevisiae. Nucleic acids research, gkv466 (2015).
11. Narnbiar, K. P., Stackhouse, J., Stauifer, D. M., Kennedy, W. P., & Eldredge, J. K. Total synthesis and cloning of a gene coding for the ribonuclease S protein. Science, 223, 1299-1301 (1984).
12. Pereira, F., Azevedo, F., Carvalho, Â., Ribeiro, G. F., Budde, M. W., & Johansson, B. Pydna: a simulation and documentation tool for DNA assembly strategies using python. BMC bioinformatics, 16(1), 142 (2015).
13. Shao, Z., & Zhao, H. Construction and engineering of large biochemical pathways via DNA assembler. Synthetic Biology, 85-106 (2013).
14. Shao, Z., Zhao, H., & Zhao, H. DNA assembler, an in vivo genetic method for rapid construction of biochemical pathways. Nucleic acids research, 37(2), e16-e16 (2009).
15. Sismour, A. M., & Benner, S. A. Synthetic biology. Expert opinion on biological therapy, 5(11), 1409-1414 (2005).
16. Tian, J., Ma, K., & Saaem, I. Advancing high-throughput gene synthesis technology. Molecular BioSystems, 5(7), 714-722 (2009).
17. Villalobos, A., Ness, J. E., Gustafsson, C., Minshull, J., & Govindarajan, S. Gene Designer: a synthetic biology tool for constructing artificial DNA segments. *BMC bioinformatics*, *7*(1), 285 (2006).
18. Weeding, E., Houle, J., & Kaznessis, Y. N. SynBioSS designer: a web-based tool for the automated generation of kinetic models for synthetic biological constructs. Briefings in bioinformatics, 11(4), 394-402 (2010).