

Lawrence Berkeley National Laboratory

LBL Publications

Title

DNA scanner: A web application for comparing DNA synthesis feasibility, price and turnaround time across vendors

Permalink

<https://escholarship.org/uc/item/7f08512g>

Journal

Synthetic Biology, 5(1)

ISSN

1939-7267

Authors

Doçi, G
Fuchs, L
Kharbanda, Y
et al.

Publication Date

2020

DOI

10.1093/SYNBIO/YSAA011

Peer reviewed

DNA Scanner: a web application for comparing DNA synthesis feasibility, price and turnaround time across vendors

Gledon Doçi,¹ Lukas Fuchs,¹ Yash Kharbanda,¹ Paul Schickling,¹
Valentin Zulkower,^{2,3} Nathan Hillson,^{3,4,5,6} Ernst Oberortner,^{3,5,7}
Neil Swainston,^{3,8} and Johannes Kabisch^{3,9,10,*}

¹Department of Computer Science, Technical University Darmstadt, Darmstadt, Germany, ²Edinburgh Genome Foundry, SynthSys, School of Biological Sciences, University of Edinburgh, EH9 3BF, Edinburgh, UK, ³Software Working Group, Global Biofoundries Alliance, ⁴DOE Agile BioFoundry, Emeryville, CA 94608, USA, ⁵DOE Joint Genome Institute, Berkeley, CA 94720, USA, ⁶Biological Systems and Engineering, Lawrence Berkeley National Lab, Berkeley, CA 94720, USA, ⁷Environmental Genomics and Systems Biology Divisions, Lawrence Berkeley National Lab, Berkeley, CA 94720, USA, ⁸Institute of Systems, Molecular and Integrative Biology, University of Liverpool, Liverpool L69 7ZB, UK, ⁹Computer-aided Synthetic Biology, Technical University Darmstadt, Darmstadt, Germany and ¹⁰Centre for Synthetic Biology, Technische Universität Darmstadt, Darmstadt, Germany

Authors in alphabetical order contributed equally.

*Corresponding author: E-mail: johannes@kabisch-lab.de

Abstract

DNA synthesis has become a major enabler of modern bioengineering, allowing scientists to simply order online in silico-designed DNA molecules. Rapidly decreasing DNA synthesis service prices and the concomitant increase of research and development scales bolstered by computer-aided DNA design tools and laboratory automation has driven up the demand for synthetic DNA. While vendors provide user-friendly online portals for purchasing synthetic DNA, customers still face the time-consuming task of checking each vendor of choice for their ability and pricing to synthesize the desired sequences. As a result, ordering large batches of DNA sequences can be a laborious manual procedure in an otherwise increasingly automatable workflow. Even when they are available, there is a high degree of technical knowledge and effort required to integrate vendors' application programming interfaces (APIs) into computer-aided DNA design tools or automated lab processes. Here, we introduce *DNA Scanner*, a software package comprising (i) a web-based user interface enabling users to compare the feasibility, price and turnaround time of synthetic DNA sequences across selected vendors and (ii) a Python API enabling integration of these functionalities into computer-aided DNA design tools and automated lab processes. We have developed *DNA Scanner* to uniformly streamline interactions between synthetic DNA vendors, members of the Global Biofoundry Alliance and the scientific community at large.

Submitted: 5 May 2020; Received (in revised form): 9 July 2020; Accepted: 17 July 2020

Published by Oxford University Press 2020. This work is written by a US Government employee and is in the public domain in the US.

1. Introduction

A single synthetic biology project may require on the order of hundreds of different pieces of DNA (spanning primers, linear DNA, plasmids, etc.) (1). While some vendors provide an application programming interface (API, see Table 1 for an overview) to enable automated bulk ordering, the global marketplace for commercial DNA synthesis is fragmented (e.g. in terms of capabilities, turnaround times and pricing), which can make provider selection difficult. This fragmentation has been curated by the Edinburgh Genome Foundry (2).

Other industries have developed web marketplaces that match provider capabilities to customer needs. This benefits customers by facilitating the ordering process, and providers by opening additional sales channels that increase revenue. While travel and insurance sectors (e.g.) have developed web marketplaces and price comparison tools, there is nothing comparable yet for synthetic DNA.

As a first step toward such a marketplace, the Global Biofoundry Alliance's (3) (GBA's) Software Working Group initiated the development of *DNA Scanner*, which beyond the GBA could be useful to the synthetic biology community. The software was envisioned to enable users to compare DNA synthesis feasibility, turnaround time and pricing across vendors and additionally to provide software engineers a standardized API for interacting with these DNA synthesis vendors. *DNA Scanner* is divided into a web-based user interface (UI) and a Python back end. The web UI enables users to use *DNA Scanner* functionalities in an intuitive way. The user can input nucleotide and amino acid sequences in various file formats, including FASTA, GenBank and Synthetic Biology Open Language (SBOL) (4). *DNA Scanner* integrates Build-Optimization Software Tools (BOOST) (5) functionalities, including the optimization of protein-coding sequences for expression and synthesis. For each sequence, *DNA Scanner* displays a comparison across vendors including violated constraints that make a synthesis unfeasible, pricing and delivery time.

The Python back end of *DNA Scanner* receives input data from the web UI and queries the APIs of DNA synthesis vendors and, depending on the vendor, reports back on the feasibility of synthesis, pricing and turnaround time. The back-end architecture is designed to be expandable for future APIs of additional

vendors and provides an API of its own, allowing integration into custom workflows and DNA design tools.

2. Architecture and workflow of DNA Scanner

A schematic depicting the components and the workflow of *DNA Scanner* is shown in Figure 1, which is elaborated in the following subsections.

2.1 Python back end

The Python back end is divided into two sections, the Pinger and the Controller. The Pinger standardizes the communication with the vendors. It is designed as a stand-alone library and can be used outside of the current use case, for example in automating design pipelines of biofoundries. The Controller connects the Pinger and front end by handling user inputs and managing the processes of *DNA Scanner*.

Every DNA synthesis vendor's API is structured differently. The Pinger-library addresses this in the use case of *DNA Scanner*. It has a unified interface that accurately describes the structure and values of inputs and performs error-handling. Behind this interface, each vendor API is used to implement a template with well-defined functions. For example, ordering five DNA sequences may be consolidated into a single request for vendor X, but may require five separate requests for vendor Y. Each vendor-specific implementation of the template (Vendor-Pinger) can immediately be used to extend the Pinger to new vendors. Thus, the Pinger-library consists of as many Vendor-Pingers as vendor APIs that have been implemented.

The task of the Controller is to connect the Pinger-library, the web UI and external services such as the BOOST API. The Controller is the continuously running component of the back-end application. When it starts, the Controller reads the configuration file, which contains technical options and credentials for each vendor-Pinger. It loads the endpoints used by the front end to upload sequence files, starts searching for offers, ordering and filtering these by price, delivery time and vendor. The filter also ensures that non-selected vendors will not be contacted to prevent unnecessary network overhead on both sides. It also connects the BOOST API for codon optimization and

Table 1. API availability and capabilities of different DNA synthesis vendors sorted in alphabetical order

Vendor (alphabetical order)	API available ^a	Construct validation	Ordering through API
ATUM	Yes, currently only disclosed to industry partners		
BaseClear	No API		
Biomatik	No API		
Blue Heron Biotech	No API		
Eurofins Genomics	No reply		
GeneArt	Yes	Yes	Yes. Redirects to the cart on the vendor's website
GeneUniversal	Development if requested		
GenScript	No interest to provide access to API		
Integrated DNA Technologies (IDT)	Yes, but currently only without prices and turnover time	Yes	Yes, only returns the order number
ProteoGenix	No reply		
SynBio Technologies	No reply		
Twist Bioscience	Yes, but currently only overall data for all sequences requested	Yes	Yes, including quote to be used as PO

^aPersonal communication, at least three communication attempts were made for each vendor.

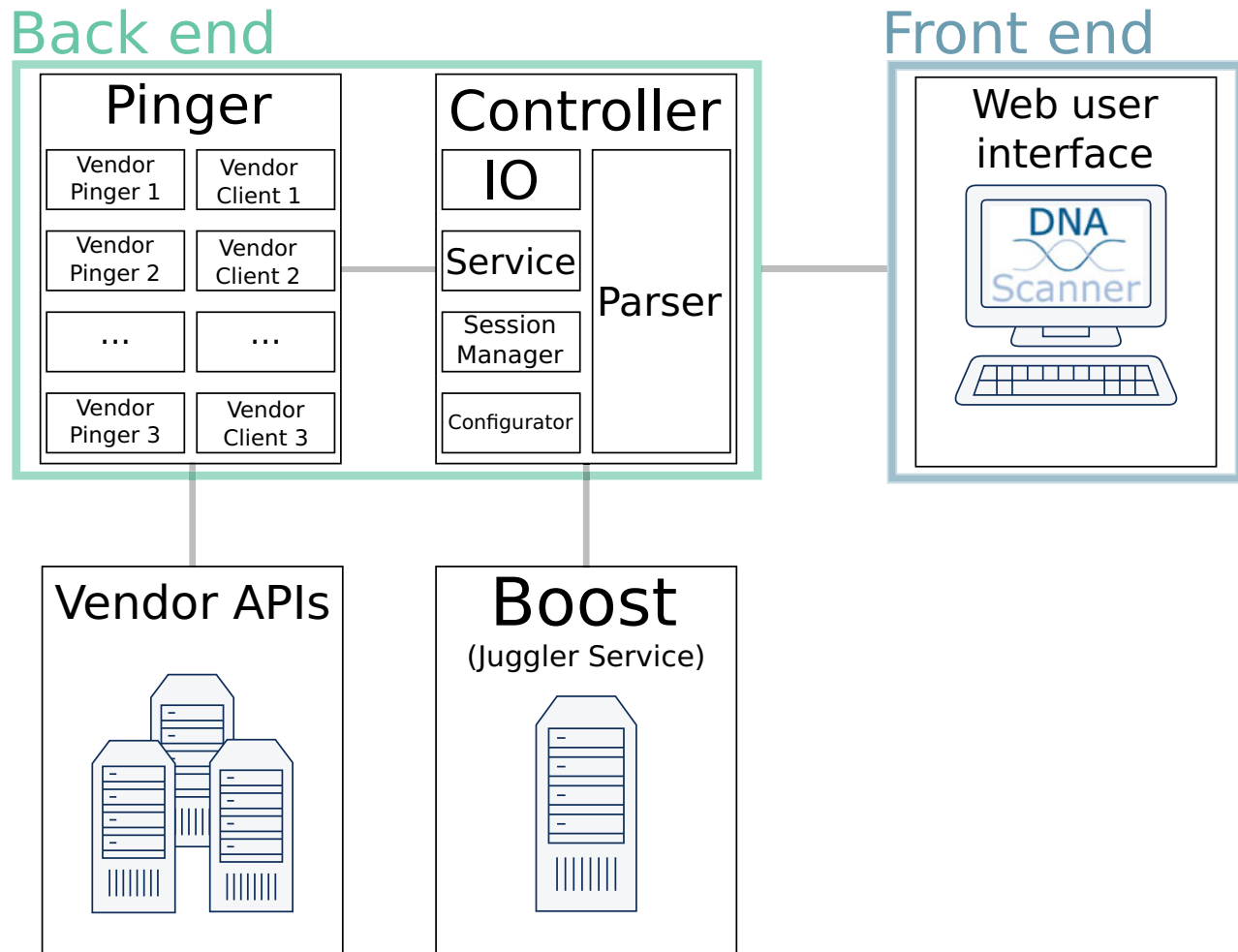


Figure 1. Schematic of DNA Scanner components and workflow. The input/output (IO) component contains all externally visible endpoints, and the Service component provides the logic behind it. The Session Manager saves session-related data on the server. The BOOST API is used if the uploaded file contains only amino acid sequences. Each vendor–client implements vendor API requests and receives the responses. The Vendor-Pinger uses the client’s functions. The connections between the APIs and web UI use HTTP. Further explanations of the functionality can be found in Sections 2 and 3.

back-translation as well as different libraries for parsing sequence files in different formats.

The workflow of the back end can be briefly described as:

1. The front end makes a call to the back end with specific information (sequence file or information e.g. selected offers).
2. In the case of submitting a nucleotide sequence file, the file is parsed directly.
3. If the user denotes the uploaded file to contain only amino acid sequences, these are back-translated and optimized using the BOOST API. The communication with the BOOST API is integrated via the Parser of the Controller.
4. The Pinger-library distributes the information to the specific Vendor-Pingers.
5. The Vendor-Pingers make the calls to various providers.
6. The Pinger-library combines the different results and returns it.
7. The information returned from the Pinger-library will be transformed again by the Controller and saved in the current session. The Controller can also sort or filter the returned quotes (e.g. for a specific vendor) in a user-defined manner. The user sets these filters and sorting criteria via the front end.
8. The information is returned to the front end.

2.2 Deployment

The DNA scanner git repository contains a shell-script which creates and deploys the containers for the back end and front end. It is a default setup which should run on most systems. It can be modified by experienced users to suit individual needs. For example, the web server parameters can be switched to different ports. For the front end, a container with a supported version of node.js is built, which upon completion is copied to another container running an NGINX-http-server to make the web application accessible in a secure way.

2.3 Adding vendors

In most cases, the DNA synthesis vendor APIs are not open. The process for integrating APIs was streamlined based on our experience. First, the vendor API developer has to be identified and contacted. This vendor representative provides the credentials to access their API. With this information, the Pinger-library is then extended by implementing a Vendor-Pinger. The Pinger contains a template for the structure for a new Vendor-Pinger. A correctly defined Vendor-Pinger registers the new vendor in the Pinger-library by passing an instance of it to the register-method.

The Configurator must also be modified to read vendor-specific settings as well as the information that the new Vendor-Pinger is to be used for this vendor from the back-end's configuration file so it can add the Vendor-Pinger to the main Pinger-library on initialization.

To handle the new vendor in the front end, the configuration has to be extended with the user's credentials and other information needed to initialize the Vendor-Pinger. Then the new Vendor-Pinger has to be added to the Configurator of the Controller. The front end appends the new vendor dynamically via the Controller.

3. DNA Scanner

3.1 Functionality of the DNA Scanner back end

Nucleotide and amino acid sequences can be submitted in FASTA, GenBank and SBOL2 (4) formats which are directly converted to sequence strings and sent to the vendors APIs. In the case of submission of amino acid sequences, these are first codon optimized by BOOST as described in Section 3.4, before the resulting nucleotide sequences are forwarded to the configured vendor APIs. The back end has its own API that can either be used to implement the back end into custom workflows or be used in conjunction with the provided front end described below.

3.2 Functionality of the DNA Scanner front end

The front end of DNA Scanner is a web UI that provides an overview of the offers from the selected vendors with the defined filterset. It is able to display offers for a given sequence across

vendors by giving each vendor a different color. At the top, a cumulative summary of total price and total turnaround time for the synthesis of all sequences is displayed for each vendor. If the user selects offers from different vendors, the total price and time are displayed for each vendor individually. The user may apply filters for vendors, price and time. Default settings allow for preselection of offers optimized by either price or time. If a sequence is not synthesizable or a vendor has other concerns, the user is informed by info-boxes with descriptions provided by the vendor. After the user clicks 'FINISH ORDER', new browser tabs for each vendor are opened summarizing the order list. Figure 2 shows an example result page for eight uploaded sequences and three vendors. Additional screenshots of the frontend's landing page and filter view are provided in Supplementary Figures S1–S4.

3.3 Currently integrated vendors and constraints

The vendors Twist Bioscience (Twist), Integrated DNA Technologies (IDT) and GeneArt have provided us access and support to integrate their APIs into DNA Scanner. These three APIs differ in their functionality.

3.3.1 Twist

The API from Twist provides a cumulative price for the complete request but does not return individual prices for each sequence. The length of non-cloned sequences must be between 300 and 1800 bp and the sequence names must be shorter than 32 characters. Longer sequence names are truncated. At the moment, DNA Scanner has only implemented the ordering of non-cloned genes using the Twist API.

Sequence	Twist	Price	Time	IDT	Price	Time	GeneArt	Price	Time
35_DNA_Scanner_seq1	<input type="checkbox"/>	!	!	<input type="checkbox"/>	!	!	<input checked="" type="checkbox"/>	182 €	6 bd.
36_DNA_Scanner_seq2_rejected_Repeat region detected.Repeat region detected.					!	!	<input type="checkbox"/>	0	0
37_DNA_Scanner_seq3	!	!	!	!	!	!	<input type="checkbox"/>	0	0
38_DNA_Scanner_seq4	<input type="checkbox"/>	!	!	<input type="checkbox"/>	!	!	<input checked="" type="checkbox"/>	89 €	5 bd.

Vendor	Price	Time
Twist	!	!
IDT	!	!
GeneArt	89 €	5 bd.
GeneArt	61.75 €	9 bd.

Figure 2. Screenshot of the main DNA Scanner screen showing vendor information. The total price and turnover times all (or selected) sequences from each vendor are displayed. Users can select preferred vendors and services. Exclamation marks indicate additional information tool tips: e.g. If a sequence is not synthesizable, a vendor-supplied message will display when the cursor hovers over the informational tool-tip icon (i.e. Twist notifies that sequence 36 is not synthesizable due to repeats). Additional screenshots detailing further functions can be found in Supplementary Figures S1–S3.

3.3.2 Integrated DNA Technologies

The API from IDT provides feedback about the feasibility of synthesizing a DNA sequence but provides neither prices nor turnover times. To submit orders to IDT, the user has to first sign a biohazard disclosure as well as provide shipping and billing details. This information must be implemented in the provided Vendor-Pinger for IDT.

3.3.3 GeneArt

The API from GeneArt accepts only a specific character set for the sequence names. If the sequence names do not fit these specifications, the *DNA Scanner* back end automatically edits them accordingly. The sequences must have lengths between 150 and 3000 bp.

3.4 Construct optimization through BOOST

DNA Scanner accepts files containing either only nucleotide sequences or only amino acid sequences. In the latter case, each sequence first must be back-translated into a nucleotide sequence. For this, *DNA Scanner* leverages the Juggler service by using the APIs provided by BOOST.

Specifically, when the uploaded file contains amino acid sequences, the user is prompted to choose a codon selection strategy (i.e. random, mostly-used and balanced) and the codon usage table from a predefined list of host organisms. The BOOST API processes the request and responds with the reverse-translated DNA sequences, which are then sent to the vendor APIs for complexity screening, for requesting a quote and for calculating the turnaround times.

4. Conclusion

DNA Scanner provides back and front ends to facilitate and expedite ordering synthetic DNA. The back end's API handles FASTA, GenBank and SBOL2 file formats, and accepts nucleotide and amino acid sequences as input. Amino acid sequences are codon optimized using the BOOST API. A web UI front end facilitates use. *DNA Scanner* must be deployed independently for each institution, and not as a centralized service for the community due to institutional/user authentication requirements. *DNA Scanner* currently supports three vendors, each of which provides different capabilities via their specific APIs. As summarized in Table 1, we contacted 12 vendors, of which five have an API, but so far only three vendors provided the details and support for their integration into *DNA Scanner*. Due to these constraints, the developed web UI is yet of limited use for ordering DNA but can serve as a convenient tool to check feasibility. While queries can be entered manually via the web UI, they can also be sent programmatically via HTTP requests to the back end's API. This enables *DNA Scanner* to be integrated with other software tools such as DNA weaver (6). The software was designed with extensibility in mind, allowing for the integration of additional vendors using the provided templates for vendor-Pingers. Ultimately for *DNA Scanner* to become a service widely applied by the scientific community more vendors would need to adopt the idea of APIs to advance DNA-building to a more streamlined, automatable process. All of the software code is available from the Global Biofoundries Alliance Software Working group git repository under an open-source MIT license (see Section 5).

Future work to further develop *DNA Scanner* could include support for additional vendor APIs, a rating system to track

actual turnover times for delivered constructs, and a learning/analytical functionality, such as recently published by Halper et al. (7), that predicts if a sequence is likely to be synthesizable. Upon more vendors offering programmatically accessible optimization tools an additional class for different optimization strategies besides BOOST could be implemented.

5. Availability

The source code and development process are documented in the Global Biofoundries Alliance Software Working Group git repository at <https://github.com/Global-Biofoundries-Alliance/DNA-scanner>.

Supplementary data

[Supplementary Data](#) are available at SYN BIO online.

Acknowledgments

J.K. would like to acknowledge the organizers of the 'Bachelor Informatikpraktikum' at the Technische Universität Darmstadt for the excellent concept of teaching computer science students by giving them interdisciplinary software development challenges. All authors would like to acknowledge the Global Biofoundries Alliance for fostering this initiative and providing a framework for international Synthetic Biology software collaborations. We further would like to thank Tatiana Konovalova from GeneArt, Gil Raytan from Twist Bioscience and Scott Ford from IDT for their support in implementing their company's APIs.

Funding

N.S. acknowledges funding from the Biotechnology and Biological Sciences Research Council (BBSRC) under grant 'GeneORator: a novel and high-throughput method for the synthetic biology-based improvement of any enzyme' (BB/S004955/1) and from the University of Liverpool. V.Z. acknowledges funding from the BBSRC under grants BB/M025659/1. BB/M025640/1 and BB/M00029X/1 to S.R. and the BBSRC/MRC/EPSCRC funded UK Centre for Mammalian Synthetic Biology (BB/M0101804/1 to S.R.) as part of the RCUK's Synthetic Biology for Growth program. This work was part of the DOE Joint Genome Institute (<https://jgi.doe.gov>) supported by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research, and was part of the Agile BioFoundry (<http://agilebiofoundry.org>) supported by the U.S. Department of Energy, Energy Efficiency and Renewable Energy, Bioenergy Technologies Office, through contract DE-AC02-05CH11231 between Lawrence Berkeley National Laboratory and the U.S. Department of Energy. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately

owned rights. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).

Conflict of interest statement. N.H. has financial interests in TeselaGen Biotechnologies and Ansa Biotechnologies.

References

1. Carbonell, P., Jarvis, A.J., Robinson, C.J., Yan, C., Dunstan, M., Swainston, N., Vinaixa, M., Hollywood, K.A., Currin, A., Rattray, N.J.W. et al. (2018) An automated Design-Build-Test-Learn pipeline for enhanced microbial production of fine chemicals. *Commun. Biol.*, 1, 66.
2. genedeals. <https://github.com/Edinburgh-Genome-Foundry/genedeals> (6 March 2020, date last accessed).
3. Hillson, N., Caddick, M., Cai, Y., Carrasco, J.A., Chang, M.W., Curach, N.C., Bell, D.J., Le Feuvre, R., Friedman, D.C., Fu, X. et al. (2019) Building a global alliance of biofoundries. *Nat. Commun.*, 10, 2040.
4. Madsen, C. et al. (2019) Synthetic Biology Open Language (SBOL) Version 2.3. *J. Integr. Bioinform.*, 16, 20190025.
5. Oberortner, E., Cheng, J.-F., Hillson, N.J. and Deutsch, S. (2017) Streamlining the design-to-build transition with build-optimization software tools. *ACS Synth. Biol.*, 6, 485–496.
6. DNAWeaver. <https://github.com/Edinburgh-Genome-Foundry/DnaWeaver> (6 July 2020, date last accessed).
7. Halper, S.M., Hossain, A. and Salis, H.M. (2020) Synthesis success calculator: predicting the rapid synthesis of DNA fragments with machine learning. *ACS Synth. Biol.*, 9, 1563–1571.