

## Article

**SynBioHub: A Standards-Enabled Design Repository for Synthetic Biology**James Alastair McLaughlin, Chris J. Myers, Zach Zundel, Goksel Misirli,  
Michael Zhang, Irina Dana Ofiteru, Angel Goñi Moreno, and Anil WipatACS Synth. Biol., **Just Accepted Manuscript** • DOI: 10.1021/acssynbio.7b00403 • Publication Date (Web): 09 Jan 2018Downloaded from <http://pubs.acs.org> on January 15, 2018**Just Accepted**

“Just Accepted” manuscripts have been peer-reviewed and accepted for publication. They are posted online prior to technical editing, formatting for publication and author proofing. The American Chemical Society provides “Just Accepted” as a free service to the research community to expedite the dissemination of scientific material as soon as possible after acceptance. “Just Accepted” manuscripts appear in full in PDF format accompanied by an HTML abstract. “Just Accepted” manuscripts have been fully peer reviewed, but should not be considered the official version of record. They are accessible to all readers and citable by the Digital Object Identifier (DOI®). “Just Accepted” is an optional service offered to authors. Therefore, the “Just Accepted” Web site may not include all articles that will be published in the journal. After a manuscript is technically edited and formatted, it will be removed from the “Just Accepted” Web site and published as an ASAP article. Note that technical editing may introduce minor changes to the manuscript text and/or graphics which could affect content, and all legal disclaimers and ethical guidelines that apply to the journal pertain. ACS cannot be held responsible for errors or consequences arising from the use of information contained in these “Just Accepted” manuscripts.

# SynBioHub: A Standards-Enabled Design Repository for Synthetic Biology

James Alastair McLaughlin,<sup>†</sup> Chris J. Myers,<sup>‡</sup> Zach Zundel,<sup>‡</sup> Göksel Mısırlı,<sup>¶</sup>  
Michael Zhang,<sup>‡</sup> Irina Dana Ofiteru,<sup>§</sup> Angel Goñi-Moreno,<sup>†</sup> and Anil Wipat<sup>\*,†</sup>

<sup>†</sup>*School of Computing, Newcastle University, NE1 7RU, UK*

<sup>‡</sup>*Department of Electrical and Computer Engineering, University of Utah, Salt Lake City,  
UT 84112, USA*

<sup>¶</sup>*School of Computing and Mathematics, Keele University, ST5 5BG, UK*

<sup>§</sup>*School of Engineering, Newcastle University, NE1 7RU, UK*

E-mail: anil.wipat@ncl.ac.uk

## Abstract

The SynBioHub repository (<https://synbiohub.org>) is an open-source software project which facilitates the sharing of information about engineered biological systems. SynBioHub provides computational access for software and data integration, and a graphical user interface that enables users to search for and share designs in a Web browser. By connecting to relevant repositories (e.g. the iGEM repository, JBEI ICE, and other instances of SynBioHub), the software allows users to browse, upload, and download data in various standard formats, regardless of their location or representation. SynBioHub also provides a central reference point for other resources to link to, delivering design information in a standardized format using the Synthetic Biology Open Language (SBOL). The adoption and use of SynBioHub, a community-driven effort, has the potential to overcome the reproducibility challenge across laboratories by helping to address the current lack of information about published designs.

## Keywords

Synthetic Biology Open Language, SBOL, Repository, Web of Registries, Data Standards, Database, Sharing

A fundamental goal of synthetic biology is to make biological systems easier to engineer.<sup>1</sup> As part of this endeavour, significant attention is being paid to the development of *workflows*<sup>2-6</sup> that will assist researchers through the synthetic biology lifecycle. By drawing on concepts from engineering,<sup>7</sup> an emphasis is placed on the design process through the use of basics such as standardization, reusability, abstracticon, modularity, and predictability. Numerous so-called *genetic design automation* (GDA) tools and techniques are now emerging to aid in the design of engineered biological systems,<sup>8-10</sup> e.g. TinkerCell<sup>11</sup>, SBOLDesigner,<sup>12</sup> iBioSim,<sup>13</sup> SBROME,<sup>14</sup> CELLO,<sup>15</sup> Pigeon,<sup>16</sup> DNAplotlib,<sup>17</sup> and VisBOL.<sup>18</sup> Nevertheless, designing new engineered organisms is still difficult due to the complexity of these systems and gaps in our knowledge of the biology involved.

One of the biggest challenges, and a barrier to the reuse of successful designs, is that biological data relevant to the design of novel systems are often not exchanged. Design efforts are usually carried out in individual laboratories, by teams in different geographic locations. The lack of data exchange not only concerns the genetic details and sequence of the designed system, but also experimental data about the characterization and behavior of the resulting system and mathematical models. As a result, it is very difficult, if not impossible, to repeat or reuse the work presented in many synthetic biology publications.<sup>19</sup>

To address this problem, several systems for storing and sharing data about engineered biological designs have been developed. Most notable are JBEI-ICE,<sup>20</sup> the iGEM Registry of Standard Biological Parts (<http://parts.igem.org>), and the SBOL Stack.<sup>21</sup> Many commercial and non-commercial software tools also exist that offer the ability to store genetic designs, such as VectorNTI,<sup>22</sup> Benchling, and Addgene.<sup>23</sup> JBEI-ICE is one such registry that allows information about biological parts to be managed and shared. The repository is accessible via both a Web browser and through a Web *application programming inter-*

1  
2  
3 *face* (API). ICE also introduced the concept of a *Web of Registries*, providing support for  
4 ICE repositories to be connected to allow distributed and interconnected use. The ICE  
5 repository supports the import and export of sequence data in various formats, but it is  
6 limited in its support of other design information, such as non-DNA components and their  
7 interactions. Another example repository is the SBOL Stack, which provides a database for  
8 programatically storing design information, but is not a "user friendly" solution suitable for  
9 users outside of computer science.  
10

11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
Despite these developments in the storage of DNA designs, there is an increasing need to  
capture information about synthetic systems at a more abstract level, including information  
about proteins, interactions, metabolites, and the organism for which a design is intended  
to operate in once implemented. It is also very important to capture the history of design,  
so that the lineage of designs can be recorded. The inclusion of this type of information  
requires richer data standards beyond the simple GenBank and EMBL formats that are still  
used routinely in the design of engineered biological systems. The *Synthetic Biology Open  
Language* (SBOL) was developed to meet this need.<sup>24,25</sup>

The initial version of SBOL focused upon exchanging genetic descriptions of parts at the  
DNA level, while SBOL version 2<sup>25,26</sup> supports other types of genetic circuit components,  
including proteins, RNAs, and small molecules. The standard also supports provenance  
information<sup>27</sup> concerning lineage and history of a design, together with mechanisms that  
allow integration with other data standards that allow laboratory experimental data to be  
included with the design for system. In short, SBOL now allows the design for a system  
to be shared, together with information on the purpose and history of that design, mathe-  
matical models of the design, visual representations, and experimental characterization data  
(captured in other complementary data standards) that describes the behavior of the system  
once built and tested. Finally, SBOL also includes a visualization standard, SBOL Visual,  
for representing genetic circuit designs in diagrams composed of a standard set of glyphs  
representing genetic components.<sup>28</sup>

1  
2  
3 The continued development of SBOL provides the opportunity for a design repository  
4 specifically for synthetic biology, which can capture data not just about sequences on a  
5 DNA level, but information about the *who*, *where*, *why*, and *how* of the design process.  
6  
7 However, so far there exists no central "hub" to deposit SBOL designs while preserving  
8 the capability to search for, retrieve, and visualize the complex information captured in the  
9 design process. Although the JBEI-ICE repository can import and export SBOL data files,  
10 its internal data model is still organized around a sequence-centric representation. In other  
11 words, JBEI-ICE stores SBOL as an opaque file and the rich information available in SBOL  
12 is not exposed to the user for querying or in a form that can be integrated with other data,  
13 such as experimental data and mathematical models. Therefore, a need exists for a data  
14 repository that can capture this genetic design information encoded in SBOL 2 in a more  
15 computationally tractable form.  
16  
17

18  
19 To address this need, we have developed SynBioHub, a repository for storing information  
20 about engineered biological systems, built on the SBOL Stack database and using SBOL 2  
21 as the core storage format. SynBioHub functions both as a place to store designs, and as an  
22 aggregation facility to bring together information from many different repositories, including  
23 JBEI-ICE, Benchling, and the iGEM Registry of Standard Biological Parts. The syntactic  
24 and semantic disparity between these repositories is automatically harmonized in SynBioHub  
25 through conversion to and from SBOL, which makes it easier to search and retrieve design  
26 components with a canonicalized syntax. Since designs are exchanged and stored in the  
27 form of SBOL, the repository is also capable of storing information about the provenance  
28 of designs; designs for which the sequence is not yet available; and designs that include  
29 information about RNA, proteins, interactions, metabolites, and mathematical models.  
30  
31

32  
33 SynBioHub provides a Web-based user interface allowing users to browse, upload, and  
34 share designs without the requirement of any locally installed software. Unlike the origi-  
35 nal SBOL Stack, SynBioHub completely abstracts away the details of the underlying data  
36 representation, enabling non-technical users to interact with the front-end while generating  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

1  
2  
3 semantically rich SBOL data to facilitate data integration and analysis.  
4

5       SynBioHub also supports the Web of Registries concept: the idea of multiple separate  
6 repositories linked together by shared common semantics. In the case of SynBioHub, SBOL  
7 is used to support the common exchange of data, thus positioning SynBioHub to support  
8 the development of synthetic biology workflows by acting as a source and a storage facility  
9 for designs. SynBioHub demonstrates the application of this harmonized data exchange  
10 through integration with existing repositories. We envisage a wider, integrated ecosystem  
11 of biological part information shared across a plethora of different repositories with different  
12 capabilities.  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22

## 23 **Results and discussion**

24  
25  
26 The architecture of SynBioHub is designed to accommodate the concept of FAIRdom previ-  
27 ously applied to systems biology:<sup>29</sup> that data should be **f**indable, **a**ccessible, **i**nteroperable,  
28 and **r**eusable. For SynBioHub, this means that powerful querying facilities should be avail-  
29 able for findability; both a user interface and computational API should be available for  
30 accessibility; design data should be represented using an open, standardized data model for  
31 interoperability; and all information possible should be preserved to facilitate reuse. To ac-  
32 complish these goals, the front page of any SynBioHub installation provides three options: to  
33 search for designs, upload designs to the repository, and share existing uploaded designs for  
34 publication or collaboration (Fig. 1). Each of these three major functionalities is described  
35 in more detail below.  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47

### 48 **Search and Data Visualization**

49  
50  
51 To facilitate efficient search capabilities, SynBioHub provides a flexible data storage mech-  
52 anism for storing SBOL data. These data are graph based and include information about  
53 biological design components, their sequences, and interactions between these components.  
54  
55  
56  
57  
58  
59  
60



Figure 1: The front page of SynBioHub. Users can choose to search for, upload, or share designs for publication or collaboration.

1  
2  
3 Designs are often hierarchical; a parent component can be formed of several child compo-  
4 nents which in turn may include complex designs. Behavioral representations of biological  
5 systems are modularized through parts that are defined with inputs and outputs to scale  
6 up to larger designs. These complex relationships are best represented as graphs. Based  
7 on RDF/XML, SBOL already takes advantage of graph representations. Although different  
8 options can be considered when storing data in a database, we chose a graph-based database  
9 approach to take advantage of the storage and integration of SBOL documents, and also the  
10 efficient querying of data and underlying complex relationships.  
11  
12  
13  
14  
15  
16  
17  
18

19 To achieve this goal, the central component of SynBioHub is the SBOL Stack<sup>21</sup> architec-  
20 ture. The SBOL Stack pioneered the concept of using an RDF triplestore to store SBOL data  
21 as triples, and retrieving designs in whole or part using SPARQL queries.<sup>30</sup> SynBioHub takes  
22 this further by providing a complete user-facing Web interface. Behind the scenes, queries to  
23 the Web interface are translated into SPARQL queries using the SBOL data model, and data  
24 are returned from the triplestore as RDF triples. These triples are then used to construct a  
25 partial graph, which is queried to produce information listings and visual depictions using  
26 VisBOL.<sup>18</sup> The database is effectively schemaless. Therefore SBOL documents that include  
27 custom and complex user annotations can also be stored efficiently, and queries can be ex-  
28 tented to retrieve custom information. As data relevant to the design of biological systems  
29 grows, this approach makes the integration and retrieval of data a streamlined process in  
30 synthetic biology.  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42

43 While querying the RDF triplestore using SPARQL queries provides a powerful means  
44 for search, constructing such queries for most target users would prove quite challenging.  
45 Therefore, SynBioHub provides several simplified means of search. For example, a user can  
46 perform simple text based searches that identify designs that include the provided terms  
47 within their identifier, name, or description. SynBioHub also includes a more advanced  
48 search capability that allows designs to be filtered by the date they were created, the name  
49 of their creator, their type and functional role, etc. Finally, a user can simply browse the  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

designs organized into public *collections* (Fig. 2a), and upon selecting a collection, the user is presented with a collection view that allows the search to be further refined (Fig. 2b).

Once a design is selected, either from browsing collections or from search results, the user is presented with the Design View (Fig. 2c). This view renders the design information data captured by SBOL in a human readable form. At the top of the page, the title, version, type, modification dates, short description, and provenance information (i.e., where the part was retrieved from) are displayed. Next, there is an SBOL Visual representation of the design rendered using VisBOL.<sup>18</sup> If the features shown in this visual depiction point to another design in the Web of Registries, they can be selected to navigate to the view for that design. Below the visual depiction, more detailed information is provided, such as *Sequence Ontology*<sup>31</sup> roles, custom user-defined annotations, data files, and other attachments.

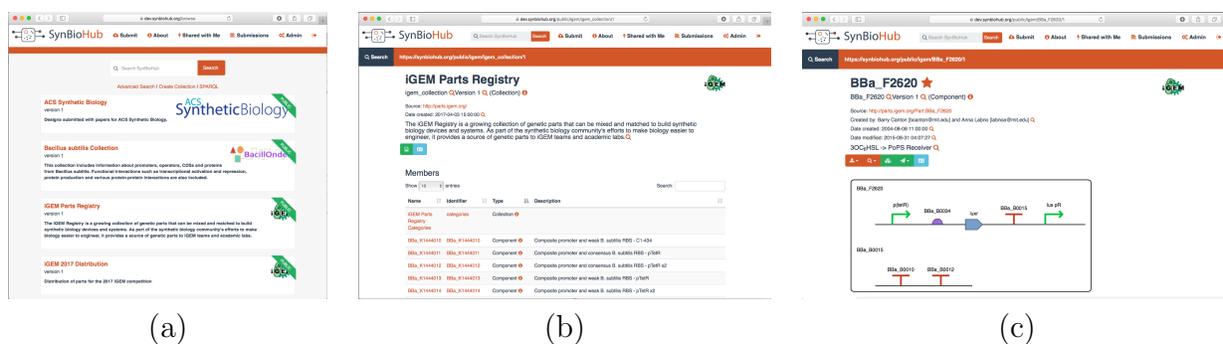


Figure 2: Search and data visualization in SynBioHub. (a) The browse design collections page shows all of the "root" collections in the repository. These are collections that are not contained by any other collection. The reference instance includes, among others, collections for designs published in ACS Synthetic Biology, a collection of *Bacillus subtilis* parts, and the iGEM registry parts. (b) The iGEM Parts Registry collection on the reference instance of SynBioHub. This collection view shows all of the parts available in the selected collection. The parts can be sorted by name, identifier, type, or description, and can be filtered using a free-text search. (c) The SynBioHub part view showing a part from the iGEM Registry. The SBOL Visual depiction is generated using VisBOL by mapping Sequence Ontology terms to glyphs. The part view can be used to send parts to the Benchling sequence design tool or to an ICE repository.

## Submission and Design Documentation

Registered users of SynBioHub can submit new designs via the submission interface. New designs are submitted into collections that are in the private repository space for the logged in user. Therefore, these submitted designs are initially only visible to the users who submit them. Methods for the sharing of this design information with other users and the public are described below. Each new design document can either be uploaded into a new private collection or merged with an existing private collection. SynBioHub further simplifies the uploading of design data by supporting a variety of data formats, including older SBOL versions, GenBank, and FASTA. SynBioHub utilizes libSBOLj,<sup>32</sup> a Java library developed to read and write compliant SBOL documents, when converting different formats to the most recent SBOL version.

Once uploaded, additional information can be added to further document a design. In particular, the design view also incorporates “wiki” functionality, by which a much longer description can be edited by multiple users, subject to the permission model described below. References to publications can also be added by entering PubMed Ids that are then used to construct full references. Finally, SynBioHub allows attachments to be added to any design. These attachments can be of any file format, such as an SBML model, a graph image, or a spreadsheet. Attachments are saved in a compressed file store and can be downloaded on demand by any user with access to the design. Image attachments can be displayed within the “wiki” description.

## Sharing and Security

As mentioned above, a newly submitted design is initially placed in the private repository for the submitting user. This private repository is a distinct, private graph in Virtuoso, the RDF triplestore database. To ensure the data in this graph remains private, queries are limited throughout the application to the public graph and the private graph of the querying user, if authenticated.

1  
2  
3 The user can share a design by creating a share link, adding other SynBioHub users as  
4 owners of the design, and making the design public. When a share link is created, a one-way  
5 hash of the design URI and a secret value is used to ensure that share links can only be  
6 generated by a user who owns the design. If a share link is accessed, SynBioHub verifies  
7 the hash to make sure that the share link is valid. If it is, then the design is rendered for  
8 view-only. In order to allow another user access to edit the description, references, and other  
9 mutable fields, an owner of a design can add another owner to the design. Finally, once  
10 a design is complete and fully documented, the owner can make the design public, which  
11 moves the design into the public graph allowing it to be queried and viewed by anyone.

12  
13  
14  
15  
16  
17  
18  
19  
20  
21 Designs can be shared not only with other users, but also with other applications. In  
22 particular, SynBioHub is able to communicate with Benchling and JBEI-ICE. SynBioHub  
23 can copy sequences to a repository within Benchling for sequence editing, as well as copy  
24 sequences back into SynBioHub collections. Similarly, designs can be transferred to/from  
25 JBEI-ICE. In both cases, these communications are enabled by translations to/from SBOL.

26  
27  
28  
29  
30  
31 A design that is uploaded to a SynBioHub instance also becomes available to users from  
32 other instances due to federation support in SynBioHub. Data federation is a technique that  
33 can be used to integrate data from remote sources using a single query<sup>33</sup>. For example, mul-  
34 tiple RDF triplestores can be computationally queried using the SPARQL query language.  
35 Since SynBioHub is backed by an RDF triplestore, federation of queries to multiple SynBio-  
36 Hub instances is natively supported. SynBioHub uses SBOL as the common semantics to  
37 query and to create integrated views of designs for users.

38  
39  
40  
41  
42  
43  
44  
45 Sharing designs between instances of SynBioHub is further facilitated via the Web of  
46 Registries service. Federated SynBioHub instances can reference parts in the public graph  
47 of other federated instances. In order to support this federation, the Web of Registries ser-  
48 vice maintains information about all SynBioHub registries, such as their name, description,  
49 administrator email, URI prefix, and *uniform resource locator* (URL) (Fig. 3). It also keeps  
50 track of whether or not the SynBioHub is being successfully updated. When a new SynBio-  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

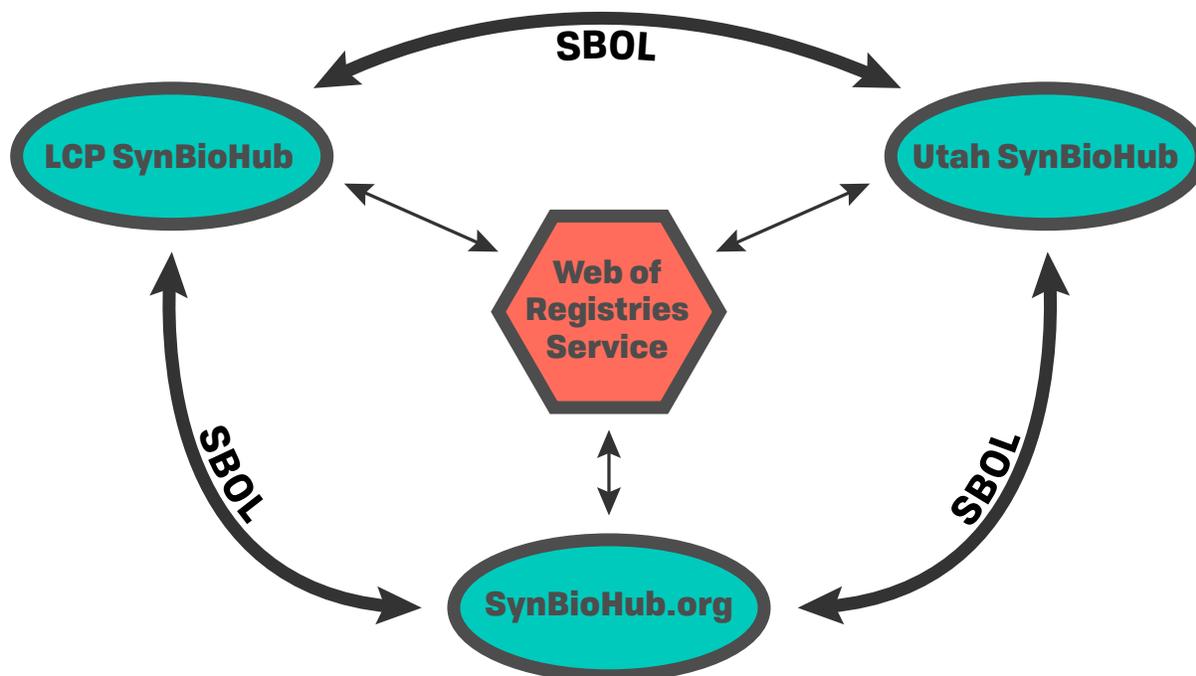


Figure 3: The Web of Registries enables communication between SynBioHub instances. Any SynBioHub can access the Web of Registries to determine information about all registered SynBioHub instances. If a design references a part within another SynBioHub instance, the information about this part can be fetched in order to render this design information locally and provide links to the corresponding design information page for this part.

Hub instance wishes to join the Web of Registries, it sends its information to the Web of Registries service, and all the Web of Registries curators are alerted via email that there is a new repository pending approval. Once the repository has been approved, its information is broadcast to all other registries in the Web of Registries. Anybody can access the current list of registries through an HTTP GET request. Once registered, an instance of SynBioHub is able to locate designs within any other instance that is registered with the Web of Registries.

## Computational API

In addition to providing a graphical user interface, SynBioHub also provides a computational API for programmatic access. The API is a RESTful Web service which can be used from any modern programming language, and has endpoints to search for, retrieve, and upload

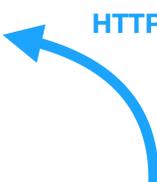
designs. Security is also a concern in the HTTP API. When a user is authenticated through the API, a token is returned. Any further queries that require authentication (submitting or searching non-public graphs) require that this token is passed in the `X-Authorization` header. On the backend, there is middleware which uses the authentication token to attach the user to the request in the same manner as requests from the SynBioHub user interface.

An interface to the API has also been implemented as a set of Java classes which have now been incorporated into `libSBOLj`<sup>32</sup> (Fig. 4). This integration of access to a SynBioHub repository has been tightly coupled with the SBOL library, and the coupling has been used to provide a novel system of completing partial designs upon load. If an SBOL file is loaded into `libSBOLj` and references URIs which are not present in the file but are present in a remote repository, `libSBOLj` can automatically retrieve the missing design information using the SynBioHub API to create a complete document. This automated resolution creates a system of "dependencies" as used in computer software libraries, but applied to biological designs.

#### SynBioHub RESTful API

```
/login  
/rootCollections  
/<collection>/subCollections  
/<type>/count  
/<type>/search  
/<collection>/<id>/sbol  
/submit
```

HTTP & SBOL



#### libSBOLj interface

```
SynBioHubFrontend frontend = new ...  
SearchQuery query = new ...  
query.addCriteria(...)  
Result[] results = frontend.search(query)
```

Figure 4: An example of some of the functionality that SynBioHub provides through its Application Programming Interface (API) and the relevant Java pseudo-code. `libSBOLj` communicates with SynBioHub using the RESTful API over HTTP, and SBOL as the data exchange format.

1  
2  
3 The first software to utilize this libSBOLj interface to the API was SBOLDesigner, a  
4 sequence based computer-aided design tool.<sup>12</sup> This support allows users of SBOLDesigner  
5 to login to various instances of the repository and query, view, download, use, and upload  
6 designs. After logging in, designs from private collections can be seen and searched through.  
7 Any of these designs can be downloaded and inserted into the current design. From there,  
8 the design can be modified and used in various ways. After the sequence-level design is  
9 finished, the complete SBOL document can be uploaded back to SynBioHub into the user's  
10 private collection.  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20

## 21 Synthetic Biology Workflows

22  
23 The architecture and features of SynBioHub enable several common workflows utilized by  
24 synthetic biologists to produce genetic design information. One such example workflow,  
25 depicted in Fig. 5a, is the publication of sufficient information to enable reproducibility of  
26 genetic designs. In this workflow, a researcher begins by using their GDA tools of choice  
27 to create a genetic design. If this tool supports direct communication with the SynBioHub  
28 HTTP API, the parts created can be directly deposited in the SynBioHub repository. Oth-  
29 erwise, GenBank, FASTA, or SBOL files exported by the design software can be uploaded  
30 into SynBioHub. Once the design is uploaded, the author can obtain a sharable link that  
31 can be provided to the journal and used by reviewers to privately access the design infor-  
32 mation and evaluate its completeness as part of the review process. SynBioHub can also  
33 produce an SBOL Visual diagram that can be downloaded from the page and included in  
34 the publication, if desired. Once the paper has been accepted for publication, the design can  
35 then moved into the public space on SynBioHub, and a link to the design can be included  
36 in the publication, granting readers direct access to all the information provided about the  
37 design.  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52

53 Another example workflow is that of a team competing in the popular *Internationally*  
54 *Genetically Engineered Machines* (iGEM) competition (Fig. 5b). One team member can  
55  
56  
57  
58  
59  
60

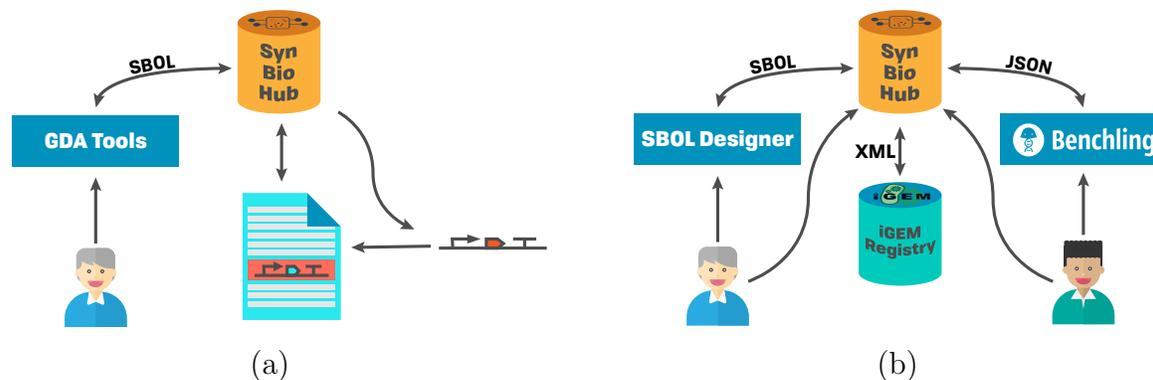


Figure 5: Synthetic biology workflows enabled by SynBioHub. (a) A research scientist prepares to publish information about genetic constructs. Constructs can be pushed to SynBioHub, where they can be referenced by a static URL in the published article. Additionally, SynBioHub can generate SBOL Visual figures (currently only in PNG format) for uploaded parts. (b) An iGEM team can utilize the sharing and ownership functionality of SynBioHub to enable multiple users to collaborate on the same design. API data transfers also support each step of the design process, allowing for data transfer without manual file download/upload.

create a design using iGEM parts and plasmids retrieved directly from SynBioHub using a sequence editing tool such as SBOLDesigner. The resulting composite design can then be uploaded directly to SynBioHub. The design information can be placed into a single collection for their project. Using SynBioHub's ownership management functionality, they can add their teammates to this collection as owners, allowing them to modify this design. Another team member may produce an assembly plan using the Benchling tool by copying from SynBioHub the design information into the Benchling repository for the team. Another teammate may add additional information to the assembly plan using the free-text mutable part description fields on SynBioHub. Finally, in the future, SynBioHub could potentially communicate with the iGEM registry through its XML API, allowing push-button publication of the team's data to the iGEM registry for the competition.

## Discussion

The description and storage of information that allows for reproducible genetic designs and constructs is one of the contemporary challenges of synthetic biology. The SynBioHub repository contributes to this endeavor by assisting the sharing of such genetic designs and facilitating the integration of existing software packages and databases. The vision for SynBioHub is three-fold. First, the system provides a user-facing repository for designs encoded in the emerging community standard SBOL. Second, the system supports the web of registries concept through data federation, while also acting as an information storage and retrieval system for supporting the operation of standardized workflows of synthetic biology tools. Third, the system can act as an integration hub for other types of biological information, including experimental data. The capabilities of SBOL to overcome the limited access to DNA sequence information displayed by other formats, such as GenBank, is key to the functioning of SynBioHub. SBOL provides detailed descriptions of not only sequences, but also other essential information such as interactions between components, hierarchies, and provenance of designs. Altogether, SBOL allows SynBioHub to structure the repository around an information-rich design format; a goal arguably unattained by any other repository to date.

SBOL is being increasingly adopted by the synthetic biology community at a steady rate. However, SynBioHub provides several features that can potentially accelerate this process. Firstly, it provides public access to SBOL encoded information, such as the iGEM dataset. Secondly, it provides an API for programmatic access by software tools to fetch and store genetic designs. Finally, it includes a means to communicate with software tools such as Benchling and JBEI ICE, allowing it to serve as a unifying hub for the development of synthetic biology workflows. Each of these features has been enabled by the ability of SBOL to store and represent genetic design information. SBOL 2, in particular, extends beyond the representation of just genetic features. The SBOL data model allows parts hierarchies, modules, and mathematical models to be stored, together with information about other

1  
2  
3 molecules such as RNAs, proteins, metabolites, and the interactions between these molecules.  
4  
5 This feature is essential to support a top-down approach to engineering biological systems  
6  
7 where the design is carried out at the phenotype and pathway level and the genetics of the  
8  
9 system is derived last of all.  
10

11 In the future, we plan to extend the data available within SynBioHub instances via data  
12  
13 integration<sup>34</sup>. Similar to the conversion that we performed of the iGEM dataset, we would  
14  
15 like to extract data from other repositories, such as UniProt, Reactome, and RegulonDB.  
16  
17 Next, while experimental data can be provided using the attachment feature of SynBioHub,  
18  
19 we plan to support the new experimental data extensions of SBOL to better link the designs  
20  
21 to their implementations and the data generated through tests, enabling the documentation  
22  
23 of the entire design-build-test process. In a similar fashion, the models and their simulation  
24  
25 experimental data should also be better linked with their designs.  
26

27 As the field of synthetic biology continues to develop we are seeing an increasing number  
28  
29 of different parts collections with different functional characteristics. The idea of a single  
30  
31 repository for all parts is no longer desirable or scalable. SynBioHub supports the federation  
32  
33 of instances. This feature allows individual laboratories and/or research projects to install,  
34  
35 and federate, their own SynBioHub repository, which will be associated to other instances  
36  
37 in the Web of Registries. We advocate that the use of SynBioHub will improve the way  
38  
39 information of synthetic biology constructs is exchanged, and will thus help to overcome the  
40  
41 reproducibility failures within a rapidly growing community.  
42  
43  
44

## 45 **Availability**

46  
47 The SynBioHub source code is available on GitHub under the BSD 2-clause license. SynBio-  
48  
49 Hub is a freely available open source project; any institution or developer can host an instance  
50  
51 of SynBioHub. An example instance of SynBioHub is provided at <http://synbiohub.org>,  
52  
53 which includes the complete iGEM registry of standard biological parts (as of 2016) converted  
54  
55 to SBOL2 format, and provides federated access to the public JBEI-ICE design repository.  
56  
57  
58  
59  
60

1  
2  
3 For programmatic access, communication with SynBioHub is provided as an integral part of  
4 the three major SBOL libraries: libSBOLj (Java), pysbol (Python), and libSBOL (C++).  
5  
6 The use of the libSBOLj SynBioHub integration has been demonstrated in SBOLDesigner<sup>12</sup>  
7  
8 and iBioSim<sup>13</sup> tools to query to find existing parts in SynBioHub and upload new designs  
9  
10 to SynBioHub.  
11  
12  
13

## 14 15 **Methods**

16  
17  
18 SynBioHub is a Node.js application backed by the Virtuoso RDF triplestore. SynBioHub  
19  
20 takes advantage of the RDF semantics of SBOL. Although SBOL has typically been used  
21  
22 as a file format, the SBOL data model is expressed in RDF and thus can be stored in a  
23  
24 triplestore graph database. This approach was first used by the SBOL Stack,<sup>21</sup> and has now  
25  
26 been extended by SynBioHub to accommodate the entire SBOL data model. This provides  
27  
28 a significant advantage in that SynBioHub does not need a custom data storage format;  
29  
30 instead, SBOL can be used as-is and queried as a graph. Figure 6 shows an example of how  
31  
32 SynBioHub queries the RDF triplestore to retrieve a list of SBOL collections in order to  
33  
34 display the collections view.  
35  
36  
37  
38

## 39 **Author Contribution**

40  
41  
42 J.A.M., G.M., I.D.O., and A.W. developed the initial version of the SynBioHub software  
43  
44 previously described in the SBOL Stack paper. The aforementioned authors, alongside  
45  
46 C.J.M., Z.Z., M.Z., and A.G.M., furthered development of SynBioHub as a stand-alone  
47  
48 software project.  
49  
50  
51

## 52 **Conflict of Interest**

53  
54  
55 None declared.  
56  
57  
58

```
1
2
3
4 PREFIX sbol2: <http://sbols.org/v2#>
5 PREFIX dcterms: <http://purl.org/dc/terms/>
6 PREFIX ncbi: <http://www.ncbi.nlm.nih.gov#>
7
8 SELECT ?Collection ?name ?description ?displayId ?version WHERE {
9     ?Collection a sbol2:Collection .
10    FILTER NOT EXISTS { ?otherCollection sbol2:member ?Collection }
11    OPTIONAL { ?Collection dcterms:title ?name . }
12    OPTIONAL { ?Collection sbol2:displayId ?displayId . }
13    OPTIONAL { ?Collection dcterms:description ?description . }
14    OPTIONAL { ?Collection sbol2:version ?version }
15 }
16
17
18
19
```

20 Figure 6: An example of a SPARQL query used by SynBioHub to retrieve the list of root col-  
21 lections. The collections are represented in the triplestore using standard SBOL `Collection`  
22 entities. The query selects all collections, then filters for collections that are not contained  
23 by another collection. The name, displayId, description, and version attributes are retrieved  
24 if present.  
25  
26

## 27 Acknowledgement

28  
29  
30  
31 The authors thank Dr. Nathan Hillson of JBEI for collaboration on the JBEI-ICE integration,  
32 and Randy Rettberg of iGEM Headquarters for providing computational access to the iGEM  
33 Parts Registry. The authors would also like to thank Professor Herbert Sauro, Dr. Bryan  
34 Bartley, and Kiri Choi for developing support within libSBOL and pySBOL to communicate  
35 with SynBioHub. The authors of this work are supported by The Engineering and Physical  
36 Sciences Research Council grants EP/J02175X/1 and EP/N031962/1 (J.A.M. and A.W.)  
37 and EP/R019002/1 (A.G.M.), and the National Science Foundation under Grant No. CCF-  
38 1218095 and DBI-1356041 (C.M., Z.Z., and M.Z.). G. M. is, in part, supported by the  
39 EPSRC grant EP/J02175X/1. J.A.M. is supported by FUJIFILM DioSynth Biotechnologies.  
40 Z.Z. has been supported by Google Summer of Code 2017. Any opinions, findings, and  
41 conclusions or recommendations expressed in this material are those of the author(s) and do  
42 not necessarily reflect the views of the funding agencies.  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

## References

- (1) Church, G. M.; Elowitz, M. B.; Smolke, C. D.; Voigt, C. A.; Weiss, R. Realizing the potential of synthetic biology. *Nature reviews. Molecular cell biology* **2014**, *15*, 289.
- (2) Myers, C. J.; Beal, J.; Goroehowski, T. E.; Kuwahara, H.; Madsen, C.; McLaughlin, J. A.; Mısırlı, G.; Nguyen, T.; Oberortner, E.; Samineni, M.; Wipat, A.; Zhang, M.; Zundel, Z. A standard-enabled workflow for synthetic biology. *Biochemical Society Transactions* **2017**, *45*, 793–803.
- (3) Beal, J.; Weiss, R.; Densmore, D.; Adler, A.; Appleton, E.; Babb, J.; Bhatia, S.; Davidsohn, N.; Haddock, T.; Loyall, J.; Schantz, R.; Vasilev, V.; Yaman, F. An end-to-end workflow for engineering of biological networks from high-level specifications. *ACS synthetic biology* **2012**, *1*, 317–331.
- (4) Litcofsky, K. D.; Afeyan, R. B.; Krom, R. J.; Khalil, A. S.; Collins, J. J. Iterative plug-and-play methodology for constructing and modifying synthetic gene networks. *Nature methods* **2012**, *9*, 1077–1080.
- (5) Goni-Moreno, A.; Carcajona, M.; Kim, J.; Martnez-Garca, E.; Amos, M.; de Lorenzo, V. An implementation-focused bio/algorithmic workflow for synthetic biology. *ACS synthetic biology* **2016**, *5*, 1127–1135.
- (6) Mısırlı, G.; Madsen, C.; de Murieta, I. S.; Bultelle, M.; Flanagan, K.; Pocock, M.; Hallinan, J.; McLaughlin, J. A.; Clark-Casey, J.; Lyne, M.; Micklem, G.; Stan, G.-B.; Kitney, R.; Wipat, A. Constructing synthetic biology workflows in the cloud. *Engineering Biology* **2017**, *1*, 61–65.
- (7) Andrianantoandro, E.; Basu, S.; Karig, D. K.; Weiss, R. Synthetic biology: new engineering rules for an emerging discipline. *Molecular systems biology* **2006**, *2*.

- 1  
2  
3  
4 (8) MacDonald, J. T.; Barnes, C.; Kitney, R. I.; Freemont, P. S.; Stan, G.-B. V. Computational design approaches and tools for synthetic biology. *Integrative Biology* **2011**, *3*,  
5  
6 97–108.  
7  
8  
9  
10 (9) Myers, C. J. *Microbial Synthetic Biology*; Academic, 2013; Vol. 40; pp 177–202.  
11  
12  
13 (10) Otero-Muras, I.; Banga, J. R. Automated Design Framework for Synthetic Biology  
14 Exploiting Pareto Optimality. *ACS Synthetic Biology* **2017**,  
15  
16  
17 (11) Chandran, D.; Bergmann, F. T.; Sauro, H. M. TinkerCell: modular CAD tool for  
18 synthetic biology. *Journal of biological engineering* **2009**, *3*, 19.  
19  
20  
21  
22 (12) Zhang, M.; McLaughlin, J. A.; Wipat, A.; Myers, C. J. SBOLDesigner 2: An Intuitive  
23 Tool for Structural Genetic Design. *ACS Synthetic Biology* **2017**,  
24  
25  
26  
27 (13) Madsen, C.; Myers, C.; Patterson, T.; Roehner, N.; Stevens, J.; Winstead, C. Design  
28 and Test of Genetic Circuits Using iBioSim. *IEEE Design & Test of Computers* **2012**,  
29  
30 *29*, 32–39.  
31  
32  
33  
34 (14) Huynh, L.; Tagkopoulos, I. Optimal part and module selection for synthetic gene circuit  
35 design automation. *ACS synthetic biology* **2014**, *3*, 556–564.  
36  
37  
38  
39 (15) Nielsen, A. A.; Der, B. S.; Shin, J.; Vaidyanathan, P.; Paralanov, V.; Strychalski, E. A.;  
40 Ross, D.; Densmore, D.; Voigt, C. A. Genetic circuit design automation. *Science* **2016**,  
41 *352*, aac7341.  
42  
43  
44  
45  
46 (16) Bhatia, S.; Densmore, D. Pigeon: a design visualizer for synthetic biology. *ACS syn-*  
47 *thetic biology* **2013**, *2*, 348–350.  
48  
49  
50  
51 (17) Der, B. S.; Glassey, E.; Bartley, B. A.; Enghuus, C.; Goodman, D. B.; Gordon, D. B.;  
52 Voigt, C. A.; Goroehowski, T. E. DNAplotlib: programmable visualization of genetic  
53 designs and associated data. *ACS synthetic biology* **2016**,  
54  
55  
56  
57  
58  
59  
60

- 1  
2  
3 (18) McLaughlin, J. A.; Pocock, M.; Mısırlı, G.; Madsen, C.; Wipat, A. VisBOL: web-based  
4 tools for synthetic biology design visualization. *ACS synthetic biology* **2016**, *5*, 874–876.  
5  
6  
7  
8 (19) Peccoud, J.; Anderson, J. C.; Chandran, D.; Densmore, D.; Galdzicki, M.; Lux, M. W.;  
9 Rodriguez, C. A.; Stan, G.-B.; Sauro, H. M. Essential information for synthetic DNA  
10 sequences. *Nature biotechnology* **2011**, *29*, 22–22.  
11  
12  
13  
14 (20) Ham, T. S.; Dmytriv, Z.; Plahar, H.; Chen, J.; Hillson, N. J.; Keasling, J. D. De-  
15 sign, implementation and practice of JBEI-ICE: an open source biological part registry  
16 platform and tools. *Nucleic acids research* **2012**, *40*, e141–e141.  
17  
18  
19  
20  
21 (21) Madsen, C.; McLaughlin, J. A.; Mısırlı, G.; Pocock, M.; Flanagan, K.; Hallinan, J.;  
22 Wipat, A. The SBOL Stack: a platform for storing, publishing, and sharing synthetic  
23 biology designs. *ACS synthetic biology* **2016**, *5*, 487–497.  
24  
25  
26  
27  
28 (22) Lu, G.; Moriyama, E. N. Vector NTI, a balanced all-in-one sequence analysis suite.  
29 *Briefings in bioinformatics* **2004**, *5*, 378–388.  
30  
31  
32  
33 (23) Herscovitch, M.; Perkins, E.; Baltus, A.; Fan, M. Addgene provides an open forum for  
34 plasmid sharing. *Nature biotechnology* **2012**, *30*, 316–317.  
35  
36  
37  
38 (24) Galdzicki, M. et al. The Synthetic Biology Open Language (SBOL) provides a commu-  
39 nity standard for communicating designs in synthetic biology. *Nat. Biotechnol.* **2014**,  
40 *32*, 545–550.  
41  
42  
43  
44 (25) Roehner, N. et al. Sharing structure and function in biological design with SBOL 2.0.  
45 *ACS synthetic biology* **2016**, *5*, 498–506.  
46  
47  
48  
49 (26) Beal, J. et al. Synthetic Biology Open Language (SBOL) Version 2.1.0. *Journal of*  
50 *Integrative Bioinformatics* **2016**, *13*, 30–132.  
51  
52  
53  
54 (27) Lebo, T.; Sahoo, S.; McGuinness, D.; Belhajjame, K.; Cheney, J.; Corsar, D.; Garijo, D.;

- 1  
2  
3 Soiland-Reyes, S.; Zednik, S.; Zhao, J. Prov-o: The prov ontology. *W3C recommenda-*  
4 *tion* **2013**, *30*.
- 5  
6  
7  
8 (28) Quinn, J. Y. et al. SBOL Visual: A Graphical Language for Genetic Designs. *PLOS*  
9 *Biology* **2015**, *13*, 1–9.
- 10  
11  
12 (29) Krebs, O.; Wolstencroft, K.; Stanford, N. J.; Morrison, N.; Golebiewski, M.; Owen, S.;  
13 Nguyen, Q.; Snoep, J. L.; Mueller, W.; Goble, C. A. FAIRDOM approach for semantic  
14 interoperability of systems biology data and models. ICBO. 2015.
- 15  
16  
17 (30) Shadbolt, N.; Berners-Lee, T.; Hall, W. The semantic web revisited. *IEEE intelligent*  
18 *systems* **2006**, *21*, 96–101.
- 19  
20  
21 (31) Eilbeck, K.; Lewis, S. E.; Mungall, C. J.; Yandell, M.; Stein, L.; Durbin, R.; Ash-  
22 burner, M. The Sequence Ontology: a tool for the unification of genome annotations.  
23 *Genome Biology* **2005**, *6*, R44.
- 24  
25  
26 (32) Zhang, Z.; Nguyen, T.; Roehner, N.; Misirli, G.; Pocock, M.; Oberortner, E.; Sami-  
27 neni, M.; Zundel, Z.; Beal, J.; Clancy, K.; Wipat, A.; Myers, C. J. libSBOLj 2.0: A  
28 Java Library to Support SBOL 2.0. *IEEE Life Sci Lett* **2016**, *1*, 34–37.
- 29  
30  
31 (33) Belleau, F.; Nolin, M.-A.; Tourigny, N.; Rigault, P.; Morissette, J. Bio2RDF: towards a  
32 mashup to build bioinformatics knowledge systems. *Journal of biomedical informatics*  
33 **2008**, *41*, 706–716.
- 34  
35  
36 (34) Misirli, G.; Hallinan, J.; Pocock, M.; Lord, P.; McLaughlin, J. A.; Sauro, H.; Wipat, A.  
37 Data integration and mining for synthetic biology design. *ACS synthetic biology* **2016**,  
38 *5*, 1086–1097.
- 39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

1  
2  
3 **Graphical TOC Entry**  
4  
5  
6

