ACS Synthetic Biology

Research Article

pubs.acs.org/synthbio

Data Integration and Mining for Synthetic Biology Design

Göksel Mısırlı,[†] Jennifer Hallinan,[†,∥] Matthew Pocock,[†,‡] Phillip Lord,[†] James Alastair McLaughlin,[†] Herbert Sauro,[§] and Anil Wipat*[,†]

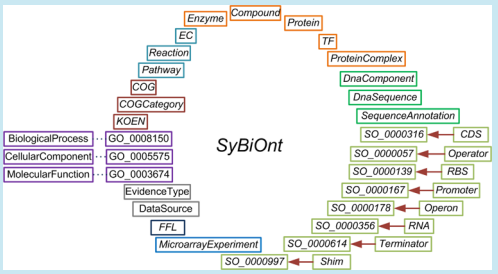
[†]School of Computing Science, Newcastle University, NE1 7RU Newcastle upon Tyne, United Kingdom

[‡]Turing Ate My Hamster Ltd, NE27 0RT Newcastle upon Tyne, United Kingdom

[§]Department of Bioengineering, University of Washington, Seattle, Washington 98105, United States

Supporting Information

ABSTRACT: One aim of synthetic biologists is to create novel and predictable biological systems from simpler modular parts. This approach is currently hampered by a lack of well-defined and characterized parts and devices. However, there is a wealth of existing biological information, which can be used to identify and characterize biological parts, and their design constraints in the literature and numerous biological databases. However, this information is spread among these databases in many different formats. New computational approaches are required to make this information available in an integrated format that is more amenable to data mining. A tried and tested approach to this problem is to map disparate data sources into a single data set, with common syntax and semantics, to produce a data warehouse or knowledge base. Ontologies have been used extensively in the life sciences, providing this common syntax and semantics as a model for a given biological domain, in a fashion that is amenable to computational analysis and reasoning. Here, we present an ontology for applications in synthetic biology design, SyBiOnt, which facilitates the modeling of information about biological parts and their relationships. SyBiOnt was used to create the SyBiOntKB knowledge base, incorporating and building upon existing life sciences ontologies and standards. The reasoning capabilities of ontologies were then applied to automate the mining of biological parts from this knowledge base. We propose that this approach will be useful to speed up synthetic biology design and ultimately help facilitate the automation of the biological engineering life cycle.



KEYWORDS: *synthetic biology, data integration, data mining, ontologies, Semantic Web, automated identification of biological parts*

One of synthetic biology's primary aims is the design of predictable biological systems, thus allowing larger and more complex systems to be successfully designed and built.[1−3] Like most engineering disciplines, in synthetic biology complex synthetic biological systems are typically developed via the composition of simple, modular components.[4−6] In order to ensure that the resulting synthetic systems behave in a predictable fashion, the parts and modules used for biological systems engineering, and the context in which they are deployed, need to be well-understood and well-characterized.[7] However, the lack of well-characterized parts and modular devices, confounded by our limited understanding of biology, is widely recognized as limiting the scale and complexity of current engineered biological systems.

The identification, characterization, and development of new modular parts, devices, and systems requires access to large amounts of biological knowledge.[5,8−11] This knowledge needs to be gathered, integrated, and made accessible to system designers. Furthermore, this knowledge also needs to be made available in a computationally tractable fashion in order to support automation and computer aided design.

Providing such information is challenging. Information is scattered over a range of different databases, which use different formats and have different semantics.[12−14] A major challenge to synthetic biology is bringing together complex, heterogeneous, disparate data sets in a form that will best inform the synthetic biology design process. Moreover, these integrated data sets need to be assembled in such a way that they are easily computationally mined.[15−17] Data mining requires data integration techniques that align disparate representations and semantics to produce a unified domain model. This model can then be mined to extract the necessary information without the need to repeatedly visit large numbers of separate data resources.[18]

The integration of biological data is still a major research challenge and has been the focus of an active research effort in the fields of bioinformatics and systems biology. Traditional methods include data warehousing,[18−20] where data from multiple databases is drawn together into a single database. In another approach, termed federated data integration, the data remain in separate databases that are queried in parallel, and the results are integrated before being returned to the user.[21−25] One of the major problems in data integration is the lack of

Special Issue: Synthetic Biology in Europe

Received: December 21, 2015
Published: April 25, 2016

DOI: 10.1021/acssynbio.5b00295
ACS Synth. Biol. 2016, 5, 1086−1097

agreement on data formats and variation in the meaning of the data (termed semantics). The value of semantically well-defined electronic representations of data for their integration is now widely recognized,[21,22,26] and a technology to exploit unified semantics on the Internet, called Semantic Web technology, has been developed.[27] Semantic Web encourages the use of common data representation formats for data, allowing data to be shared across boundaries and easing the integration process.[18,28] Ontologies[29] underpin the Semantic Web concept since they can be used to standardize data representation by adding computationally tractable meaning to the syntax of data entities and the relationships between them.[27,30] In this respect, ontologies are increasingly being recognized as a powerful approach to identifying, integrating, and organizing large amounts of complex data.[25,31−33]

Semantic Web technologies have become increasingly popular for modeling, accessing, and exchanging data in the life sciences.[27,34] Numerous databases now provide data in the Resource Description Framework (RDF; http://www.w3.org/TR/rdf-syntax-grammar) format.[21] These databases use standard terms from biological ontologies[31,32] for the annotation of biological concepts and their interactions. Furthermore, off-the-shelf tools that support Semantic Web technologies are used for the storage[22] and querying[27] of, and reasoning with, biological data.[25,35,36]

These technologies are also increasingly being used within the synthetic biology community. Tool and part catalogue developers, representatives from industry and academics, have agreed on a format for the electronic exchange of information about biological designs and their component parts. This format is called Synthetic Biology Open Language (SBOL; http://sbolstandard.org/development/developers). (At the time of writing, SBOL developers are around 120 members from over 50 institutions in 15 countries.)[37] In SBOL, version 1.0, the core data model is small, focusing upon the exchange of sequence-based information. The recently released version 2.0[38] extends the initial data model to capture additional types of design components such as proteins and compounds and the functional relationships between them.

SBOL is valuable for promoting the exchange of synthetic biology designs, for example, between part repositories and design tools. Many SBOL compliant tools are available, and many more are under development (http://sbolstandard.org/software/tools/). For example, existing data in SBOL format, describing BioBricks from the Registry of Standard Biological Parts,[39] have been made available using an RDF triple store, enabling SPARQL querying of the parts.[40] The utility of SBOL to facilitate data exchange between different tools and different users to carry out tasks that could not be achieved using a single tool was demonstrated recently. In this workflow, SBOL was used to pass designs between a range of different tools to model and combinatorially design a genetic toggle switch for *Escherichia coli*, which was then codon optimized, and the resulting designs were stored in SBOL compliant repositories.[41]

SBOL utilizes standard terms and a standard syntax (based on RDF) to describe synthetic biology designs. The semantics of SBOL entities are described using terms from external ontologies and controlled vocabularies. These terms are useful to unambiguously represent information about biological parts. Ontologies can also be effectively used in other languages and tools for synthetic biology, particularly to help facilitate the development of automated design processes. Using ontologies, large amounts of data about biological parts and constraints about how they work can be presented in a form that is readily utilizable by computational design tools. The availability of biological knowledge in a computationally tractable manner is important to enable the development of tools that will aid in the design of biologically feasible systems. In the process of the ontological modeling of data, a conceptual language is used to define objects and their relationships in order to make data accessible to a wide range of computational tools. The use of logics[34] allows reasoning over the data by employing reasoners, which are used to make implicit knowledge explicit through ontological queries. Although the use of these queries, together with reasoners, can be a powerful tool to mine different types of biological parts from semantically enriched integrated data sets, this approach has not been applied in synthetic biology to the best of our knowledge.

In this work, we demonstrate how designs for parts and devices can be derived from integrated data sources using Semantic Web technology to enhance the synthetic biology design process. We build upon our previous work in the integration of data using a warehousing approach[42] to produce a semantically well-defined knowledge base. We employ the W3C standard specification, the Web Ontology Language (OWL; http://www.w3.org/2004/OWL), to describe biological data and the relationships between those data items that are relevant to the design of synthetic biology parts. The result is a knowledge base to support both manual and automated synthetic biology design.

In order to facilitate the development of this knowledge base, it was necessary to define the metadata underpinning the data entities and the relationships between them in a semantically well-defined way. We therefore developed an ontology (called SyBiOnt) to model the domain of genetic designs in synthetic biology. Information about data items and their relationships was stored as RDF in the form of subject−predicate−object triples in a triple store database (see the Supporting Information). We demonstrated how this data resource could be queried using semantic reasoning and biologically rich queries to mine the knowledge base for new genetic parts and devices. Finally, we exported novel parts represented in the form of the standard interchange format, SBOL.[41]

1. RESULTS

1.1. SyBiOnt Ontology. The basic biological parts used in the bottom-up design of synthetic systems include genetic features such as promoters, coding sequences (CDSs), ribosome binding sites (RBSs), terminators, and operators.[7] The relationships among these parts and the gene products they encode, such as proteins, RNAs, transcription factors (TFs), and enzymes, need to be captured in order to design genetic circuits. Moreover, the incorporation of additional information about biological pathways and gene function is necessary to identify appropriate biological parts. Our goal when creating SyBiOnt was to allow a data definition framework to formalize the representation of the information that describes these parts and the relationships among them. SyBiOnt was designed to allow the incorporation of further information in the form of annotations that add extra, useful knowledge such as gene function. The ontology was developed using OWL semantics. The rich expressivity of OWL enables the construction of complex computational queries and automated reasoning across the integrated data.

When using SyBiOnt, types of biological entities, such as protein, CDS, and pathway, are represented as the first level

DOI: 10.1021/acssynbio.5b00295
ACS Synth. Biol. 2016, 5, 1086−1097

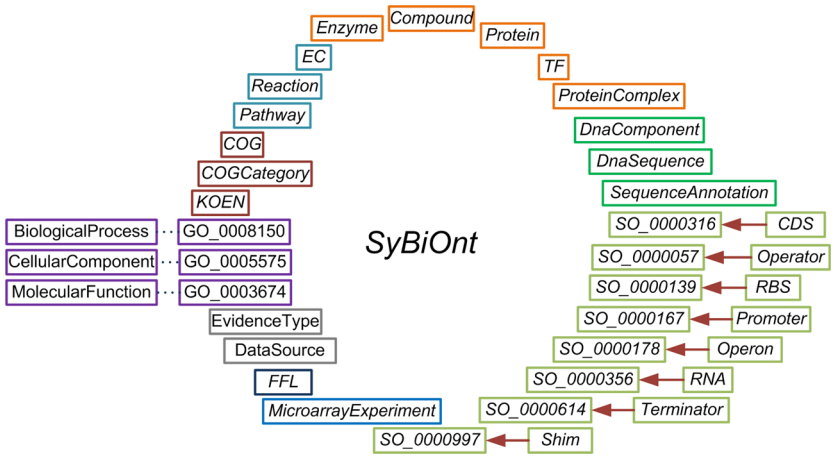


Figure 1. Classes that represent the types of biological entities and classes from GO, SO, and SBOL ontologies included in SyBiOnt. Solid lines represent the subclassing relationship, with the arrow pointing at the parent classes, and dashed lines show the equivalent classes.

superclasses that are subclasses of `owl:Thing` in the ontology (Figure 1). SyBiOnt also includes classes for reactions, pathways, microarray experiments, feed-forward loops, data sources, and evidence types. Relationships among these biological entities are also modeled. Such relationships include protein−protein interactions, formation of protein complexes, enzyme interactions with compounds, compound transportation into cells, and TFs binding to DNA sequences.

The first level classes representing sequence-based biological entity types, such as `Promoter` and `CDS`, are linked to terms from Sequence Ontology (SO)[32] through subclassing. For example, `Promoter` is a subclass of the SO promoter term `SO_0000167`. Other molecules such as proteins, TFs, RNAs, enzymes, protein complexes, and compounds are also modeled as OWL classes. In SyBiOntKB, TFs and their corresponding proteins or RNAs are modeled as equivalent classes and can therefore be used interchangeably in OWL queries. Enzymes are special proteins that catalyze reactions and are modeled as subclasses of the corresponding `Protein` classes.

Classes that are used to classify or place restrictions on classes representing physical entities include enzyme classifications, KEGG ortholog enzymes, molecular functions, biological processes, cellular components, and the Clusters of Orthologous Groups (COG) classes[43] and categories. The classes `MolecularFunction`, `CellularComponent`, and `BiologicalProcess` are equivalent to Gene Ontology (GO)[31] classes `molecular_function` (`GO_0003674`), `biological_process` (`GO_0008150`), and `cellular_component` (`GO_0005575`).

1.2. Development of the SyBiOnt Knowledge Base (SyBiOntKB). As an example of the use of the SyBiOnt ontology, we used the formal data definition framework provided to develop a knowledge base, termed SyBiOntKB, to capture major aspects of the cell biology of *Bacillus subtilis* in a computationally amenable form. The data to populate this knowledge base were sourced from the previously integrated BacillOndex data set,[42] which includes information from BacilluScope,[44] DBTBS,[45] the Kyoto Encyclopedia of Genes and Genomes (KEGG),[12] KEGG Expression,[46] STRING,[14] GO, and GO annotations.[47]

When building an ontology, entities can be modeled as classes or as individuals. In this work, we modeled entities as classes, since classes are beneficial for representing high-level common knowledge in a way that allows automated reasoning and inference.[34] The entities modeled in SyBiOntKB, such as CDSs and proteins, do not represent individual molecules but types of molecules that exist in all cells. Such molecules were therefore modeled with classes. These classes can then be instantiated by individuals. This approach has previously been applied to the modeling of knowledge in Open Biological and Biomedical Ontologies (OBO) and in biomedical knowledge bases that are annotated using the classes from OBO ontologies.[25,34,48] For example, the Spo0A protein entity in SyBiOntKB represents a class to which all individual Spo0A protein molecules belong. By relating the Spo0A class to the Spo0B protein class using the "is phosphorylated by" restriction, all Spo0A individuals inherit this relationship. Hence, SyBiOnt and the knowledge base models described shared features of proteins, but they do not describe all properties of individual protein molecules.

In SyBiOntKB, restrictions were usually expressed using OWL's *someValuesFrom* (*some*) restriction.[49] For a class `A`, `(r some B)` restriction means that for every instance of `A` there is an instance of `B` related to `A` by `r`. However, such a restriction does not rule out the possibility of an individual being in the same relationship to instances of other classes. For example, a restriction can be used to say that the Spo0A TF binds to the kinA operator. SyBiOntKB represents this restriction as "binds to some kinA operator" on the Spo0A class. The statement does not specify whether or not there are additional operators to which the TF binds. This approach facilitates the modeling of biological entities without making overly restrictive or specific claims.[35] Attributes of biological entities were modeled using OWL's `hasValue`(*value*) restrictions.

The resulting SyBiOntKB for *B. subtilis* includes 42259 OWL classes, with 41 objects, 21 datatypes, and 26 annotation properties. There are 269726 `SubClassesOf`, 386 `EquivalentClass`, 169 `DisjointClass`, and 274003 `AnnotationAssertion` axioms. As the ontology conforms to RDF and OWL standards, it can be manipulated using existing ontology editors such as Protégé (http://protege.stanford.edu), and information can be extracted using reasoners such as Pellet[50] and HermiT.[51] The ontology is also available at an RDF repository to allow the querying of information using standard SPARQL queries (see the Supporting Information). The base URI of the ontology is http://w3id.org/synbio/ont.

DOI: 10.1021/acssynbio.5b00295
ACS Synth. Biol. 2016, 5, 1086−1097

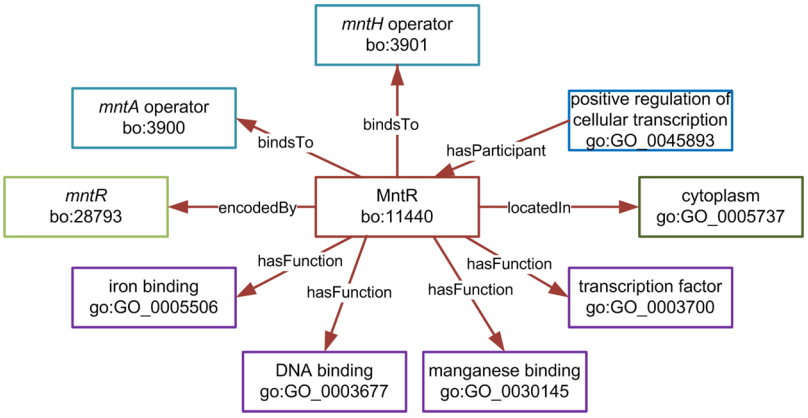


Figure 2. An example of protein relationships captured in SyBiOntKB. The diagram shows a subset of the relationships for the protein MntR, modeled as restrictions on object properties, such as `encodedBy` and `hasFunction`. The information includes the molecular functions of the protein, where it is located, a biological process in which the protein participates, the encoding CDS, and its binding sequences.

Figure 2 shows a subset of information about the relationships of the MntR protein as represented using SyBiOntKB. This information includes the molecular functions of the protein, its location, the biological processes in which the protein participates, the CDS encoding this protein, and any DNA binding sequences.

1.3. Testing the Competency of SyBiOnt. The scope of ontologies can be identified with a set of questions, called competency questions.[52] These questions do not have to be exhaustive and can be written informally, but they serve to test whether an ontology contains enough detailed information for its intended application. We used SyBiOntKB to demonstrate the validity of SyBiOnt.

SyBiOnt was designed to answer competency questions that are of interest in the design of synthetic biological systems. With this requirement in mind, a number of competency questions were devised. These questions also serve to demonstrate the power of this approach for deriving designs for engineered biological systems. These questions, and queries that we would make over the ontology, are listed as follows:

- *Which parts are SigmaA type promoters?* The SigmaA sigma factor is the `TF` with the `accession` of "BSU25200" and `binds to Promoters`, which can be identified as `SigmaA Promoters`.
- *Which promoters are constitutive?* `SigmaA Promoters` that do not have any `Operators` can be `Constitutive Promoters`.
- *Which parts can be used as inducible promoters?* `Operators` have `regulation type` restrictions to indicate whether they are used positively or negatively in regulating gene expression. A `Promoter` with one `Operator` part that has the "Positive" `regulation type` restriction is an `Inducible Promoter`.
- *Which parts are SigmaA type inducible promoters?* `Promoters` that are both `subclasses` of `SigmaA` and `Inducible Promoters` are candidate `Promoters`.
- *Which parts are regulated by the MntR TF?* MntR `binds to` some (*mntA* and *mntH*) `Operators`.
- *What are the nucleotide sequences that the Spo0A TF binds to?* `Operators` that are `bound by` the Spo0A `TF` have restrictions on the `nucleotide sequence` property.
- *Which parts encode two-component systems (TCSs)?* These parts are `CDSs encoding Proteins` that `have functions` of kinase activity and response regulator activity. GO classes `GO_0000155` and `GO_0000156`, respectively, represent these functions.
- *Which parts can be used to upregulate the production of ammonium?* The `Compound` ammonia with the `accession` of "C00014" is `produced by Reaction` RN:R00131, which `consumes` the `Compound` carbamide (C00086). Carbamide is `produced by` a `Reaction` that is `catalyzed by` an `Enzyme`, which is a `subclass` of a `Protein encoded by` the *argI* `CDS` with the `accession` BSU40320.
- *Which pathways should be targeted for the overproduction of ammonium?* Ammonium is `produced` by `Reactions` that are `member of` the Arginine and proline metabolism and Purine metabolism `Pathways`.
- *How can the Spo0A protein, the master regulator of sporulation, be phosphorylated to trigger sporulation?* Spo0A is `phosphorylated by` the KinC and Spo0B `Proteins`. The Spo0B `Protein` is `phosphorylated by` Spo0F `Protein`, which is further `phosphorylated by` the KinA and KinB `Proteins`.
- *What are the possible NAND gate promoters?* `NAND gate Promoters` can be searched for in the list of `Promoters` that have two `Operator` parts with Negative `regulation type` restrictions.
- *Which parts should be upregulated to increase mannose compound transport to the cells?* The "D-mannose 6-phosphate `Compound` with the `accession` of C00275 `interacts with` a `ProteinComplex`. ManP and LevF `Proteins` are part of this complex.

1.4. Mining SyBiOntKB for Biological Parts. SyBiOnt can be used to answer certain types of questions in a richer fashion than a conventional relational database. As an example, we showed how automated reasoning over this ontology could be used to identify parts and devices that could be used in synthetic designs. Particularly, we focused on the automated identification of promoters that could be used as logic gates (such as inducible or repressible), the building blocks of many synthetic biology designs. We then demonstrated the mining of CDS parts based on the molecular functions of their encoded products. In principle, the textual descriptions of classes from

DOI: 10.1021/acssynbio.5b00295
ACS Synth. Biol. 2016, 5, 1086−1097

```
Class: NegativelyRegulatedOperator
   EquivalentTo:
      Operator
         and (NA some PlainLiteral)
         and (regulationType value "Negative")
       SubClassOf:
          Operator

Class: PositivelyRegulatedOperator
   EquivalentTo:
      Operator
         and (NA some PlainLiteral)
         and (regulationType value "Positive")
      SubClassOf:
         Operator
```

Figure 3. Class definitions for operator classification in the Manchester OWL syntax. `NA` indicates the nucleic acid sequence, and `regulationType` indicates the regulation type. Operators with known sequences are therefore classified according to their regulation type restrictions.

```
Promoter
  and (has_part exactly 1 Operator)
  and (has_part exactly 1 PositivelyRegulatedOperator)
```

Figure 4. Inducible promoter class definition. Promoters with one operator for an activator are classified as inducible promoters.

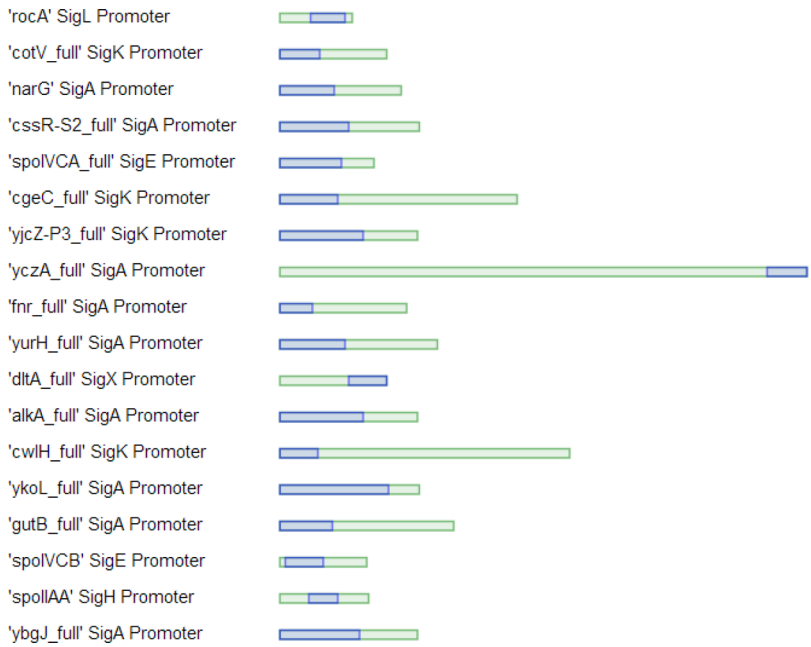


Figure 5. Some of the inducible promoters mined from SyBiOntKB. The outer green rectangles and inner blue rectangles represent the promoters and TF binding sites, respectively. The length of a box is proportional to the corresponding promoter's sequence length.

the ontology could be read by eye and used by humans to make assertions manually, but the use of automated reasoning vastly speeds up the process. Automated reasoning is a much faster computational way of extracting information from the ontology.

The automating reasoning process requires two steps that were carried out as follows. First, we specified a question by stating the conditions that must be fulfilled to provide answers to this question.[53] The logical reasoner was then used to search the ontology to provide these answers in a rapid and efficient manner. For example, the `Protein and (hasFunction some "kinase activity")` query was used to classify kinases (the GO term for kinase activity is `GO_0000155`). In practice, to provide the correct format for the reasoner, the query was implemented as an OWL class with the necessary and sufficient conditions,[35] which requires that all subclasses must be Proteins and must have the `hasFunction "kinase activity"` restriction.

In order to classify promoters that can be used as logic gates, first, their operator subparts were classified. This process requires information about whether an operator is involved in negative or positive regulation. To enable operator classification in SyBiOnt, operator classes have *hasValue* restrictions on the `regulationType` property that specify that binding is for either activation or repression with a `regulationType` value of Negative or Positive, respectively (Figure 3). In total,

DOI: 10.1021/acssynbio.5b00295
ACS Synth. Biol. 2016, 5, 1086−1097

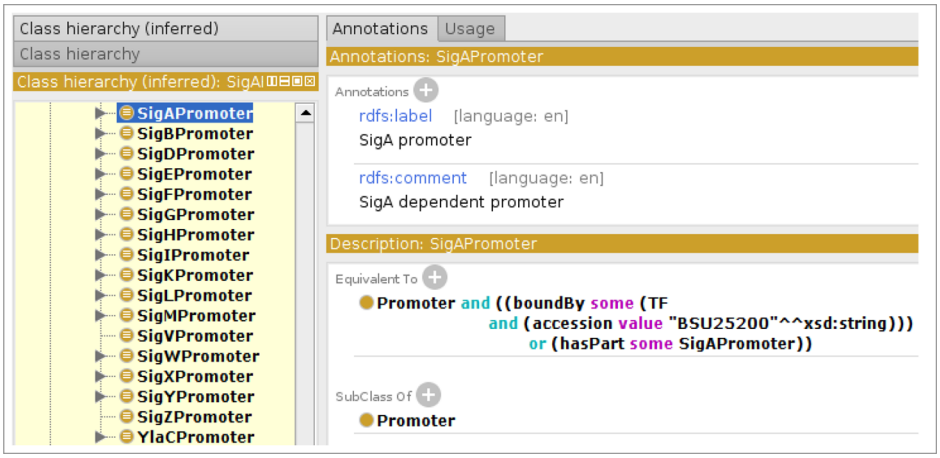


Figure 6. Classification of promoters based on sigma factors. The right pane displays the definition of the `SigAPromoter` class used for the classification of SigA promoters.

```
CDS and
   (encodes some (Protein and
   (bindsTo some NegativelyRegulatedOperator)))
```

Figure 7. OWL expression for the `RepressorEncodingCDS` defined class. A CDS that encodes for a protein binding to at least one repressor site is classified as `RepressorEncodingCDS`.

333 repressor and 222 activator sites, with known nucleotide sequences, were classified.

Promoters were also classified in a similar fashion, according to their regulation type. Classes were defined for inducible, repressible, and constitutive promoters. In addition, a range of classes were defined to classify promoters according to their sigma factors.

The positions of the operators in promoters and the cooperativity in TF binding results in different transcriptional logic gate behaviors. Transcriptional logic gates are useful parts for circuit implementation in synthetic biology. For example, a promoter with two activator sites can function as an AND or an OR gate.[54−56] Conversely, a promoter with two repressor sites can function as a NAND or a NOR gate.[54,56,57] Thus, we attempted to use reasoning over the ontology to find examples of promoters from *B. subtilis* that could be used as the basis of logic gates.

First, we set out to mine for inducible promoters with a single operator acting as an activator site. Therefore, inducible promoters were identified that possessed only one operator for an activator TF (Figure 4). In total, 51 promoters were identified. A subset of these promoters is shown in Figure 5. This subset was termed `InduciblePromoter`.

Second, 85 repressible promoters that bind a repressor using a single operator (and act as one-input inverters) were mined by using reasoning over SyBiOntKB and classified as `RepressiblePromoter`. The corresponding class definition for mining the ontology therefore specified that a repressible promoter has only one operator for a repressor TF (Figure S3).

Third, we mined the ontology for transcriptional AND gates or OR gates. These kinds of gates can be formed from promoters with multiple activator sites. For this exercise, we limited the scope to two activator sites. In order to mine for examples of promoters that could act as AND or OR gates, a class `InduciblePromoterWith2Operators` was defined and used to identify 15 promoters that possessed two activator binding sites (Figure S4).

NAND and NOR gates can be also constructed from promoters with multiple operators. In this case, these operators correspond to repressor binding sites. We therefore defined a `RepressiblePromoterWith2Operators` class to identify promoters of this type. Twenty-five promoters that possessed two repressor binding sites were identified in SyBiOntKB (Figure S5).

Promoters were also classified based on the sigma factors of RNA polymerase that can be used to add specificity for a given promoter class. For example, the SigAPromoter is a promoter to which the RNA polymerase subunit sigma A binds. Sigma factors in SyBiOntKB are represented as transcription factors and can be identified using their accession identifiers (e.g., BSU25200 for sigma A). Such a promoter may be a core SigA promoter or a composite promoter that includes a core SigA promoter (Figure 6). Similarly, classes were defined for other sigma factors. In total, 465 SigA, 67 SigB, 33 SigD, 97 SigE, 30 SigF, 63 SigG, 31 SigH, 1 SigI, 71 SigK, 10 SigL, 8 SigM, 0 SigV, 39 SigW, 16 SigX, 2 SigY, 0 SigZ, and 1 YlaC type promoters were classified.

Constitutive promoters are dependent only upon RNA polymerases, and their definition does not rely upon information about transcriptional regulation. In *B. subtilis*, SigA promoters without any TF binding sites are constitutive promoters. To classify constitutive SigA promoters, the `ConstitutiveSigAPromoter` class was defined as a subclass of both the `SigAPromoter` and `ConstitutivePromoter` classes. In addition, a restriction class was added to specify that these promoters cannot have any operators (Figure S8). As a result, 311 constitutive promoters were identified.

Many synthetic biology projects focus upon the use of regulators such as transcriptional activators and repressors[56,58−61] and sensory systems such as two-component systems (TCSs).[62−66] Similar to TFs, TCSs have the potential

DOI: 10.1021/acssynbio.5b00295
ACS Synth. Biol. 2016, 5, 1086−1097

```
CDS and
   (encodes some (Protein and
   (hasFunction some go:GO_0000155)))
```

Figure 8. OWL expression for the KinaseEncodingCDS defined class. A CDS that encodes for a protein that has function go:GO_0000155 is classified as KinaseEncodingCDS.

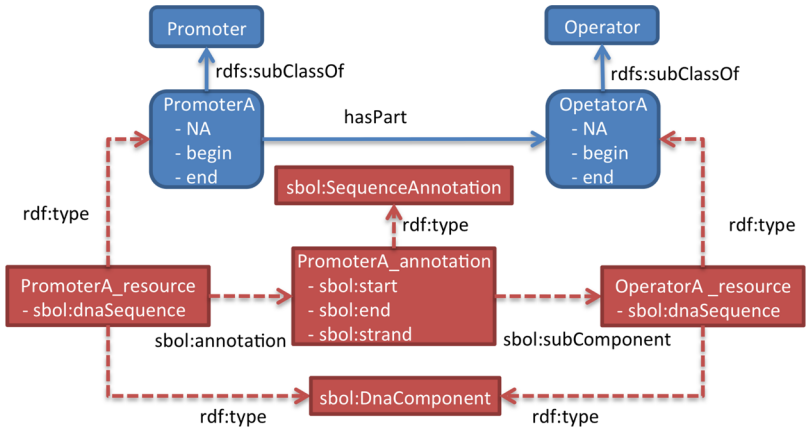


Figure 9. Representation of the mapping for a simple promoter that contains an operator. Blue boxes with round corners and straight lines represent ontology classes and their relationships. Red boxes and dashed lines represent SBOL resources and their relationships.

to be used as modular biological parts.[62] These systems can be introduced into other host cells that do not have any analogous systems, hopefully providing well-isolated circuits. To contribute to the list of available CDS parts in these categories, CDSs were classified as transcriptional activator-, transcriptional repressor-, kinase-, or response regulator-encoding CDSs. In order to classify CDS parts that encode transcriptional repressors, the RepressorEncodingCDS class was defined as a CDS that codes for a protein that binds to at least one repressor site (Figure 7). These repressor sites have known nucleotide sequences. Therefore, reliable parts that encode and provide binding sites can be retrieved as pairs. Similarly, the ActivatorEncodingCDS class was defined as a CDS that codes for a protein that binds to at least one activator site (Figure S9). Using the reasoners, 44 activator- and 55 repressor-encoding CDSs were identified.

When using SyBiOnt, CDSs that encode TCS kinase and response regulators are classified based on the relevant GO terms. A CDS that encodes a protein that has the GO_0000155 molecular function (two-component system sensor activity) is classified as a KinaseEncodingCDS class (Figure 8). Similarly, a CDS that encodes a protein that has the GO_0000156 molecular function (two-component response regulator activity) is classified as a ResponseRegulator-EncodingCDS class. Using this approach, in total, 40 kinase- and 38 response regulator-encoding CDSs were identified.

SyBiOnt includes a variety of information about biological entities, attributes, and relationships that can be used to automate the identification of CDS parts. COG numbers and the GO molecular function, biological process, and cellular component terms can be used to classify gene products and hence to find the CDS that encodes a given protein with a given function. Furthermore, classes such as RNA, TF, and Enzyme may be used to specify the roles of gene products more explicitly.

1.5. Mapping SyBiOnt to the SBOL Data Model. One of the advantages of representing knowledge in a computationally tractable format is that it can be readily exported in a different exchange languages. In order to demonstrate this process, we exported the SyBiOnt ontology in SBOL format. While there are a range of data formats capable of representing genetic designs (Genbank, EMBL, etc.), we chose SBOL since it was developed specifically for representing synthetic biology designs. The ability to represent these designs in standard formats makes it easier to exchange designs among these tools, part catalogues, and synthesis companies, ultimately enhancing reproducibility of synthetic biology designs. SBOL was developed to address this issue and provide a standard format for the exchange of synthetic biology designs.[37] Sequence-based features and their part-whole hierarchy of part composition were expressed in SBOL. These SBOL encoded parts could then be exported and imported for reincorporation into SyBiOntKB as required.

In the SyBiOnt ontology, promoters, CDSs, terminators, shims, RBSs, and operators are basic biological parts and have corresponding OWL classes, with specified nucleotide sequence restrictions. These sequence features were modeled with the DnaComponent class of SBOL. In SyBiOnt, some sequence-based features such as operator sites were modeled, via SBOL annotations, as part of other features. For example, a promoter with two operator sites can be modeled as a DnaComponent with two annotations that have operators as subcomponents.

Sequence annotations in SBOL include the start and end positions of sequence features. Although such information does not exist directly in the ontology, it can be inferred from the chromosomal start and end positions. SBOL's DnaSequence class is used to represent nucleotide sequences. Although SBOL provides terms to describe the relationships between sequence features and their sequence annotations, information about these sequence features is represented with RDF resources that represent individual sequence features.[40] Therefore, individuals representing sequence features were created and mapped to the SBOL data model in SyBiOnt.

Rule-based mapping is one way of presenting the data from SyBiOnt in SBOL format. Example rules include "If a SyBiOnt class inherits from Promoter, then it is associated with a

DOI: 10.1021/acssynbio.5b00295
ACS Synth. Biol. 2016, 5, 1086−1097

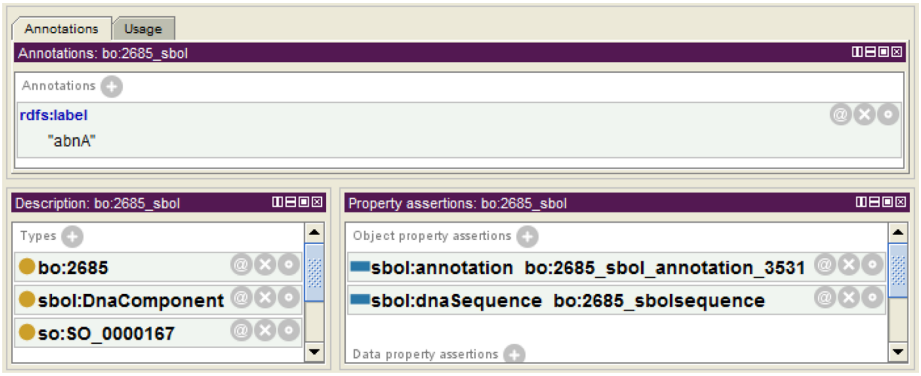


Figure 10. Shows an example of a promoter in SBOL format. The resource is of both types `sbol:DnaComponent` and `so:SO_0000167` (promoter sequence), classes that are both mandatory in the SBOL model.

DnaComponent individual in the SBOL representation" and "If a Promoter has Operator parts, then the DnaComponent associated with this Promoter has Operator annotations that are also DnaComponents" (Figure 9).

Classes were mapped using five rules:

- If a class is a subclass of `Promoter`, `RBS`, `Operator`, `Shim`, `Terminator`, or `CDS`, then there is an individual resource that has the type `sbol:DnaComponent`. Another type is the class from SyBiOnt, for which the individual is created.
- The `sbol:dnaSequence` property identifies the resource that includes the nucleotide sequence of an SBOL individual using the `sbol:nucleotides` property. The sequence is extracted from the restriction on the `NA` property of the class from SyBiOnt.
- A sequence feature individual that is part of a `DnaComponent` individual is also a type of `sbol:DnaComponent`.
- If a sequence feature class includes another sequence feature class, then the individual of the former has an `sbol:SequenceAnnotation` resource describing the start and end locations of the annotation. The annotation resource's `sbol:subComponent` property identifies the individual of the latter class.
- The start, end, and strand properties of an `sbol:SequenceAnnotation` resource are inferred from the genome positions of the parent and child classes.

SyBiOnt contains 7754 `DnaComponent` parts that can be exchanged using SBOL. For each OWL class representing a sequence feature, an individual of type `DnaComponent` was created. The SBOL model enforces the rule that each `DnaComponent` resource must be a type of sequence feature from the SO. SO-based superclasses included in SyBiOnt are used to infer these types. The names, descriptions, and nucleotide sequences of the resources were extracted from the OWL classes and stored using the `rdfs:label`, `rdfs:comment`, and `sbol:nucleotides` properties, respectively.

Relationships of type `hasPart` were used to create SBOL sequence annotations. Differences between the genome positions of sequence features linked by a specific relationship were used to calculate the `sbol:bioStart` and `sbol:bioEnd` properties of the sequence annotations. The `sbol:subComponent` property was used to identify the sequence feature resources used for annotation.

Figure 10 shows an example of a promoter in SBOL format. The resource is of both types `sbol:DnaComponent` and `so:SO_0000167` (promoter sequence), classes that are both mandatory in the SBOL model. In addition, the resource is also of type `bo:2685` from SyBiOnt. The promoter has an annotation identified by the `sbol:annotation` property.

2. DISCUSSION

Currently, the synthetic biology design process is often limited by access to biological knowledge and access to the sequence of suitable parts. The data to provide this knowledge often exists, but it is fragmented in a variety of databases across the world, in different formats and of varying quality. In this work, we aimed to demonstrate the power of an integrative approach to design in synthetic biology, where data from remote resources can be sourced, integrated, and mined to aid in the design process.

In particular, we show the value of ontologies for integrating disparate data sources and providing a standardized data model for helping to define these resources and the information necessary to aid in the design of engineered biological systems. We have developed an ontology for application to data integration and mining in synthetic biology. To our knowledge, this is first report of an ontology designed specifically for synthetic biology. Using this ontology, we have demonstrated this integrated approach to produce an exemplar data warehouse populated with data about the model Gram-positive bacterium *B. subtilis* derived from many different data sources. We now aim to extend our approach to other model organisms with rich data resources such as *Escherichia coli* and *Saccharomyces cerevisiae*.

One of the advantages of ontologies over other data models is that they can be reasoned over. Reasoning over an ontology is a much more powerful and expressive method of data mining than querying over a standard relational data model.[22] Ontological reasoning can include implicitly derived facts and can answer conceptual as well as extensional queries. We show how SyBiOntKB captures domain knowledge about *B. subtilis* using description logics and can be queried using existing ontological reasoners. We also demonstrate how OWL queries, in the form of OWL classes, can be used to mine SyBiOnt.

Finding suitable parts for producing designs of engineered biological systems is a time-consuming process. Integrating data sources also brings together information about the sequence of genetic parts with data about their functional characteristics. The incorporation of this type of information allowed us to mine SyBiOntKB for genetic parts of a given type. We showed,

DOI: 10.1021/acssynbio.5b00295
ACS Synth. Biol. 2016, 5, 1086−1097

as an example, how operators that have binding sites for repressors or activators were identified. Furthermore, we also showed how promoter classes can be assigned different types of class membership based on information about the type of transcriptional regulation and sigma factors involved in transcriptional initiation. Examples for the functional classification and mining of parts, such as the identification of CDSs that encode activators, repressors, kinases, and response regulators, were also identified. Furthermore, we also developed more complex and rich queries to mine examples of devices potentially encoding logic gates and to address more high-level questions with respect to the analysis and design of biochemical pathways and regulatory systems. These examples are only a very limited subset of the possible types of parts that could be mined and design questions that can be addressed. We hope that development and use of this ontology serves as a model for how automated reasoning can be used to inform design in synthetic biology.

When developing SyBiOntKB, we have tried to reuse existing standard formats and ontologies wherever possible. SyBiOnt, therefore, builds on and incorporates other well-used standards such as GO and SO. For example, the classification of CDSs in SyBiOntKB is achieved by defining classes that refer to GO terms. In addition, the SBOL data model has been used to provide terms to model biological parts for computational access; therefore, these terms can be applied to standardize the querying of sequence features from SyBiOntKB. In addition to building genetic designs, the information from the ontology can also be used as a basis to create and annotate computational models of synthetic systems. In the future, we will seek to expand SyBiOnt still further, with the help of the community, to incorporate further ontologies such as the Systems Biology Ontology.[67] The availability of SyBiOnt also promises to provide a unifying semantics for the expansion of the SBOL standard, potentially proving a way to match the semantics of the entities in the core data model to entities in extended, attached data items such as dynamic models and experimental data.

The SyBiOntKB ontology captures information in a computational and programmatically accessible fashion in a standard format. As a result, information about biological parts and molecular interactions captured within it is available in a form suitable for the automated design of complex and large-scale biological systems. We envisage that data warehouses built using the SyBiOntKB ontology can provide a useful resource to enhance the process of biodesign automation. Since data warehouses that employ SyBiOntKB can be made available in RDF form as triple stores, the data is also available to integrate with the vision of the Semantic Web.[27] In summary, here, we demonstrated the use of data integration and automated mining of biological parts for synthetic biology. We used ontologies to represent extensive biological data formally for computational access. This approach has allowed us to write complex queries that could not be executed previously in an automated fashion in order to classify biological parts. The resources presented here will accelerate further data integration and mining of data and will facilitate scaling up the designs of biological systems using computational approaches, advancing the field of synthetic biology.

3. METHODS

3.1. RDF Graph Representation of Biological Data. Information about the biological entities, their relationships, and attributes from the previously developed integrated knowledge base for *B. subtilis*, BacillOndex,[42] was initially converted into RDF triples, which were then used to build the SyBiOnt ontology and the knowledge base using OWL axioms. The data set was read into Ondex[68] and exported as an RDF graph. The graph was saved as a single RDF file containing triples for entities, relations, and attributes, together with the Ondex metadata, including annotations regarding entity and relation types and the relation hierarchy. The BacillOndex RDF graph was then converted into OWL format in order to formally model the *B. subtilis* domain knowledge as an ontology.

3.2. Building the Ontological Representation in OWL. The resources that represent entity types and their associated entities from BacillOndex were modeled as OWL classes in the ontology. The relations and attributes of entities were modeled as subclass restrictions on these classes. This approach allowed the knowledge from BacillOndex to be made explicit for machine access and to have reasoning capabilities over the data.

Scripts, in the Clojure programming language, were developed to map the RDF model to OWL using the Tawny-OWL API.[69] Tawny-OWL allows the definition of ontology classes both programmatically and using a domain specific language (DSL); hence, it facilitates the rapidly development of large ontologies. The Clojure programming language was chosen since Tawny-OWL is also available in Clojure and existing Java libraries can still be used. The programmatic approach was used to map the RDF data to OWL, and the DSL provided by Tawny-OWL was used to manually define additional SyBiOnt classes. The DSL is designed to be human readable and similar to the widely used Manchester Syntax (http://www.w3.org/TR/owl2-manchester-syntax), with the advantage of easily validating OWL classes using a standard integrated development environment such as Eclipse (https://eclipse.org). The resulting ontology was exported in the form of RDF and was stored in the Sesame RDF triple store (http://www.openrdf.org).

Information representing biological entities was modeled as a class hierarchy. Associations between biological entities were modeled as OWL restrictions. To model biological constraints not represented in the RDF, *closure* and *disjoint* axioms and *cardinality* restrictions were added to the OWL representations.

3.3. Mining SyBiOntKB. OWL classes with necessary and sufficient conditions[25,35] that identified genetic entities relevant to synthetic biology design were defined. These conditions were used to provide logical definitions of classes[53] for the computational classification process. These conditions were implemented in the SyBiOntKB classes using restrictions acting as superclasses. When implemented via `equivalentClass` axioms[70] by defining additional classes, such restrictions become necessary and sufficient conditions.

Criteria described in defined classes were used by reasoners to categorize classes. After reasoning, new subclass relationships were inferred between classes with necessary conditions and these defined classes. As a result, these defined classes acted as queries for mining part descriptions from the OWL representation of the data. OWL reasoners, including FaCT++[71] and HermiT,[51] were run to execute these queries programmatically using the Tawny-OWL library or manually using Protégé. In these queries, subsets of SyBiOntKB, which were created programmatically, were used to improve the query performance.

DOI: 10.1021/acssynbio.5b00295
ACS Synth. Biol. 2016, 5, 1086−1097

```
Class: OperatorA
   DisjointWith:
      OperatorB
Class: OperatorB
Class: PromoterX
   SubClassOf:
      hasPart exactly 1 OperatorA,
      hasPart exactly 1 OperatorB,
      hasPart only (OperatorA or OperatorB),
```

Figure 11. Closure axioms and disjointness statements are added to enable reasoners to infer that `PromoterX` has two operators.

The classification of entities such as promoters in terms of their compositional features requires that the set of features is explicitly specified and that these compositional features can be distinguished from each other.[35] Therefore, in order to classify a promoter with only one TF binding site, in addition to the necessary condition to have an operator, the sufficient condition that this operator is the only binding site must be included. Such a sufficient condition can be provided by closure axioms, which are used to indicate that no other information except what is provided would be available.[49] These closure axioms were added to the ontology using OWL's universal *allValuesFrom* (*only*) restrictions.[36,72]

The number of operators for a promoter was made explicit in the ontology using cardinality restrictions to facilitate reasoning about the number of a promoter's inputs. In OWL, these restrictions are used to describe the minimum, maximum, or exact number of relationships for a class. Figure 11 shows a promoter, `PromoterX`, and its cardinality restrictions. The promoter has precisely one `OperatorA` and one `OperatorB`, which are explicitly defined as two disjoint operators. In addition, the universal `hasPart only (OperatorA or OperatorB)` closure axiom is added to specify that the promoter can have only `OperatorA` or `OperatorB`. Reasoners can therefore infer that `PromoterX` has exactly two distinct operators.

In order for reasoners to distinguish a promoter and its operators, we wanted to normalize the ontology and make all the sibling classes disjoint. However, adding these disjoint axioms for all classes reduced the speed of reasoners. Instead, disjointness was defined between the promoter and operator superclasses, making all of the operators subclasses disjoint from all of the promoter subclasses. Disjointness axioms were then added to operators that are part of the same promoters.

3.4. Constructing SBOL Parts. SBOL mapping of classes representing DNA-based parts was carried out using the Jena API (http://jena.apache.org) and SPARQL[73] queries. Rules that provide the mapping between the ontology presented here and SBOL objects were implemented as CONSTRUCT queries, allowing the returning of query results in the form of RDF graphs that can directly be used to update the underlying graph data. RDF rule-based mapping was used to map all classes representing sequence features into the corresponding SBOL, version 1.0, RDF representation. These SBOL RDF data were imported back into the RDF triple store.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acssynbio.5b00295. The SyBiOnt ontology the SyBiOntKB knowledge base and Clojure scripts that utilize the Tawny-OWL API to create these resources are available from a repository at http://w3id.org/synbio/ont. The repository also contains information for accessing an RDF end point for the knowledge base. The Tawny-OWL API is available at https://github.com/phillord/tawny-owl.

Various class definitions referenced in this article (PDF)

■ AUTHOR INFORMATION

Corresponding Author

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

Present Address

∥(J.H.) Biological Sciences, Macquarie University, Sydney, NSW 2109, Australia.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

G.M. and A.W. were supported by Engineering and Physical Sciences Research Council (EPSRC) grant EP/J02175X/1. K.F. was funded by EPSRC grant EP/K031953/1. J.M. gratefully acknowledges funding from FUJIFILM Diosynth Biotechnologies. H.S. was funded through the generous support of the National science Foundation, Biological Infrastructure award no. 1355909 and Molecular and Cellular Bioscience award no. 1158573.

■ REFERENCES

(1) de Lorenzo, V., and Danchin, A. (2008) Synthetic biology: discovering new worlds and new words. *EMBO Rep. 9*, 822−827.

(2) Agapakis, C. M., and Silver, P. A. (2009) Synthetic biology: exploring and exploiting genetic modularity through the design of novel biological networks. *Mol. BioSyst. 5*, 704−713.

(3) Hallinan, J. S., Park, S., and Wipat, A. (2012) Bridging the gap between design and reality − a dual evolutionary strategy for the design of synthetic genetic circuits, *Proceedings of the International Conference on Bioinformatics Models, Methods and Algorithms*, pp 263−268.

(4) Endy, D. (2005) Foundations for engineering biology. *Nature 438*, 449−453.

(5) Koide, T., Lee Pang, W., and Baliga, N. S. (2009) The role of predictive modelling in rationally re-engineering biological systems. *Nat. Rev. Microbiol. 7*, 297−305.

(6) Guido, N. J., Wang, X., Adalsteinsson, D., McMillen, D., Hasty, J., Cantor, C. R., Elston, T. C., and Collins, J. J. (2006) A bottom-up approach to gene regulation. *Nature 439*, 856−860.

(7) Smolke, C., and Silver, P. (2011) Informing Biological Design by Integration of Systems and Synthetic Biology. *Cell 144*, 855−859.

(8) Szallasi, Z., Stelling, J. A., and Periwal, V. (2006) *System Modeling in Cell Biology: From Concepts to Nuts and Bolts*, MIT Press.

(9) Stelling, J. (2004) Mathematical models in microbial systems biology. *Curr. Opin. Microbiol. 7*, 513−518.

(10) Endler, L., Rodriguez, N., Juty, N., Chelliah, V., Laibe, C., Li, C., and Le Novere, N. (2009) Designing and encoding models for synthetic biology. *J. R. Soc., Interface 6*, S405−S417.

DOI: 10.1021/acssynbio.5b00295
ACS Synth. Biol. 2016, 5, 1086−1097

(11) Ro, D.-K., Paradise, E. M., Ouellet, M., Fisher, K. J., Newman, K. L., Ndungu, J. M., Ho, K. A., Eachus, R. A., Ham, T. S., Kirby, J., Chang, M. C. Y., Withers, S. T., Shiba, Y., Sarpong, R., and Keasling, J. D. (2006) Production of the antimalarial drug precursor artemisinic acid in engineered yeast. *Nature 440*, 940−943.
(12) Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T., and Yamanishi, Y. (2007) KEGG for linking genomes to life and the environment. *Nucleic Acids Res. 36*, D480−D484.
(13) Vallenet, D., Labarre, L., Rouy, Z., Barbe, V., Bocs, S., Cruveiller, S., Lajus, A., Pascal, G., Scarpelli, C., and Medigue, C. (2006) MaGe: a microbial genome annotation system supported by synteny results. *Nucleic Acids Res. 34*, 53−65.
(14) von Mering, C., Jensen, L. J., Kuhn, M., Chaffron, S., Doerks, T., Kruger, B., Snel, B., and Bork, P. (2007) STRING 7-recent developments in the integration and prediction of protein interactions. *Nucleic Acids Res. 35*, D358−362.
(15) Lanza, A. M., Crook, N. C., and Alper, H. S. (2012) Innovation at the intersection of synthetic and systems biology. *Curr. Opin. Biotechnol. 23*, 712−717.
(16) Medema, M. H., van Raaphorst, R., Takano, E., and Breitling, R. (2012) Computational tools for the synthetic design of biochemical pathways. *Nat. Rev. Microbiol. 10*, 191−202.
(17) De Las Heras, A., Carreño, C. A., Martínez-García, E., and De Lorenzo, V. (2010) Engineering input/output nodes in prokaryotic regulatory circuits. *FEMS Microbiol. Rev. 34*, 842−865.
(18) Goble, C., and Stevens, R. (2008) State of the nation in data integration for bioinformatics. *J. Biomed. Inf. 41*, 687−693.
(19) Balakrishnan, R., Park, J., Karra, K., Hitz, B. C., Binkley, G., Hong, E. L., Sullivan, J., Micklem, G., and Michael Cherry, J. (2012) YeastMine-an integrated data warehouse for Saccharomyces cerevisiae data as a multipurpose tool-kit. *Database 2012*, bar062.
(20) Contrino, S., et al. (2012) modMine: flexible access to modENCODE data. *Nucleic Acids Res. 40*, D1082−D1088.
(21) Belleau, F., Nolin, M.-A., Tourigny, N., Rigault, P., and Morissette, J. (2008) Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. *J. Biomed. Inf. 41*, 706−716.
(22) Cheung, K.-H., Smith, A. K., Yip, K. Y. L., Baker, C. J. O., and Gerstein, M. B. (2007) Semantic Web approach to database integration in the life sciences, in *Semantic Web* (Baker, C. J. O., and Cheung, K.-H., Eds.) pp 11−30, Springer.
(23) Lenzerini, M. (2002) Data integration. *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 233−246.
(24) Stein, L. D. (2003) Integrating biological databases. *Nat. Rev. Genet. 4*, 337−345.
(25) Antezana, E., Egana, M., Blonde, W., Illarramendi, A., Bilbao, I., De Baets, B., Stevens, R., Mironov, V., and Kuiper, M. (2009) The Cell Cycle Ontology: an application ontology for the representation and integrated analysis of the cell cycle process. *Genome Biol. 10*, R58.
(26) UniProt Consortium (2011) UniProt Knowledgebase: a hub of integrated protein data. *Database 2011*, bar009.
(27) Shadbolt, N., Hall, W., and Berners-Lee, T. (2006) The Semantic Web Revisited. *IEEE Intell Syst 21*, 96−101.
(28) Cheung, K.-H., Samwald, M., Auerbach, R. K., and Gerstein, M. B. (2010) Structured digital tables on the Semantic Web: toward a structured digital literature. *Mol. Syst. Biol. 6*, 403.
(29) Gruber, T. R. (1995) Toward principles for the design of ontologies used for knowledge sharing? *Int. J. Hum Comput. Stud 43*, 907−928.
(30) Bard, J. B. L., and Rhee, S. Y. (2004) Ontologies in biology: design, applications and future challenges. *Nat. Rev. Genet. 5*, 213−222.
(31) Consortium, T. G. O. (2001) Creating the Gene Ontology Resource: Design and Implementation. *Genome Res. 11*, 1425−1433.
(32) Eilbeck, K., Lewis, S., Mungall, C., Yandell, M., Stein, L., Durbin, R., and Ashburner, M. (2005) The Sequence Ontology: a tool for the unification of genome annotations. *Genome Biol. 6*, R44.
(33) Natale, D. A., Arighi, C. N., Barker, W. C., Blake, J. A., Bult, C. J., Caudy, M., Drabkin, H. J., D'Eustachio, P., Evsikov, A. V., Huang, H., Nchoutmboube, J., Roberts, N. V., Smith, B., Zhang, J., and Wu, C. H. (2011) The Protein Ontology: a structured representation of protein forms and complexes. *Nucleic Acids Res. 39*, D539.
(34) Blondé, W., Mironov, V., Venkatesan, A., Antezana, E., De Baets, B., and Kuiper, M. (2011) Reasoning with bio-ontologies: using relational closure rules to enable practical querying. *Bioinformatics 27*, 1562.
(35) Stevens, R., Egaña Aranguren, M., Wolstencroft, K., Sattler, U., Drummond, N., Horridge, M., and Rector, A. (2007) Using OWL to model biological knowledge. *Int. J. Hum Comput. Stud 65*, 583−594.
(36) Lin, Y., Xiang, Z., and He, Y. (2011) Towards a Semantic Web Application: Ontology-Driven Ortholog Clustering Analysis, *International Conference on Biomedical Ontology*, Buffalo, NY, July 26−30.
(37) Galdzicki, M., et al. (2012) *Synthetic Biology Open Language (SBOL)*, version 1.1.0, BioBricks Foundation Request for Comments.
(38) Bartley, B., Beal, J., Clancy, K., Misirli, G., Roehner, N., Oberortner, E., Pocock, M., Bissell, M., Madsen, C., Nguyen, T., Zhang, Z., Gennari, J. H., Myers, C., Wipat, A., and Sauro, H. (2015) Synthetic Biology Open Language (SBOL) Version 2.0.0. *J. Integr. Bioinform. 12*, 272.
(39) Peccoud, J., Blauvelt, M. F., Cai, Y., Cooper, K. L., Crasta, O., DeLalla, E. C., Evans, C., Folkerts, O., Lyons, B. M., Mane, S. P., Shelton, R., Sweede, M. A., and Waldon, S. A. (2008) Targeted Development of Registries of Biological Parts. *PLoS One 3*, e2671.
(40) Galdzicki, M., Rodriguez, C., Chandran, D., Sauro, H. M., and Gennari, J. H. (2011) Standard Biological Parts Knowledgebase. *PLoS One 6*, e17005.
(41) Galdzicki, M., et al. (2014) The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nat. Biotechnol. 32*, 545−550.
(42) Misirli, G., Wipat, A., Mullen, J., James, K., Pocock, M., Smith, W., Allenby, N., and Hallinan, J. (2013) BacillOndex: An Integrated Data Resource for Systems and Synthetic Biology. *J. Integr. Bioinform. 10*, 224.
(43) Tatusov, R. L., Natale, D. A., Garkavtsev, I. V., Tatusova, T. A., Shankavaram, U. T., Rao, B. S., Kiryutin, B., Galperin, M. Y., Fedorova, N. D., and Koonin, E. V. (2001) The COG database: new developments in phylogenetic classification of proteins from complete genomes. *Nucleic Acids Res. 29*, 22−28.
(44) Barbe, V., Cruveiller, S., Kunst, F., Lenoble, P., Meurice, G., Sekowska, A., Vallenet, D., Wang, T., Moszer, I., Medigue, C., and Danchin, A. (2009) From a consortium sequence to a unified sequence: the Bacillus subtilis 168 reference genome a decade later. *Microbiology 155*, 1758−1775.
(45) Sierro, N., Makita, Y., de Hoon, M., and Nakai, K. (2007) DBTBS: a database of transcriptional regulation in Bacillus subtilis containing upstream intergenic conservation information. *Nucleic Acids Res. 36*, D93−D96.
(46) Goto, S., Kawashima, S., Okuji, Y., Kamiya, T., Miyazaki, S., Numata, Y., and Kanehisa, M. (2000) KEGG/EXPRESSION: A Database for Browsing and Analysing Microarray Expression Data. *Genome Informatics 11*, 222−223.
(47) Camon, E., Magrane, M., Barrell, D., Binns, D., Fleischmann, W., Kersey, P., Mulder, N., Oinn, T., Maslen, J., Cox, A., and Apweiler, R. (2003) The Gene Ontology Annotation (GOA) Project: Implementation of GO in SWISS-PROT, TrEMBL, and InterPro. *Genome Res. 13*, 662−672.
(48) Natale, D., Arighi, C., Barker, W., Blake, J., Chang, T.-C., Hu, Z., Liu, H., Smith, B., and Wu, C. (2007) Framework for a Protein Ontology. *BMC Bioinf. 8*, S1.
(49) Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C., Motta, E., Shadbolt, N., Stutt, A., and Gibbins, N. (2004) Engineering Knowledge in the Age of the Semantic Web, *Lecture Notes in Computer Science*, Vol. *3257*, pp 63−81, Springer, Berlin.
(50) Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. (2007) Pellet: A practical OWL-DL reasoner. *Web Semant 5*, 51−53.
(51) Motik, B., Shearer, R., and Horrocks, I. (2009) Hypertableau reasoning for description logics. *J. Artif Intell Res. 36*, 165−228.

DOI: 10.1021/acssynbio.5b00295
ACS Synth. Biol. 2016, 5, 1086−1097

(52) Noy, N., and McGuinness, D. L. (2001) *Ontology Development 101: A Guide to Creating Your First Ontology*, Stanford University, Stanford, CA.
(53) Stevens, R., and Hull, D. (2010) Defining Definitions, *Ontogenesis*, http://ontogenesis.knowledgeblog.org/824.
(54) Buchler, N. E., Gerland, U., and Hwa, T. (2003) On schemes of combinatorial transcription logic. *Proc. Natl. Acad. Sci. U. S. A. 100*, 5136−5141.
(55) Bolouri, H., and Davidson, E. H. (2002) Modeling transcriptional regulatory networks. *BioEssays 24*, 1118−1129.
(56) van Hijum, S. A. F. T., Medema, M. H., and Kuipers, O. P. (2009) Mechanisms and Evolution of Control Logic in Prokaryotic Transcriptional Regulation. *Microbiol. Rev. 73*, 481−509.
(57) Silva-Rocha, R., and de Lorenzo, V. A. (2008) Mining logic gates in prokaryotic transcriptional regulation networks. *FEBS Lett. 582*, 1237−1244.
(58) Barnard, A., Wolfe, A., and Busby, S. (2004) Regulation at complex bacterial promoters: how bacteria use different promoter organizations to produce different regulatory outcomes. *Curr. Opin. Microbiol. 7*, 102−108.
(59) Cox, R. S., Surette, M. G., and Elowitz, M. B. (2007) Programming gene expression with combinatorial promoters. *Mol. Syst. Biol. 3*, 145.
(60) Harvie, D. R., Andreini, C., Cavallaro, G., Meng, W., Connolly, B. A., Yoshida, K.-i., Fujita, Y., Harwood, C. R., Radford, D. S., Tottey, S., Cavet, J. S., and Robinson, N. J. (2006) Predicting metals sensed by ArsR-SmtB repressors: allosteric interference by a non-effector metal. *Mol. Microbiol. 59*, 1341−1356.
(61) Elowitz, M. B., and Leibler, S. (2000) A synthetic oscillatory network of transcriptional regulators. *Nature 403*, 335−338.
(62) Ninfa, A. J. (2010) Use of two-component signal transduction systems in the construction of synthetic genetic networks. *Curr. Opin. Microbiol. 13*, 240−245.
(63) Szurmant, H., and Hoch, J. A. (2010) Interaction fidelity in two-component signaling. *Curr. Opin. Microbiol. 13*, 190−197.
(64) Levskaya, A., Chevalier, A. A., Tabor, J. J., Simpson, Z. B., Lavery, L. A., Levy, M., Davidson, E. A., Scouras, A., Ellington, A. D., Marcotte, E. M., and Voigt, C. A. (2005) Synthetic biology: Engineering Escherichia coli to see light. *Nature 438*, 441−442.
(65) Skerker, J. M., Perchuk, B. S., Siryaporn, A., Lubin, E. A., Ashenberg, O., Goulian, M., and Laub, M. T. (2008) Rewiring the Specificity of Two-Component Signal Transduction Systems. *Cell 133*, 1043−1054.
(66) Clarke, E. J., and Voigt, C. A. (2011) Characterization of combinatorial patterns generated by multiple two-component sensors in E. coli that respond to many stimuli. *Biotechnol. Bioeng. 108*, 666−675.
(67) Courtot, M. (2011) Controlled vocabularies and semantics in systems biology. *Mol. Syst. Biol. 7*, 543.
(68) Köhler, J., Baumbach, J., Taubert, J., Specht, M., Skusa, A., Rüegg, A., Rawlings, C., Verrier, P., and Philippi, S. (2006) Graph-based analysis and visualization of experimental results with ONDEX. *Bioinformatics 22*, 1383−1390.
(69) Lord, P. (2013) The Semantic Web takes Wing: Programming Ontologies with Tawny-OWL, *Experiences and Directions Workshop (OWLED)*, Montpellier, France, May 26−27.
(70) Horrocks, I. (2008) Ontologies and the Semantic Web. *Commun. ACM 51*, 58−67.
(71) Tsarkov, D., and Horrocks, I. (2006) Automated Reasoning, in *Lecture Notes in Computer Science* (Furbach, U., and Shankar, N., Eds.) Vol. *4130*, pp 292−297, Springer, Berlin.
(72) Beisswanger, E., Lee, V., Kim, J.-J., Rebholz-Schuhmann, D., Splendiani, A., Dameron, O., Schulz, S., and Hahn, U. (2008) Gene Regulation Ontology (GRO): design principles and use cases. *Stud Health Technol. Inform 136*, 9−14.
(73) Pasquier, C. (2008) Biological data integration using Semantic Web technologies. *Biochimie 90*, 584−594.

DOI: 10.1021/acssynbio.5b00295
ACS Synth. Biol. 2016, 5, 1086−1097