Published online 20 April 2016 *Nucleic Acids Research, 2016, Vol. 44, Web Server issue* ***W587–W592***
doi: 10.1093/nar/gkw289

GIANT API: an application programming interface for functional genomics

Andrew M. Roberts[1,†], Aaron K. Wong[1,†], Ian Fisk[1] and Olga G. Troyanskaya[1,2,3,*]

[1]Simons Center for Data Analysis, Simons Foundation, New York, NY 10010, USA, [2]Department of Computer Science, Princeton University, Princeton, NJ 08540, USA and [3]Lewis-Sigler Institute for Integrative Genomics, Princeton University, Princeton, NJ 08540, USA

Received February 20, 2016; Revised April 1, 2016; Accepted April 8, 2016

ABSTRACT

GIANT API provides biomedical researchers programmatic access to tissue-specific and global networks in humans and model organisms, and associated tools, which includes functional re-prioritization of existing genome-wide association study (GWAS) data. Using tissue-specific interaction networks, researchers are able to predict relationships between genes specific to a tissue or cell lineage, identify the changing roles of genes across tissues and uncover disease-gene associations. Additionally, GIANT API enables computational tools like NetWAS, which leverages tissue-specific networks for re-prioritization of GWAS results. The web services covered by the API include 144 tissue-specific functional gene networks in human, global functional networks for human and six common model organisms and the NetWAS method. GIANT API conforms to the REST architecture, which makes it stateless, cacheable and highly scalable. It can be used by a diverse range of clients including web browsers, command terminals, programming languages and standalone apps for data analysis and visualization. The API is freely available for use at http://giant-api.princeton.edu.

INTRODUCTION

Tissue-specificity is a critical aspect of most complex human diseases. However, even with innovations in high-throughput sequencing, which provide biologists with vast collections of heterogeneous genomic data, direct assay of protein function and interactions remains infeasible in many human tissues. Computational methods can infer tissue-specific functional interactions between genes by integrating large data compendia, measuring each dataset for its relevance to a tissue context, even in tissues and cell lineages for which no or few tissue-specific data exist (1–4). The resulting tissue-specific functional networks provide a detailed, genome-scale portrait of protein function and interactions in specific tissues and cell lineages. They can predict lineage-specific molecular response, reveal the changing functional role of genes depending on tissue context and facilitate powerful methods to re-prioritize existing genome-wide association study (GWAS) results to improve disease-gene association (2).

The GIANT web server (2) provides an interactive interface to a large collection of human tissue-specific functional maps and related network analyses. However, functional networks can still be challenging to integrate into researchers' workflows when systematic analysis of large groups of genes and/or interactions is needed. Accessing these networks and accompanying computational methods requires users to download large data files, interact with online forms, or run tools locally. GIANT API has been designed to remove these hurdles by providing a convenient, programmatic way to explore and work with the tissue-specific functional networks and computational tools available at GIANT. This is the first API to allow automatable access to 144 tissue-specific networks in human (2), global functional networks for human and six common model organisms (5), and the NetWAS method for integrated, functional re-prioritization of GWAS results (2).

An API for functional genomic web services offers researchers many ancillary benefits beyond ease of access. By adopting well-defined standards for data representation and access, GIANT API promotes information sharing and rapid dissemination of novel computational techniques in network biology. Programmatic execution enables researchers to rapidly test multiple hypotheses with reduced risk of manual errors and increased repeatability of results. Finally, an API allows easy extensibility of its features to new programming languages. We show, for example, how GIANT API can be invoked from Python by importing a tiny wrapper library. The API can be similarly adapted to other programming languages with small marginal effort.

Several popular functional genomic resources have implemented APIs for computational biologists, underscoring the need for programmatic access. Resources such as Bi-

*To whom correspondence should be addressed. Tel: +1 609 258 1749; Fax: +1 609 258 1771; Email: ogt@genomics.princeton.edu
†These authors contributed equally to the paper as first authors.

oGRID (6), KEGG (7) and ArrayExpress (8) enable high-throughput access to protein interaction data, curated pathways and expression data, respectively through their API services. Other resources also integrate diverse data to predict genome-wide functional networks (9), with accompanying API services. GIANT API is similar to these APIs in that they are all *RESTful* APIs, i.e. conforming to the generally accepted constraints of REpresentational State Transfer architecture. However, GIANT API also fills a critical gap that no existing resource provides programmatic access to tissue-specific interaction networks, nor does any other resource enable biologists to reanalyze their GWAS results in the context of those networks. With GIANT API, biologists can perform complex programmatic queries against a large collection of tissue-specific networks, retrieve the surrounding functionally related genes, explore the underlying evidence for the predictions and integrate the result in visualization tools.

EXAMPLE USE CASES

This section contains three detailed cases exemplifying typical usage scenarios for GIANT API. Our examples span tissue-specific functional networks, NetWAS analysis and API integration in visualization tools.

Exploring human tissue-specific functional networks

One of the main web services provided by GIANT API is the prediction of tissue-specific functional gene networks for human. A *functional network* is an undirected weighted graph where vertices represent genes and edges represent predicted functional relations between gene pairs (2). A functional relation implies that both genes participate in the same biological process in a specific tissue or functional context. Functional relationships are predicted using a naive Bayesian classifier, which is first trained using already known gene interactions (i.e. a gold standard). The trained classifier can then be used to make predictions of relations between previously unstudied gene pairs. Each edge weight is the posterior probability of a functional relation between two genes. In a *tissue-specific* network, a separate Bayesian classifier is trained for each tissue. This distinction is important because gene interactions differ significantly based on tissue type. Functional networks served by the GIANT API have been evaluated computationally and experimentally both in human (2) and in model organisms (1).

Consider a researcher who wants to study how the Parkinson's disease-associated SNCA (10) and PARK7 (11) genes interact with other genes in the context of the human brain. Figure 1 shows how to predict these interactions from a terminal shell using the **networks** function of the API. Steps 1 and 2 optionally verify that the tissue and gene names are valid and can be used by the API to make functional predictions. Step 3 retrieves the desired functional network from the GIANT server, which we filter through the Python JSON module for easier visual inspection. The weight field in the output represents the predicted probability of having a functional relationship between the corresponding source and target genes. By filtering the JSON response, the researcher is able to identify those gene pairs with highly probable interactions, e.g. ATP5J and PARK7.

```
$ curl giant-api.princeton.edu/tissues/check/brain                    ❶
["brain"]

$ curl giant-api.princeton.edu/genes/check/snca+park7                 ❷
["SNCA","PARK7"]

$ curl giant-api.princeton.edu/networks \                             ❸
       -d tissue=brain                  \
       -d genes=SNCA+PARK7 | python -m json.tool
[
    ...
    {"source": "NDUFA1", "target": "PARK7", "weight": 0.7948},
    {"source": "ATP5J",  "target": "PARK7", "weight": 0.8271},
    {"source": "SNCA",   "target": "AP2S1", "weight": 0.1368},
    ...
]

$ curl giant-api.princeton.edu/networks    \                          ❹
       -d tissue=brain -d genes=SNCA+PARK7 \
       -d num_genes=20 -d prior=0.15
[...]

$ curl giant-api.princeton.edu/networks/evidence \                    ❺
       -d source=ATP5J -d target=PARK7          \
       -d tissue=brain | python -m json.tool
{
    "datatypes": [{"name": "Co-expression",
                   "weight": 0.9806},
                  {"name": "GSEA chemical and genetic perturbations",
                   "weight": 0.01934}],
    "datasets": [
        {
            "dataset": "GSEA C2 CGP",
            "title": "Chemical and genetic perturbations",
            "version": "3.0",
            "posterior": 0.0141
        },
        {
            "dataset": "GDS1295",
            "title": "Lethal congenital contracture syndrome",
            "posterior": 0.0070
        },
        ...
}
```

Figure 1. Exploring functional networks using GIANT API. Labeled commands illustrate how to call the API from a terminal shell to: (1) verify that a given tissue is valid and the API can make predictions for it, (2) validate gene names, (3) retrieve a functional network, (4) specify custom prediction parameters, (5) query for datasets that support a particular prediction.

The **networks** function also features optional API fields, which allow over-riding default values used for computations and output formatting. This is illustrated in Step 4, where the API user specifies custom prediction parameters, e.g. number of genes to be returned in the result network graph.

GIANT API predicts functional relationships by incorporating evidence from thousands of diverse datasets. Therefore, it is useful to know which datasets in the GIANT compendium most strongly support any particular prediction. A researcher can retrieve a list of these datasets, i.e. evidence, by using the **networks/evidence** endpoint, which is shown in Step 5. Here, the API user requests the top 10 datasets that drive the strong ATP5J-PARK7 prediction from Step 3. The response contains names and weightings for each dataset, as well as version or date information if available.

NetWAS for refining GWAS results

A second web service accessible through GIANT API is NetWAS, which is a method of re-ranking the results of an existing GWAS by incorporating knowledge of tissue-specific gene interactions captured in functional networks (2). NetWAS involves training a support vector machine

```
>>> from giantapi import NetwasJob                                    1

>>> nj = NetwasJob(gwas_file="bmi-2012.out",                          2
                   gwas_format="vegas",
                   tissue="adipose_tissue",
                   p_value=0.01)

>>> nj.start()                                                        3

>>> print nj                                                          4
{
    "id": "a99e5848-ee59-4e55-a8d2-95186441634c",
    "created": "2015-12-26T19:52:09Z",
    "tissue": "adipose_tissue",
    "p_value": "0.0100",
    "status": "running",
    ...
}

>>> print nj.log()                                                    5
Reading genes
NEW Class array
Cross Validation Trial 0
SLACK NORM = 1
ALG = 3
Learned
Classified 3099 examples
...

>>> df = nj.results()                                                 6
>>> print df.head()
gene       class     z_score
PNRC1      -1        0.512196
DUSP6      -1        0.495562
MYADM      -1        0.46268
KRT6B      -1        0.439248
RANBP6     -1        0.407589
...
```

Figure 2. NetWAS analysis using GIANT API from Python. Labeled steps show how to: (1) import Python adapter for the API, (2) define a new NetWAS job, (3) start SVM training, (4) print job status, (5) print training log, (6) print results.

(SVM) to discover new genes that are associated with the phenotype of interest in the underlying GWAS. It has been systematically evaluated and demonstrated to improve GWAS ranking with respect to prioritizing known disease genes and drug targets (2). A NetWAS task is specified by (i) a GWAS results file, (ii) a functional network (tissue specific or global) and (iii) a *P*-value, which is used for significance testing on the GWAS data.

NetWAS was previously available only through a web form but it is now possible to use GIANT API to create NetWAS tasks programmatically. For example, Figure 2 shows the NetWAS workflow in Python to re-prioritize a GWAS of BMI (12) based on connectivity of genes in the *adipose tissue*. After importing the API functions (Step 1), the researcher initializes the NetWAS job with the desired tissue context and *P*-value threshold (Step 2). The input GWAS file is contained in the **bmi-2012.txt** file. Step 3 starts SVM training on the GIANT server, where the output vectors are derived from the GWAS results and feature vectors are based on the functional network.

Task status and progress can be monitored as shown in Steps 4 and 5 respectively. Once SVM training is complete, the user can retrieve NetWAS results in a Pandas data frame (Step 6), which can be further filtered and analyzed using native Python functions (Step 7).

GIANT API also has an HTTP interface that allows its resources to be accessed from a web browser, which provides an alternative method for monitoring NetWAS tasks. For example, a researcher can initiate multiple NetWAS jobs programmatically using a Python script and then monitor their progress and retrieve results from the web interface.

Biovisualization

Another benefit of GIANT API is that it enables scientists to easily add functional networks and relevant computational methods to their existing research infrastructures. The API uses ubiquitous formats for data representation and access (e.g. JSON, HTTP GET/POST) so it can easily interface with popular visualization tools such as D3 (13).

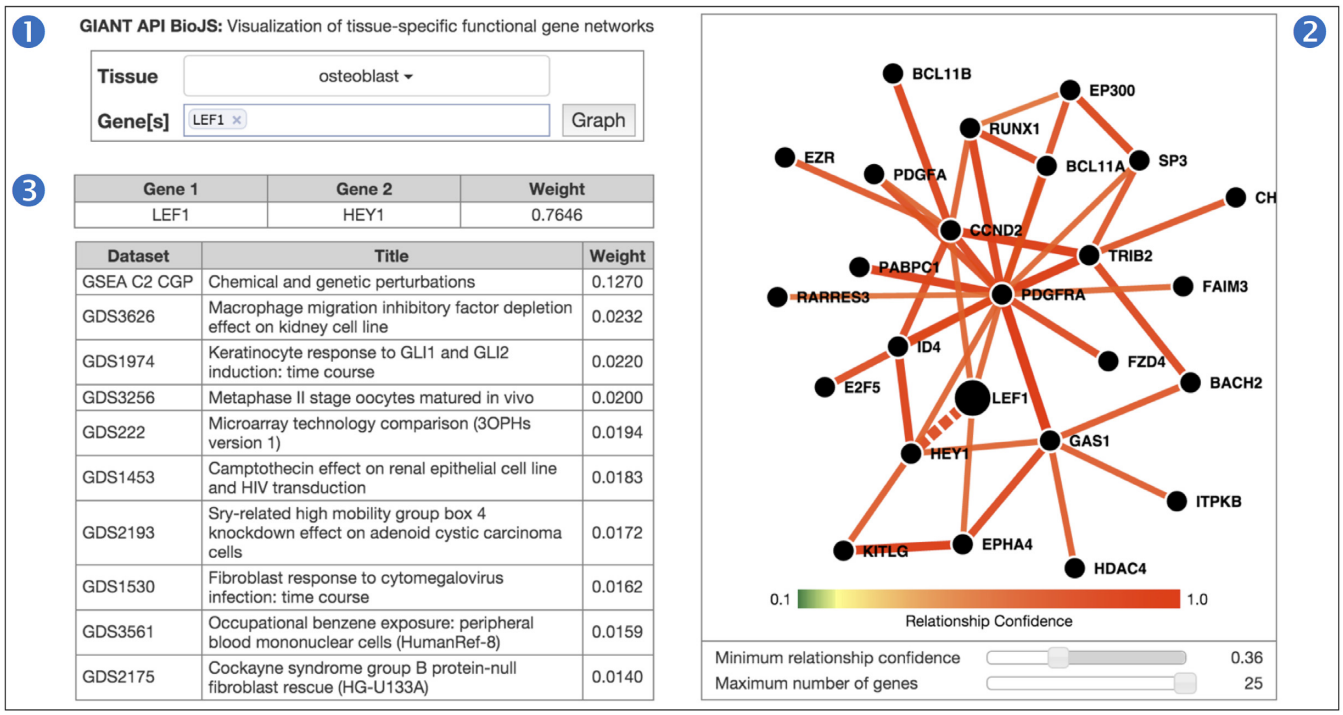
To demonstrate how developers can use GIANT as a building block, we developed a functional network visualization module for BioJS, which is an open-source library of modular JavaScript tools for bioinformatics (Figure 3) (14). This module relies on the GIANT API to execute all of its underlying data queries, and uses D3 for final rendering of network graphs. Users are able to explore functional networks for a set of query genes in any one of 144 human tissues. The graphs are interactive: they can be filtered for network size and confidence thresholds, and also allow the users to query dataset evidence by clicking on any of the edges. For example, Figure 3 shows the BioJS user exploring the functional network for the osteoblast tissue in the neighborhood of the LEF1 gene (Step 1). The LEF1–HEY1 interaction is selected by the user (Step 2), which triggers a GIANT API call to get the top datasets supporting this particular gene interaction (Step 3).

SYSTEM ARCHITECTURE AND DESIGN

Block diagram for the GIANT API is given in Figure 4. API calls can originate from diverse sources such as web browsers, terminals shells, programming languages (e.g. Python) or data visualization tools. Apache HTTP server acts as the API gateway. It is also responsible for serving static resources such as JavaScript code. API services for data and computations are implemented in Python using the Django REST Framework. Communication between the HTTP server and Python is enabled via the Web Server Gateway Interface.

MySQL database is used for persistent storage of any data needed for API services. To improve responsiveness, API uses Memcached (http://memcached.org), which saves results of recent API requests in memory. If the same API call is encountered before cache timeout, the response can be served from memory, thus avoiding any database queries or additional computations.

API services that require significant computation time (e.g. NetWAS) are processed asynchronously. Requests for these services are first entered in a FIFO queue implemented using the Celery task scheduler (http://www.



Gene 1	Gene 2	Weight
LEF1	HEY1	0.7646

Dataset	Title	Weight
GSEA C2 CGP	Chemical and genetic perturbations	0.1270
GDS3626	Macrophage migration inhibitory factor depletion effect on kidney cell line	0.0232
GDS1974	Keratinocyte response to GLI1 and GLI2 induction: time course	0.0220
GDS3256	Metaphase II stage oocytes matured in vivo	0.0200
GDS222	Microarray technology comparison (3OPHs version 1)	0.0194
GDS1453	Camptothecin effect on renal epithelial cell line and HIV transduction	0.0183
GDS2193	Sry-related high mobility group box 4 knockdown effect on adenoid cystic carcinoma cells	0.0172
GDS1530	Fibroblast response to cytomegalovirus infection: time course	0.0162
GDS3561	Occupational benzene exposure: peripheral blood mononuclear cells (HumanRef-8)	0.0159
GDS2175	Cockayne syndrome group B protein-null fibroblast rescue (HG-U133A)	0.0140

Figure 3. Functional gene network visualization using BioJS + GIANT API. Labeled panels show: (1) tissue and gene selection, (2) the corresponding network graph, (3) datasets supporting the selected gene interaction.

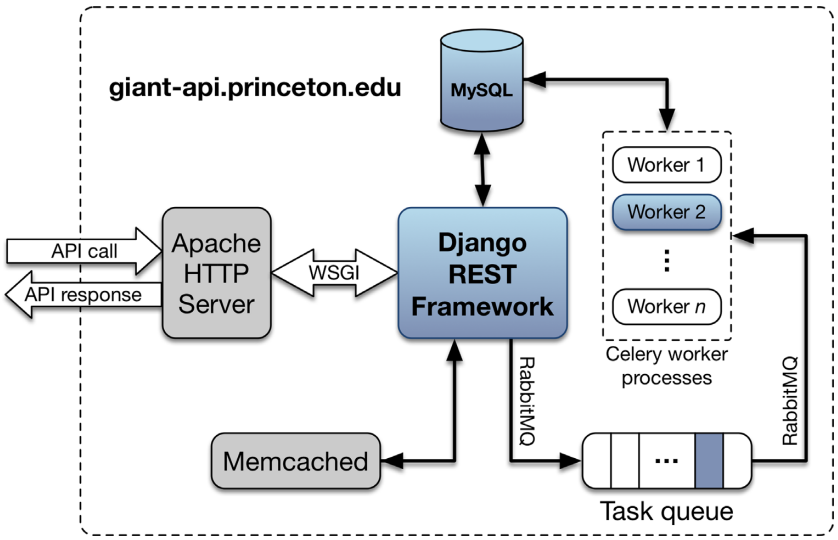


Figure 4. GIANT API architecture

celeryproject.org). Queued tasks are then executed by the next available worker process. Communications between Django and Celery use the RabbitMQ message broker (https://www.rabbitmq.com). Upon completion, task results are saved in MySQL for later retrieval. Users can check the progress of their requests by polling the API with a unique job ID, or alternatively be notified via email when their job has completed.

API ENDPOINTS

GIANT is a RESTful API in that it conforms to the constraints of the Representational State Transfer architecture. In particular, GIANT API (i) is stateless and cacheable, (ii) supports standard HTTP methods such as GET and POST, (iii) returns its responses in JSON format, and (iv) allows hyperlinks to its resources.

Table 1 summarizes the endpoints of the GIANT API. The first set of endpoints, prefixed with **networks**, is used for querying 144 tissue-specific functional networks for hu-

Table 1. Summary of API endpoints

Access method	Endpoint^	Fields	Description
GET	<API_ROOT>/**networks**/<tissue>/<gene>[+<gene2>[+…]]	[?num_genes=<int>] [&prior=<real>] [&enum=<bool>]	**tissue:** One of 144 human tissues supported **gene[s]:** One or more query genes. API will predict functional relationships between these genes and all other genes. **num_genes:** Optional field specifying the number of genes returned in the network graph. Default is 50. **prior:** Optional prior probability used for computing functional relationship posteriors. Default is 0.1. **enum:** Optional field indicating whether gene names should be enumerated in the API response (useful for graph libraries). Default is false.
POST	<API_ROOT >/**networks**	-d tissue=<string> -d genes=<string> [-d num_genes=<int>] [-d prior=<real>] [-d enum=<bool>]	
POST	<API_ROOT >/**networks**/evidence	-d tissue=<string> -d source=<string> -d target=<string>	For a pair of genes predicted to be functionally related in a tissue context: **tissue:** Name of the tissue **source, target:** Names of genes in the pair
GET	<API_ROOT>/**imp**/<organism>/<gene>[+<gene2>[+…]]	[?num_genes=<int>] [&enum=<bool>]	**organism:** One of 6 organisms supported **genes**, **num_genes**, and **enum** fields are same as in the **networks** endpoint.
POST	<API_ROOT >/**imp**	-d organism=<string> -d genes=<string> [-d num_genes=<int>] [-d enum=<bool>]	
POST	<API_ROOT>/**netwas**/jobs	-F gwas_file=<@file> -F gwas_format=<string> -F tissue=<string> -F p_value=<real> [-F title=<string>] [-F email=<string>]	**gwas_file:** An existing GWAS results file to be reprioritized by NetWAS **gwas_format:** Format of the input GWAS file (e.g. Vegas, CSV, etc.) **tissue:** One of 144 human tissues supported **p_value:** *P*-value threshold for statistical significance **title:** Optional label to assign to this NetWAS job (useful for tracking multiple jobs) **email**: Optional email that will receive notification of job results
GET	<API_ROOT>/**netwas**/jobs/<job_id>		**job_id:** A UUID that identifies an existing NetWAS job
POST	<API_ROOT>/**netwas**/jobs	-d job_id=<uuid>	
POST	<API_ROOT>/**netwas**/jobs/<job_id>/start		
GET	<API_ROOT>/**tissues**		Returns a list of valid tissue names.
GET	<API_ROOT>/**tissues**/check/<tissue>[+<tissue2>[+…]]		Checks validity of given tissue name[s]. **tissues:** One or more tissue names to be validated
POST	<API_ROOT>/**tissues**/check	-d tissues=<string>	
GET	<API_ROOT>/**genes**/check/<gene>[+<gene2>[+…]]		Checks validity of given gene name[s]. **genes:** One or more gene names to be validated
POST	<API_ROOT>/**genes**/check	-d genes=<string>	

^ Optional version fields are omitted for brevity. A version number can be added after the endpoint prefix, e.g., **netwas/1.0/jobs**.

man. The following set, labeled **imp**, is used for querying integrated global functional networks for six model organisms and human. Finally, the **netwas** endpoints are used for initializing, starting and monitoring NetWAS tasks. The API also includes several helper functions which can query names of supported tissue networks or validate gene names. These are useful for programmatically building web forms that use the API.

Many of the API endpoints accept both GET and POST methods. Consider the following two requests for the functional relationships in the brain network around the SNCA gene:

```
curl http://giant-api.princeton.edu/networks/brain/SNCA

curl http://giant.api.princeton.edu/networks -d tissue=brain \
                                             -d genes=SNCA   \
                                             -d prior=0.15
```

The first form, which invokes a direct hyperlink using the GET method, is convenient to use in a web browser and can be easily bookmarked or cached. The second form, which uses the POST method, embeds query parameters inside the HTTP message body instead of the URL. This allows more complex queries with additional parameters, and may be preferable for programmatic execution in scripts.

The web services underlying GIANT API often use predictive models trained on a continuously evolving body of datasets and gold standards. As the size and accuracy of these data improve over time, predictions returned by GIANT API for the same query parameters may also change. To ensure future reproducibility of current predictions, GIANT API allows users to specify an optional version number in their queries. A version number uniquely identifies a point-in-time snapshot of functional networks (e.g. for **imp** and **networks** endpoints) and services (e.g. the **netwas** endpoint). For example, we can modify the previous query so that it always uses the first version of the networks:

```
curl http://giant-api.princeton.edu/networks/1.0/brain/SNCA
```

Without an explicit version number, GIANT API uses the most recent version of each network or service by default. Where applicable, API responses also contain date or version information for networks and underlying datasets, e.g. **evidence** endpoint in Figure 1.

GIANT API is hosted at Princeton University and is freely accessible at **giant-api.princeton.edu**

(denoted API_ROOT in Table 1). API documentation and wiki is publicly available on GitHub (https://github.com/FunctionLab/giant-api). This repository also contains code samples and client libraries. Our JavaScript visualization module and relevant demos are available at the BioJS registry (http://biojs.io/d/giant-api-biojs).

SUMMARY

GIANT API allows biomedical researchers to explore and work with functional gene networks programmatically. API users can query both *tissue-specific* and *global networks* in human and other model organisms, as well invoke related computational methods such as NetWAS. As a RESTful API, GIANT uses ubiquitous standards for data representation and access, thus encouraging dissemination of new datasets and computational methods. Researchers can use the API to replace manual steps in their research workflows with automatable scripts, easily test multiple hypotheses and rapidly follow up on the results from high-throughput experiments.

In addition to biomedical research scientists, GIANT API also benefits developers working in computational biology. Using the API as a building block, programmers can develop additional web services for functional network analysis and visualization. By providing an abstraction layer between data access and presentation, GIANT API makes it easy for genomic information to be consumed by a diverse set of clients with minimal code repetition. For example, a website, a mobile app and a desktop app can all use the same API calls for data retrieval, with the only code differences being confined to presentation. This also allows developers to write minimalist wrapper libraries, which extend existing API services to new programming languages.

FUNDING

National Science Foundation (NSF) career award [DBI-0546275]; National Institutes of Health [R01 GM071966, R01 HG005998, T32 HG003284]; National Institute of General Medical Sciences (NIGMS) Center of Excellence [P50 GM071508]; Funding for open access charge: Simons Foundation.

Conflict of interest statement. None declared.

REFERENCES

1. Goya,J., Wong,A.K., Yao,V., Krishnan,A., Homilius,M. and Troyanskaya,O.G. (2015) FNTM: a server for predicting functional networks of tissues in mouse. *Nucleic Acids Res.*, **43**, W182–W187.
2. Greene,C.S., Krishnan,A., Wong,A.K., Ricciotti,E., Zelaya,R.A., Himmelstein,D.S., Zhang,R., Hartmann,B.M., Zaslavsky,E., Sealfon,S.C. *et al.* (2015) Understanding multicellular function and disease with human tissue-specific networks. *Nat. Genet.*, **32**, 453–465.
3. Guan,Y., Gorenshteyn,D., Burmeister,M., Wong,A.K., Schimenti,J.C., Handel,M.A., Bult,C.J., Hibbs,M.A. and Troyanskaya,O.G. (2012) Tissue-specific functional networks for prioritizing phenotype and disease genes. *PLoS Comput. Biol.*, **8**, e1002694.
4. Chikina,M.D., Huttenhower,C., Murphy,C.T. and Troyanskaya,O.G. (2009) Global prediction of tissue-specific gene expression and context-dependent gene networks in Caenorhabditis elegans. *PLoS Comput. Biol.*, **5**, e1000417.
5. Wong,A.K., Krishnan,A., Yao,V., Tadych,A. and Troyanskaya,O.G. (2015) IMP 2.0: a multi-species functional genomics portal for integration, visualization and prediction of protein functions and networks. *Nucleic Acids Res.*, **43**, W128–W133.
6. Chatr-Aryamontri,A., Breitkreutz,B.-J., Oughtred,R., Boucher,L., Heinicke,S., Chen,D., Stark,C., Breitkreutz,A., Kolas,N., O'Donnell,L. *et al.* (2015) The BioGRID interaction database: 2015 update. *Nucleic Acids Res.*, **43**, D470–D478.
7. Kanehisa,M., Sato,Y., Kawashima,M., Furumichi,M. and Tanabe,M. (2016) KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Res.*, **44**, D457–D462.
8. Kolesnikov,N., Hastings,E., Keays,M., Melnichuk,O., Tang,Y.A., Williams,E., Dylag,M., Kurbatova,N., Brandizi,M., Burdett,T. *et al.* (2015) ArrayExpress update-simplifying data submissions. *Nucleic Acids Res.*, **43**, D1113–D1116.
9. Szklarczyk,D., Franceschini,A., Wyder,S., Forslund,K., Heller,D., Huerta-Cepas,J., Simonovic,M., Roth,A., Santos,A., Tsafou,K.P. *et al.* (2015) STRING v10: Protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.*, **43**, D447–D452.
10. Polymeropoulos,M.H., Lavedan,C., Leroy,E., Ide,S.E., Dehejia,A., Dutra,A., Pike,B., Root,H., Rubenstein,J., Boyer,R. *et al.* (1997) Mutation in the alpha-synuclein gene identified in families with Parkinson's disease. *Science*, **276**, 2045–2047.
11. Bonifati,V., Rizzu,P., van Baren,M.J., Schaap,O., Breedveld,G.J., Krieger,E., Dekker,M.C.J., Squitieri,F., Ibanez,P., Joosse,M. *et al.* (2003) Mutations in the DJ-1 gene associated with autosomal recessive early-onset parkinsonism. *Science*, **299**, 256–259.
12. Randall,J.C., Winkler,T.W., Kutalik,Z., Berndt,S.I., Jackson,A.U., Monda,K.L., Kilpeläinen,T.O., Esko,T., Mägi,R., Li,S. *et al.* (2013) Sex-stratified genome-wide association studies including 270, 000 individuals show sexual dimorphism in genetic loci for anthropometric traits. *PLoS Genet.*, **9**, e1003500.
13. Bostock,M., Ogievetsky,V. and Heer,J. (2011) D3 data-driven documents. *IEEE Trans. Vis. Comput. Graph.*, **17**, 2301–2309.
14. Gómez,J., García,L.J., Salazar,G.A., Villaveces,J., Gore,S., García,A., Martín,M.J., Launay,G., Alcántara,R., Ayllón,N.D.T. *et al.* (2013) BioJS: an open source JavaScript framework for biological data visualization. *Bioinformatics.*, **29**, 1103–1104.