API Access to SGD and The Alliance of Genome Resources

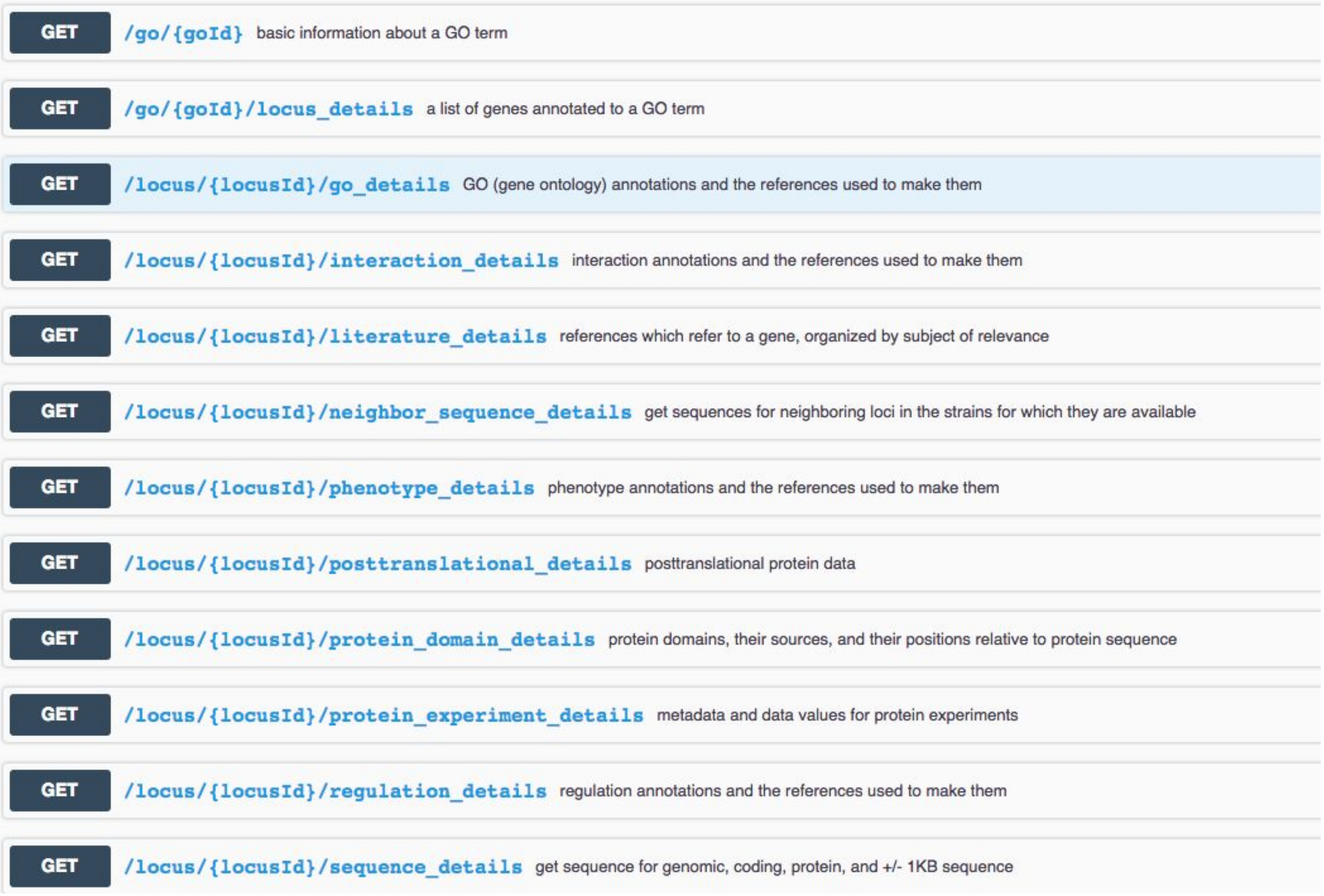
Travis Sheppard, Gail Binkley, Felix Gondwe, Kalpana Karra, Ajay Shrivatsav Vichanthangal Prathivadhib, Shuai Weng, Pedro H.R. de Assis, J. Michael Cherry

Department of Genetics, Stanford University, Stanford, CA 94305

The *Saccharomyces* Genome Database (SGD; https://www.yeastgenome.org) provides a plethora of information traditionally known to users through a website. In order to enable programmatic access to the same data, SGD has developed a public application programming interface (API) which has all the same data as the website. Everything which appears on SGD webpages comes though the same API which is now available to the public. The HTTPS API is documented at http://bit.ly/sgddocs and http://bit.ly/sgdsmartapi. Some example python scripts which access the data can be found at http://bit.ly/apiexamples. We also demonstrate similar access to The Alliance of Genome Resources, which has integrated data from several model organism databases. The Alliance database is available to download as a Docker container running a populated Neo4j database. This work is supported by a grant from the NHGRI (U41 HG001315).

SGD API Documentation Online

- SmartAPI registry (http://bit.ly/sgdsmartapi)
- Github documentation (http://bit.ly/sgddocs)



GET	`/go/{goId}`	basic information about a GO term
GET	`/go/{goId}/locus_details`	a list of genes annotated to a GO term
GET	`/locus/{locusId}/go_details`	GO (gene ontology) annotations and the references used to make them
GET	`/locus/{locusId}/interaction_details`	interaction annotations and the references used to make them
GET	`/locus/{locusId}/literature_details`	references which refer to a gene, organized by subject of relevance
GET	`/locus/{locusId}/neighbor_sequence_details`	get sequences for neighboring loci in the strains for which they are available
GET	`/locus/{locusId}/phenotype_details`	phenotype annotations and the references used to make them
GET	`/locus/{locusId}/posttranslational_details`	posttranslational protein data
GET	`/locus/{locusId}/protein_domain_details`	protein domains, their sources, and their positions relative to protein sequence
GET	`/locus/{locusId}/protein_experiment_details`	metadata and data values for protein experiments
GET	`/locus/{locusId}/regulation_details`	regulation annotations and the references used to make them
GET	`/locus/{locusId}/sequence_details`	get sequence for genomic, coding, protein, and +/- 1KB sequence

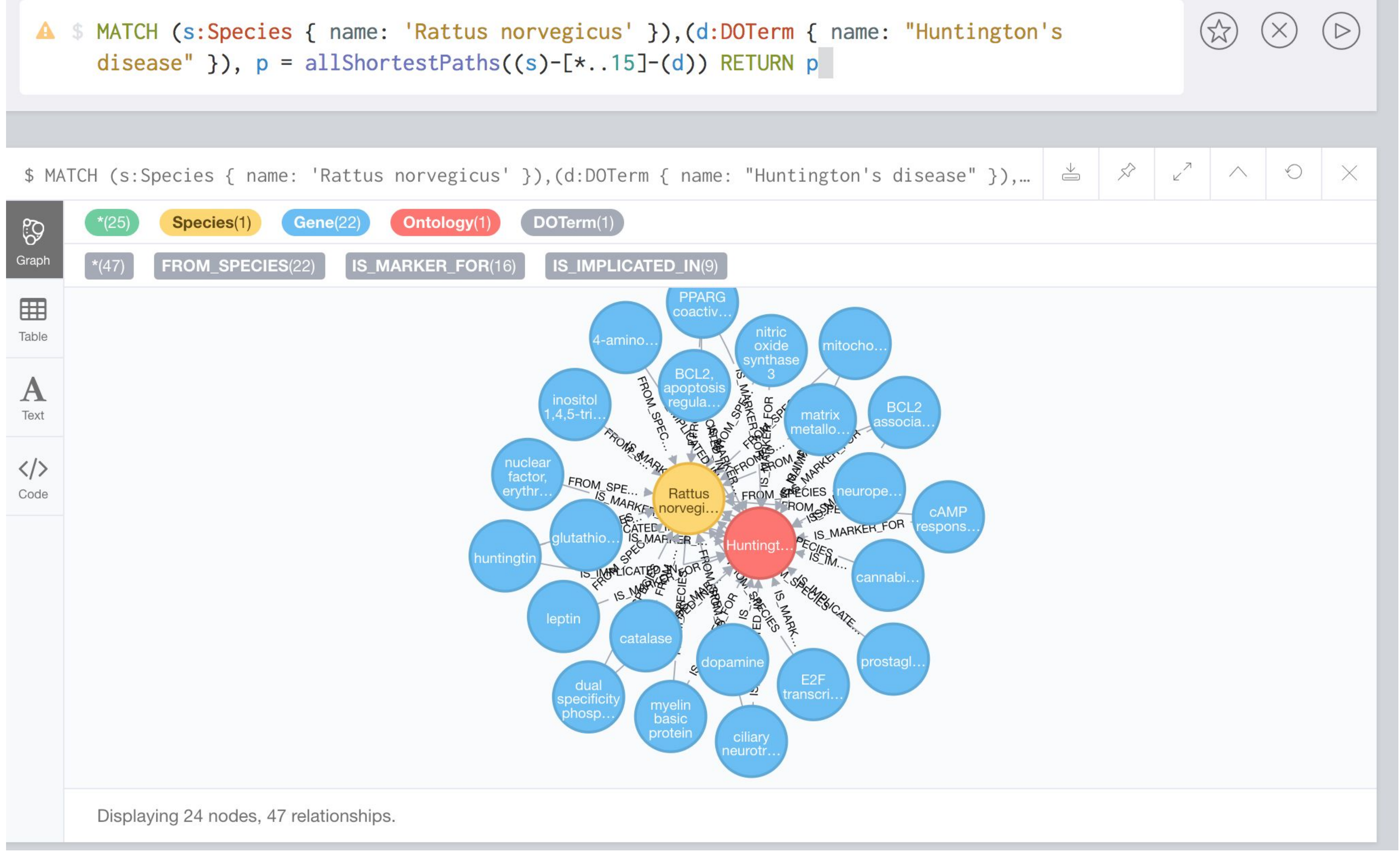
Python Script Examples

- http://bit.ly/apiexamples
    - Chromosome I sequences
    - Genes annotated to phenotype
    - Basic gene information

```python
import requests

SGD_BASE_URL = 'https://www.yeastgenome.org/webservice'

def print_act1_info():
    url = SGD_BASE_URL + '/locus/act1'
    response = requests.get(url=url).json()
    # print references
    print('ACT1 REFERENCES')
    for ref in response['references']:
        name = ref['display_name']
        pmid = str(ref['pubmed_id'])
        print(pmid + ': ' + name)
     # print large scale phenotypes
    print('ACT1 OVEREXPRESSION LARGE SCALE PHENOTYPES')
    for pheno in response['phenotype_overview']['large_scale_phenotypes']['overexpression']:
        name = pheno['display_name']
        print(name)

if __name__ == '__main__':
    print_act1_info()

```

Alliance Neo4j

- integrated data from several model organisms
- Neo4j graph database
- available as Docker container
- clone github repo from http://bit.ly/MODneo4j then run `make data-run`



```
$ MATCH (s:Species { name: 'Rattus norvegicus' }),(d:DOTerm { name: "Huntington's disease" }), p = allShortestPaths((s)-[*..15]-(d)) RETURN p
```

```
$ MATCH (s:Species { name: 'Rattus norvegicus' }),(d:DOTerm { name: "Huntington's disease" }),…
```

Displaying 24 nodes, 47 relationships.

www.yeastgenome.org