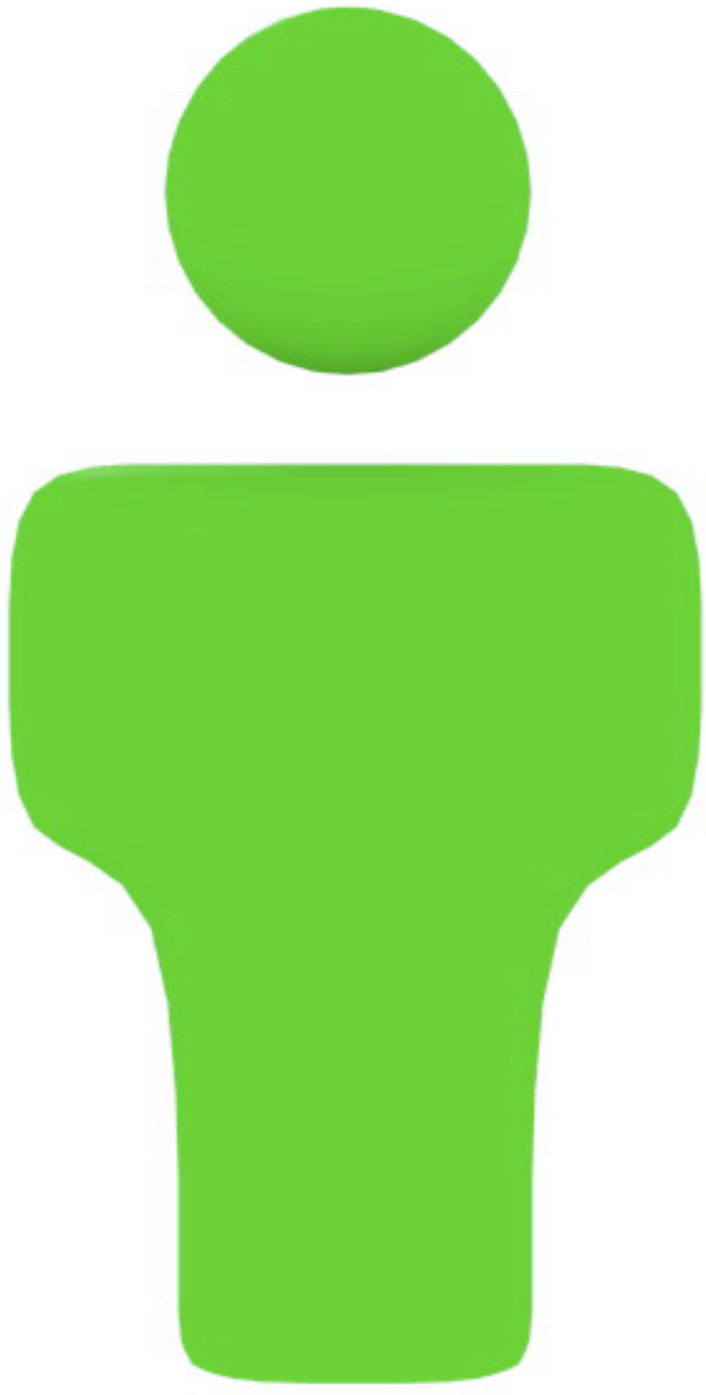
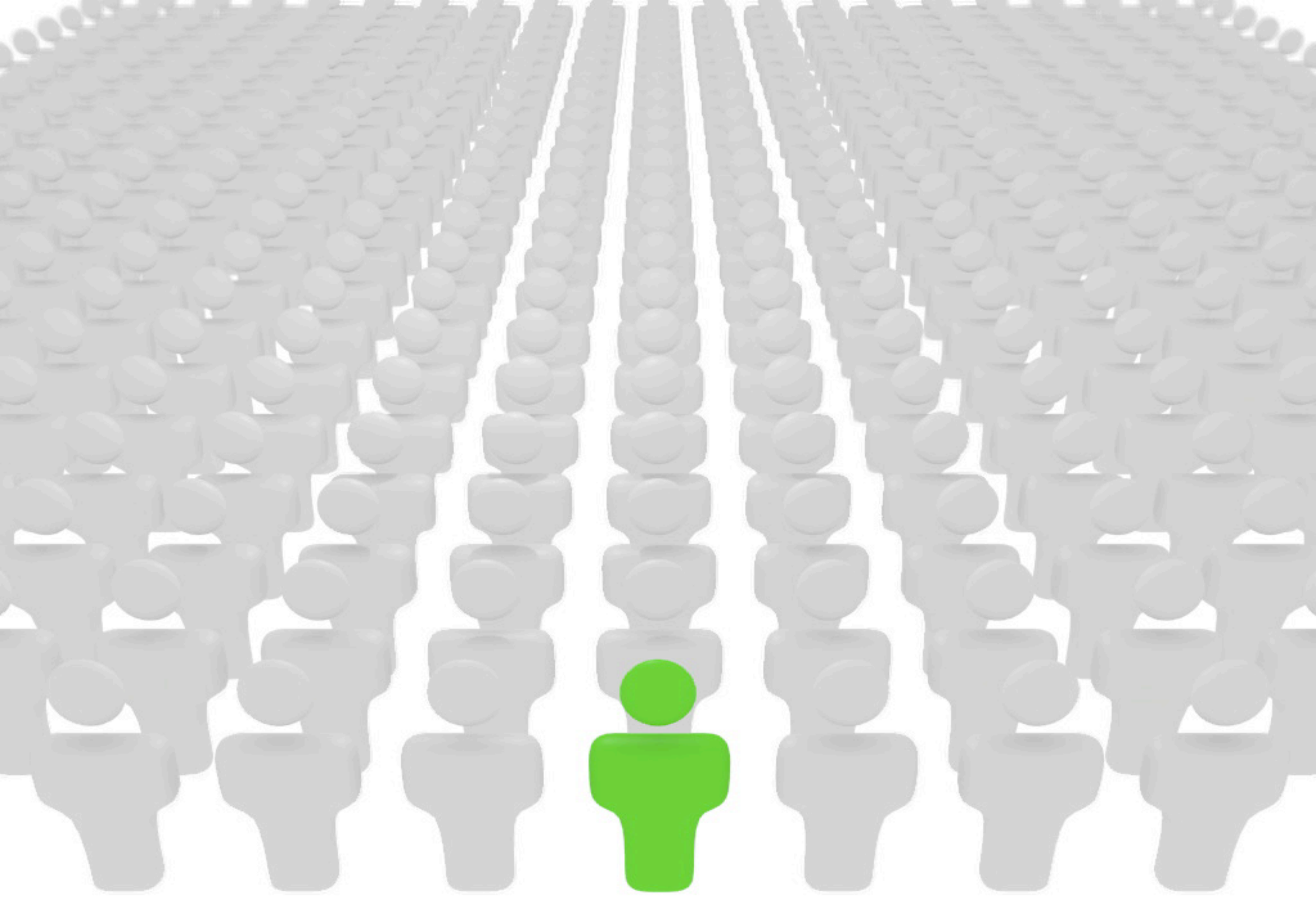
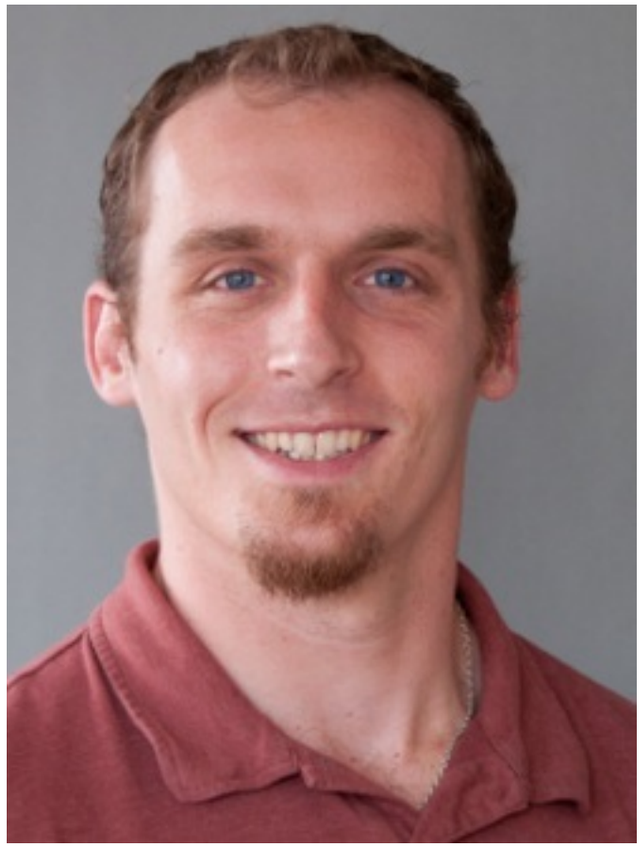
MongoDB and the Connectivity Map

making connections between genetics and disease







Corey

Rajiv

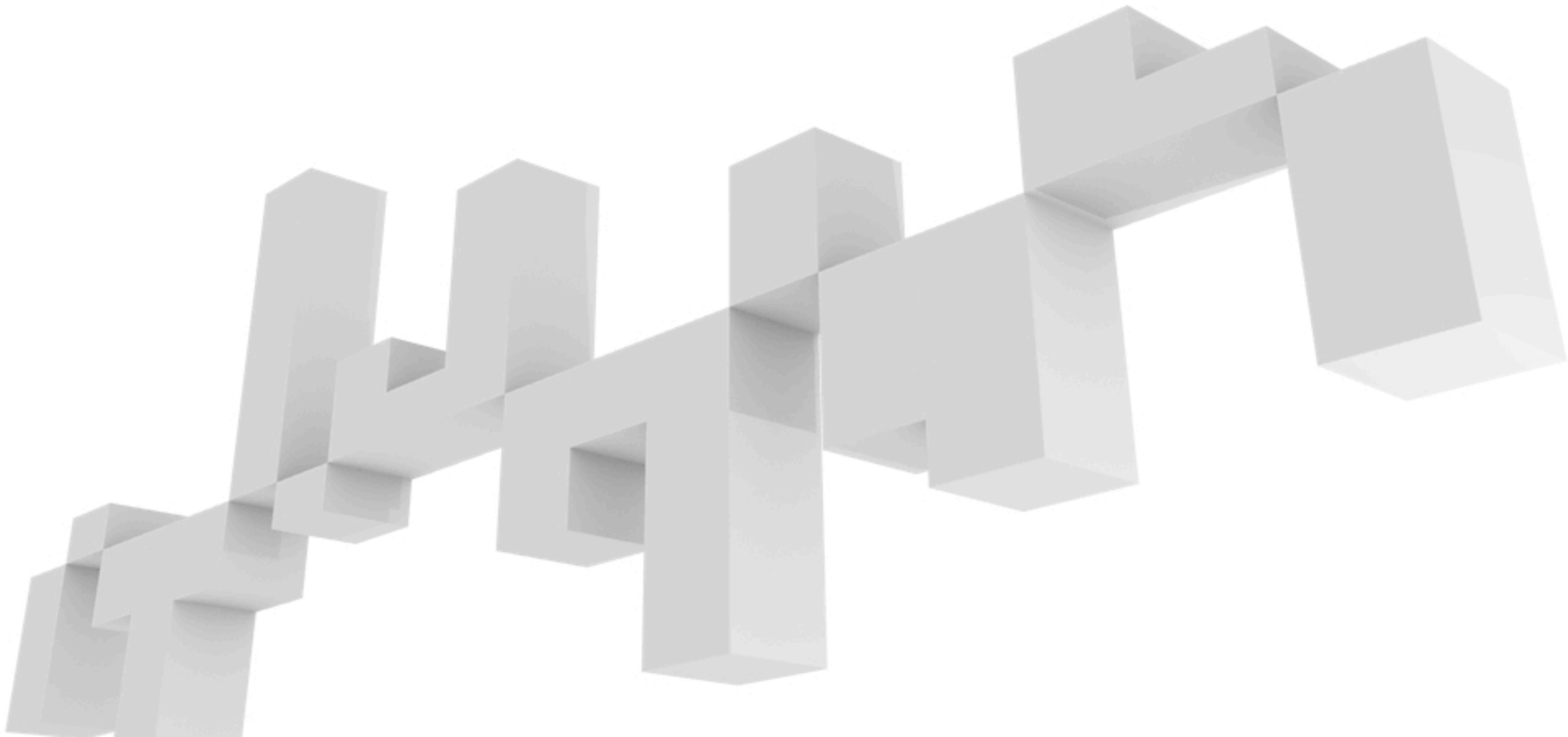
Gene Expression

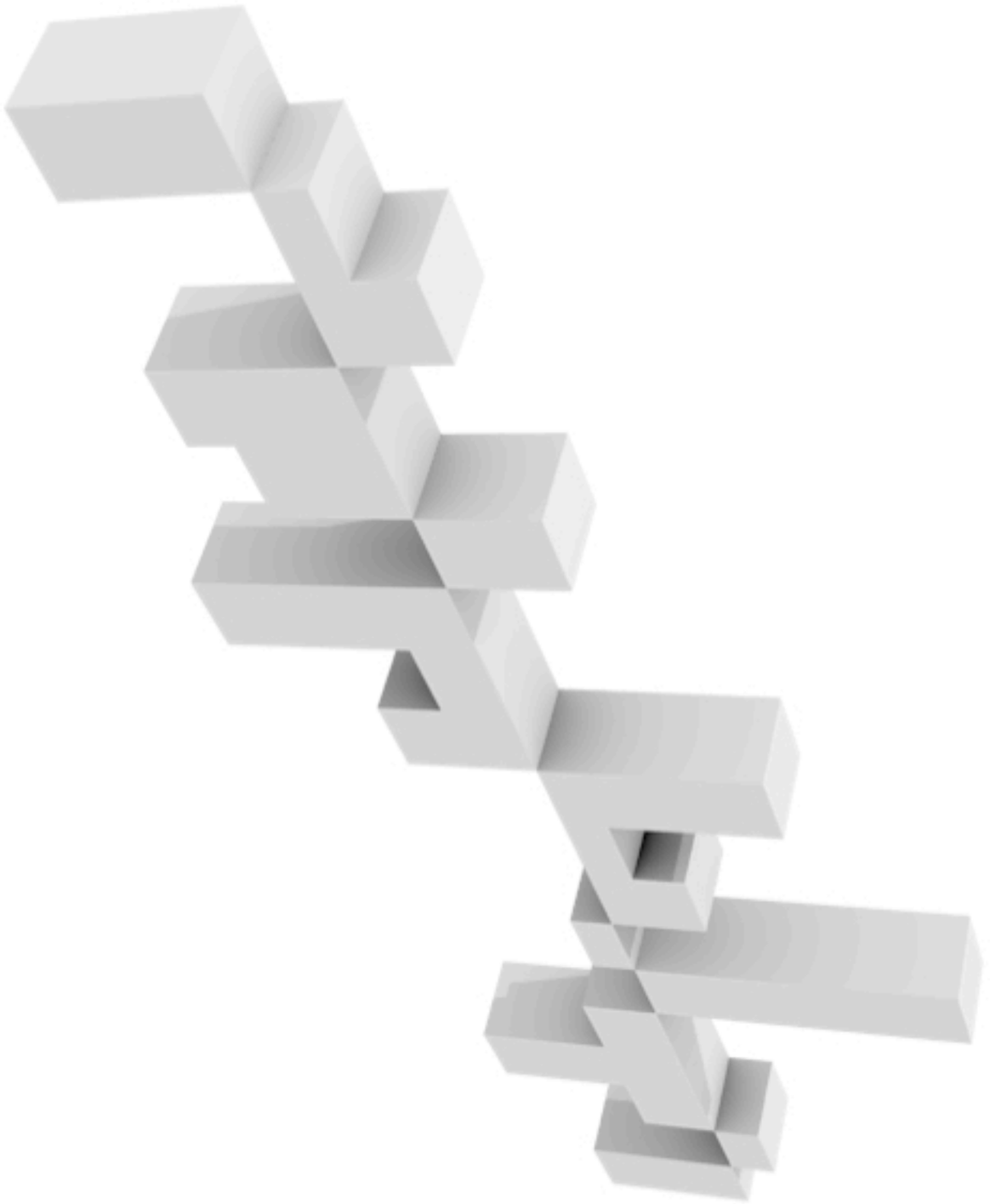
a common language

Gene A B C D E

Gene ABCDE

Disease Normal **Drug**





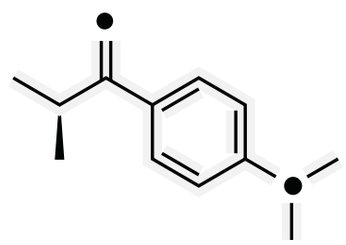
~7,000 experiments
Over 19,000 registered users
Cited by over 1,200 scientific reports

2006

2014

CMap-LINCS dataset

1.4 million gene expression profiles

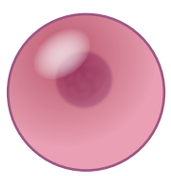


12,488 Compounds

- FDA approved drugs
- Bioactive tool compounds
- Screening hits

3,800 Genes (shRNA & cDNA)

- Targets/pathways of approved drugs
- Candidate disease genes
- Community nominations

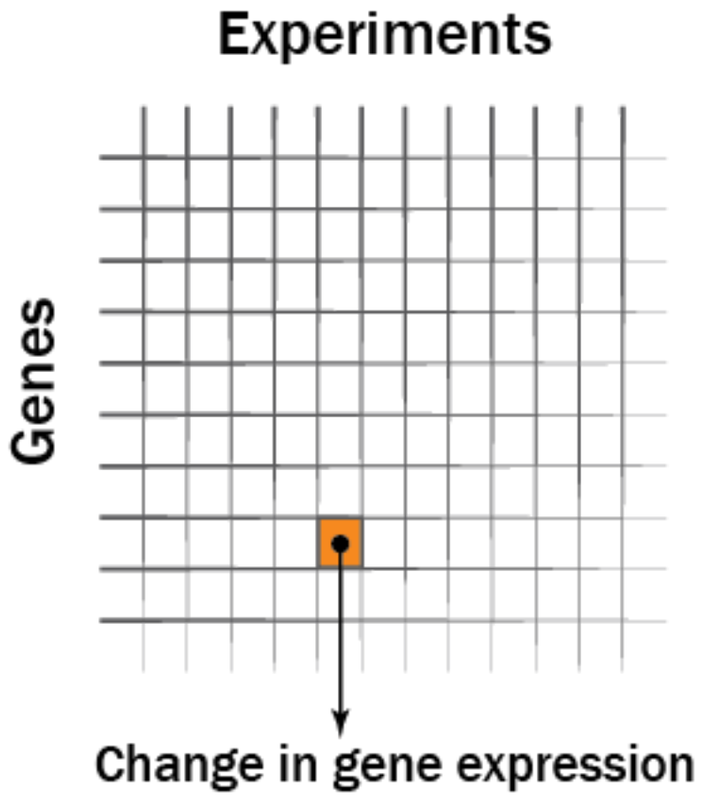


15 Cell types

- Banked primary cell types
- Cancer cell lines
- Primary hTERT-immortalized
- Patient-derived iPS cells
- Community nominated

CMap Data

Easy to describe, tough to Model



- Diverse use-cases
- Users with varying technical expertise
- Annotations are complex and incomplete
- Frequent updates

Data Model

An agile philosophy keeps the model tractable

Store just what's needed

Refactor frequently

Test and use daily

Data Model

An inventory of signatures

siginfo

```
{
    treatment_id: "BRD-K02404261",
    treatment_name: "caffeine",
    cell_id: "MCF7",
    treatment_duration: "6 h",
    treatment_type: "trt_cp"
}
```

Data Model

Shared fields as separate collections

siginfo

```
{
    treatment_id: "BRD-K02404261",
    treatment_name: "caffeine",
    cell_id: "MCF7",
    treatment_duration: "6 h",
    treatment_type: "trt_cp"
}
```

cellinfo

```
{
    cell_id: "MCF7",
    cell_type: "cancer",
    cell_lineage: "breast"
    cell_source: "ATCC",
    cell_source_id: "HTB-2
    gender: "F",
    is_from_metastasis: "Y
}
```

pertinfo

```
{
    treatment_id: "BRD-K02404261",
    treatment_name: "caffeine",
    molecular_wt: 194.191,
    molecular_formula: "C8H10N4O2",
    canonical_smiles: "Cn1cnc2n(C)c(=O)n(C)c(=O)c12",
    vendor: "MicroSource"
}
```

Data Model

Add computed fields and external meta-data

siginfo

```
{
    treatment_id: "BRD-K02404261",
    treatment_name: "caffeine",
    cell_id: "MCF7",
    treatment_duration: "6 h",
    treatment_type: "trt_cp",
    distil_cc_q75: 0.25,
  - geneset_up: [
        "203067_at",
        "203815_at",
        "208478_s_at",
        "217312_s_at",
        "204698_at",
        "212725_s_at"
    ],
  - geneset_down: [
        "203067_at",
        "203815_at",
        "208478_s_at",
        "217312_s_at",
        "204698_at",
        "212725_s_at"
    ]
}
```

cellinfo

```
{
    cell_id: "MCF7",
    cell_type: "cancer",
    cell_lineage: "breast",
    cell_source: "ATCC",
    cell_source_id: "HTB-22",
    gender: "F",
    is_from_metastasis: "Y",
  - mutations: [
        "BCR",
        "EPHA5",
        "ERBB4",
        "ERCC6",
        "MAP3K13",
        "MYH9",
        "NLRP1",
        "PIK3CA",
        "PTPRD",
        "TCERG1L",
        "TNIK",
        "TTN",
        "WNK2"
    ]
}
```

Data Model

Duplicate data to optimize lookups

siginfo

```
{
    treatment_id: "BRD-K02404261",
    treatment_name: "caffeine",
    cell_id: "MCF7",
    treatment_duration: "6 h",
    treatment_type: "trt_cp",
    distil_cc_q75: 0.25,
  - geneset_up: [
        "203067_at",
        "203815_at",
        "208478_s_at",
        "217312_s_at",
        "204698_at",
        "212725_s_at"
    ],
  - geneset_down: [
        "203067_at",
        "203815_at",
        "208478_s_at",
        "217312_s_at",
        "204698_at",
        "212725_s_at"
    ]
}
```

pertinfo

```
{
    treatment_id: "BRD-K02404261",
    treatment_name: "caffeine",
    molecular_wt: 194.191,
    molecular_formula: "C8H10N4O2",
    canonical_smiles: "Cn1cnc2n(C)c(=O)n(C)c(=O)c12",
    vendor: "MicroSource",
  - cell_id: [
        "A375",
        "HA1E",
        "HCC515",
        "HEPG2",
        "HT29",
        "MCF7",
        "PC3"
    ]
}
```

APIs

Are awesome, we need more of them

Picked functionality over convention

`/siginfo?q={“cell”:”A”} vs /siginfo/cell/A`

API

MongoDB inspired a rich query syntax

Function	Example
Query	`/siginfo?q={“cell:”A”,”name”:”B”}`
Field selection	`/siginfo?q={}&f={“name”:1}`
Document count	`/siginfo?q={}&c=true`
Document limit	`/siginfo?q={}&l=10`
Skip documents	`/siginfo?q={}&l=10&sk=10`
Sort order	`/siginfo?q={}&s={“name”:-1,”cell”:1}`
Distinct values	`/siginfo?q={}&d=name`
Aggregation	`/siginfo?q={}&g=name`

API

Node and Mongoose enable easy API creation

```
1
2   // MongooseJS model for cellinfo
3   var CellInfoSchema = new Schema({
4       cell_id: String,
5       cell_type: String,
6       cell_lineage: String,
7       cell_source: String,
8       is_from_metastasis: Boolean,
9       mutations: [String]
10  });
11
12  var CellInfo = db.model('CellInfo',
13                          CellInfoSchema, 'cell_info');
```

```
1   /* Execute find */
2   var execFind = function(Collection, args, res){
3       return Collection.find(args.filter, args.fields,
4           {limit: args.limit,
5            skip: args.skip,
6            sort: args.sort_order},
7           function(err, found) {
8               if (!err) {
9                   res.jsonp(found);
10              }
11          });
12  }
```

Language Bindings

JSON as a universal format

Javascript

```
var url = "http://api.lincscloud.org/a2/siginfo?callback=?";
var params = {q:'{"pert_iname":"sirolimus"}',c: true, user_key: "lincsdemo"};
$.getJSON(url,params,function(response){console.log(response);})
```

Python

```
import urllib2
import json
url = 'http://api.lincscloud.org/a2/siginfo?q={"pert_iname":"sirolimus"}&c=1&user_key=lincsdemo'
response = urllib2.urlopen(url).read()
json.loads(response)
```

R

```
library('rjson')
url = 'http://api.lincscloud.org/a2/siginfo?q={"pert_iname":"sirolimus"}&c=1&user_key=lincsdemo'
fromJSON(file=url)
```

lincscloud.org

www.lincscloud.org

lincscloud

cflynn

ACCESS THE DATA

Cell Types Profiled

Pertubagens Assayed

Gene Expression Data

Phosphoproteomics Data

Imaging Data

Download the Data

About

Team

About | For Biologists | For Developers

Our goal is to develop comprehensive signatures of cellular states and tools to analyze them in an effort to understand protein function, small-molecule action, physiological states, and disease characteristics.

LINCS

The Library of Integrated Cellular Signatures (LINCS) is an NIH program which funds the generation of perturbational profiles across multiple cell and perturbation types, as well as read-outs, at a massive scale.

NIH LINCS PROGRAM

LINCSCLOUD

This website, lincscloud.org, brings together datasets and tools from the LINCS consortium.

The Challenge

How can we make LINCS data accessible to researchers of all types so that it can help accelerate biomedical discovery?

Coupled with analytical tools, the vision is to, someday, make it possible for researchers to simply "look up" any cellular response in a genome-scale library of cellular signatures.

To date, LINCS has generated over 1 billion data points of perturbational profiles spanning small-molecules and genetic gain- and loss-of-function across multiple cell types.

1. fish /Users/cflynn (fish)

```
wm398-ae4 [~]:
```

Bookmarks Window Users Help

Query

apps.lincscloud.org/query

lincscloud

Query

Match user-defined gene sets to L1000 signatures

take a tour

cflynn

Name your query

Enter **Up-regulated** genes

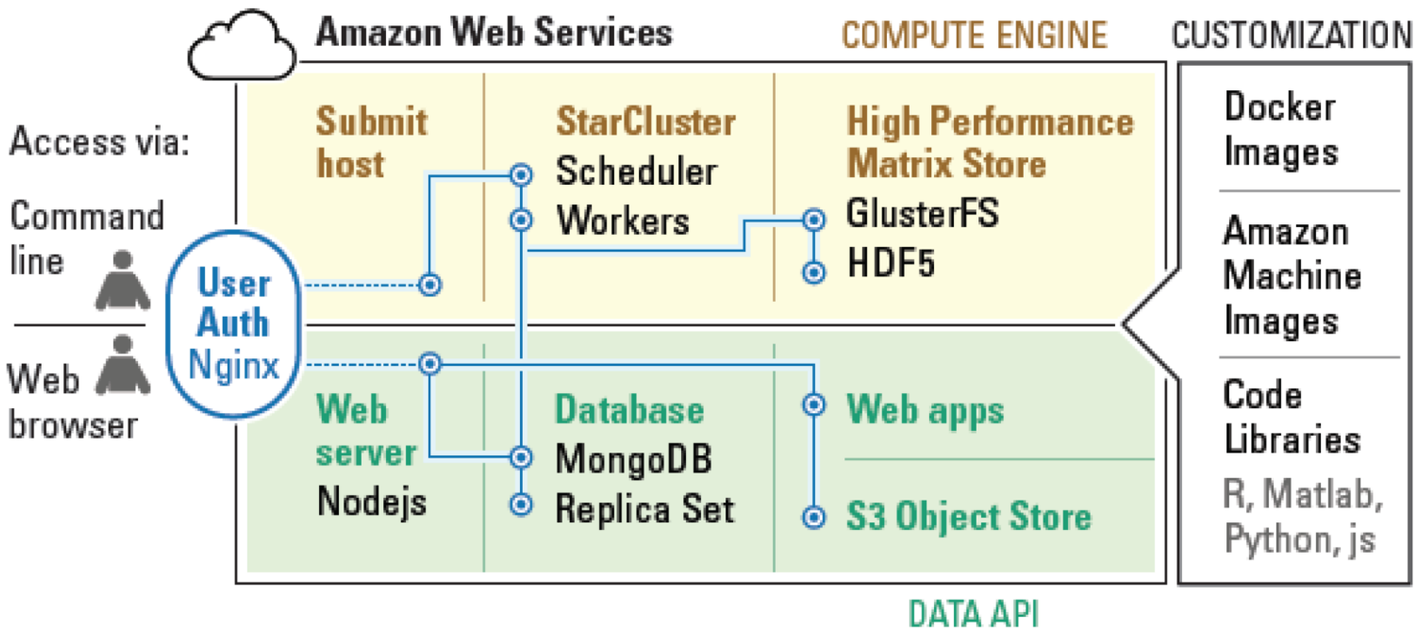
Enter one gene symbol or Affymetrix U133A probe ID per line or drag and drop a plain text file here.

Enter **Down-regulated** genes

Enter one gene symbol or Affymetrix U133A probe ID per line or drag and drop a plain text file here.

Analytic Tools

A compute API liberates command line scripts



Compute API

Messaging handled via a capped collection

```
// Tail the job queue
var stream = queue.find({}).tailable().stream();

// Handle job execution asynchronously
stream.on("data", function(doc){
    Q.nfcall(build_arguments, doc);
        .then(function(obj){
            return Q.nfcall(submit_job, obj.doc, obj.arguments);
        })
        .then(function(job_object){
            return Q.nfcall(poll_job, job_object);
        })
        .then(function(job_object){
            return Q.nfcall(s3_upload, job_object);
        })
        .then(function(job_object){
            return Q.nfcall(cleanup,  job_object);
        })
        .catch(function(err){console.log('error: '+ err.stack)});
});
```

Input Validation

JSON Schema simplifies validation

```
{
    $schema: "http://json-schema.org/draft-04/schema#",
    title: "SigQuestToolSchema",
    type: "object",
  - properties: {
      - uptag: {
            type: "string",
            minLength: 1
        },
      - dntag: {
            type: "string",
            minLength: 1
        },
      - row_space: {
            type: "string",
          - enum: [
                "lm",
                "full"
            ]
        },
      - metric: {
            type: "string",
          - enum: [
                "wtcs",
                "cs"
            ]
        }
    },
    additionalProperties: false,
  - required: [
        "tool_id",
        "uptag",
        "dntag"
    ]
}
```

```
result = tv4.validateMultiple(params, tool_schema);
      if (!result.valid) {
        error_msg = [];
        result.errors.forEach(function(item){
            error_msg.push([item.message, item.dataPath].join(" ").trim())
            });
        errors = {'num_errors': result.errors.length,
                  'errors': error_msg};
        callback(errors, params);
        };
```

```
{
    status: "fail",
  - result: {
        num_errors: 2,
      - errors: [
            "Missing required property: dntag",
            "String is too short (0 chars), minimum 1 /uptag"
        ]
    }
}
```

Numeric Matrix Data

HDF5 offers efficient storage for large matrices

GCTX : A binary format based on HDF5

Cross platform

Multi-language support

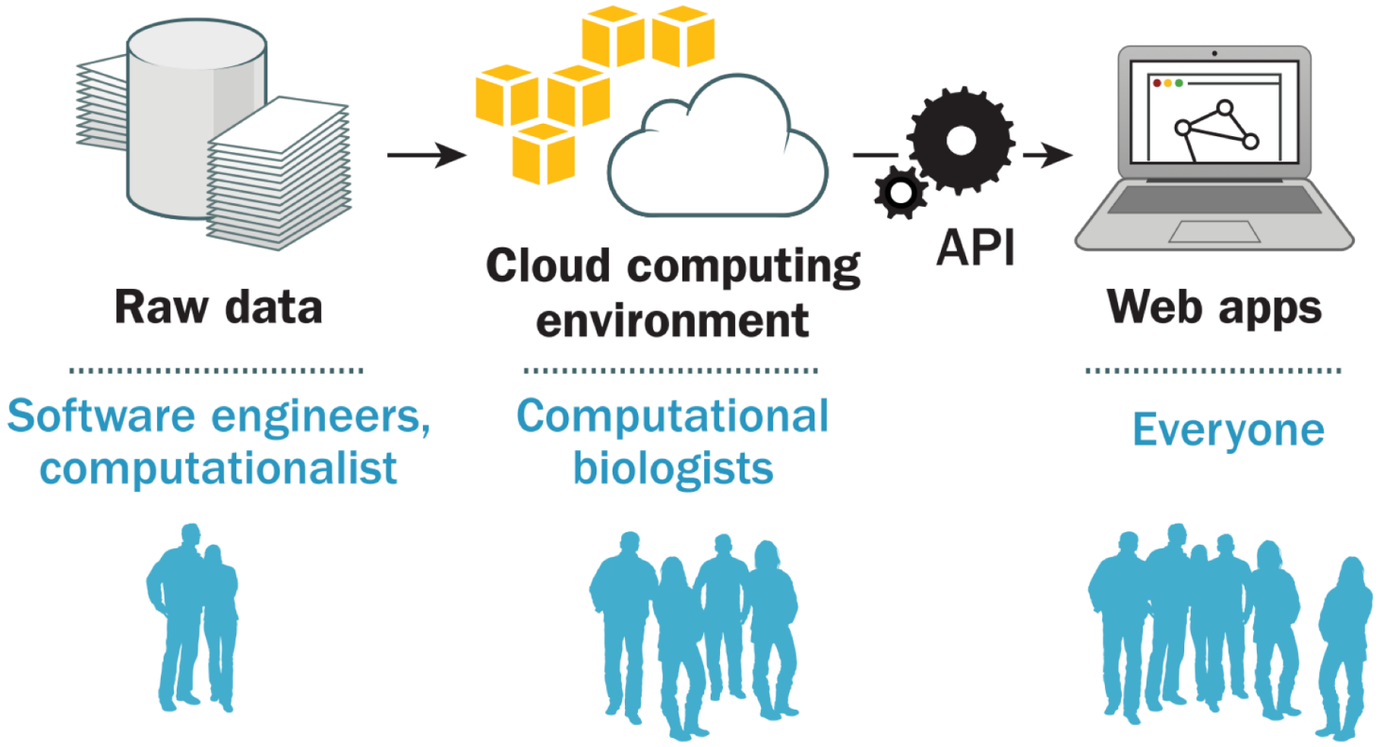
Efficient I/O

Storage size for 30 billion data points is 110 Gb

The HDF Group

Lincscloud

A platform for easy access to perturbational data

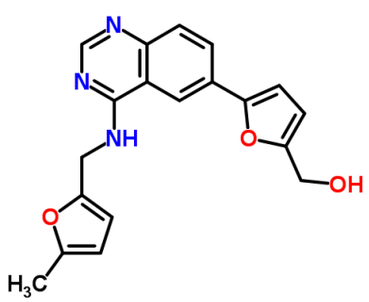


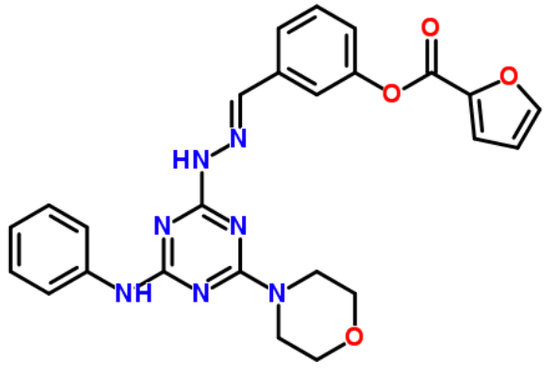
Sign up at lincscloud.org

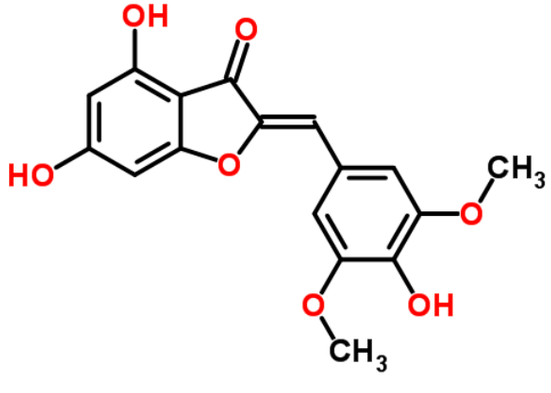
Free for academic use

Predicting Drug Function

Diverse structures, common activities

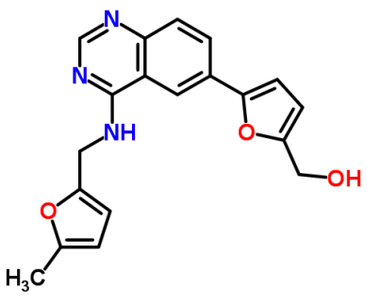






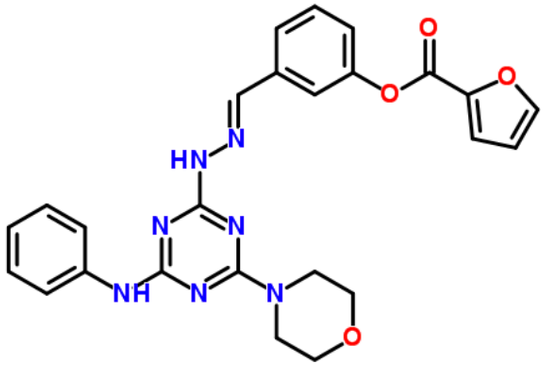
Predicting Drug Function

Diverse structures, common activities



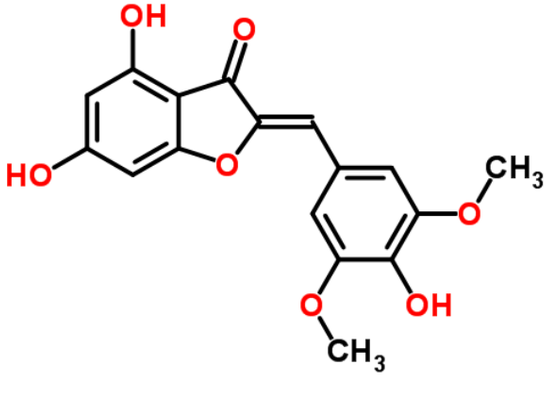
❌ VEGFR inhibitor

❌ PPARG agonist



✔ PI3K/MTOR inhibitor

❌ ROCK inhibitor

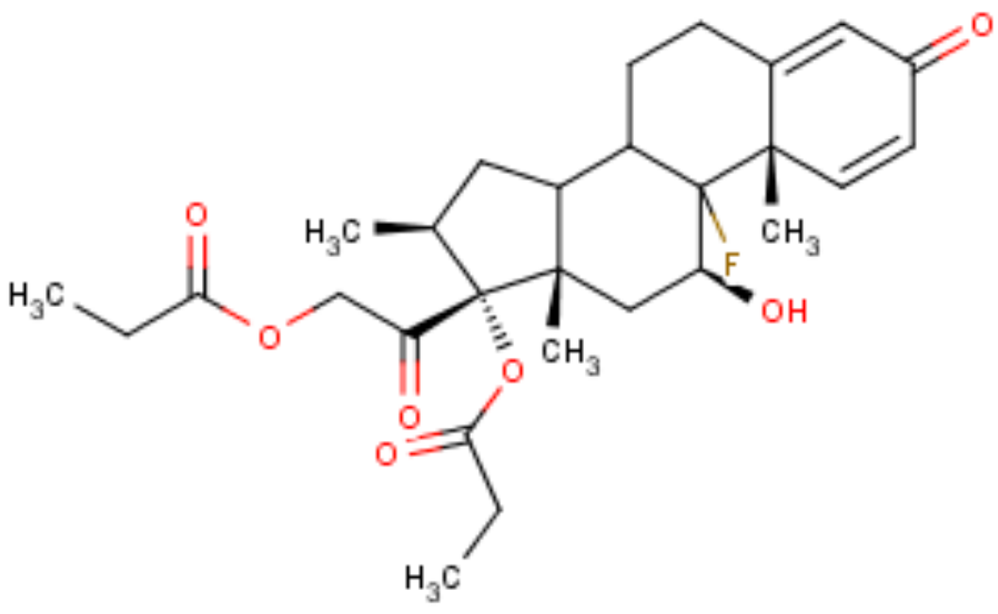


❌ Estrogen agonist

Finding Novel Drug Targets

Repurposing failed drugs

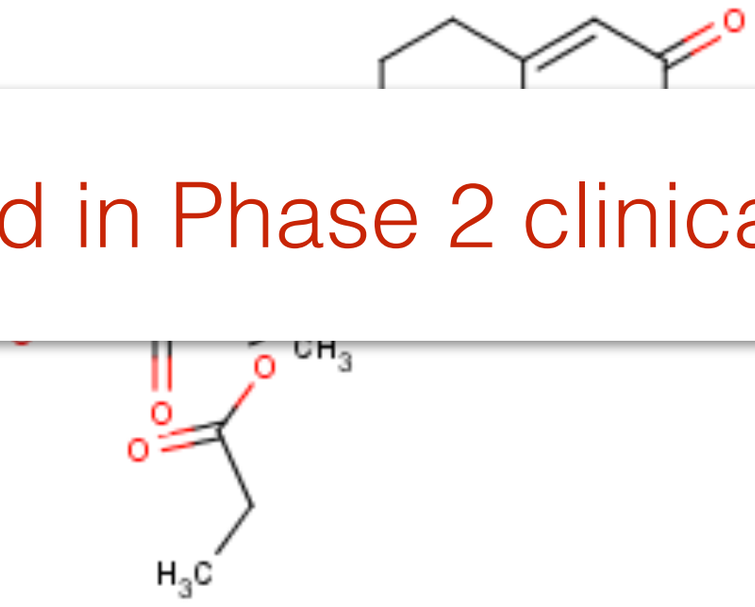
✔ Original target



Finding Novel Drug Targets

Repurposing failed drugs

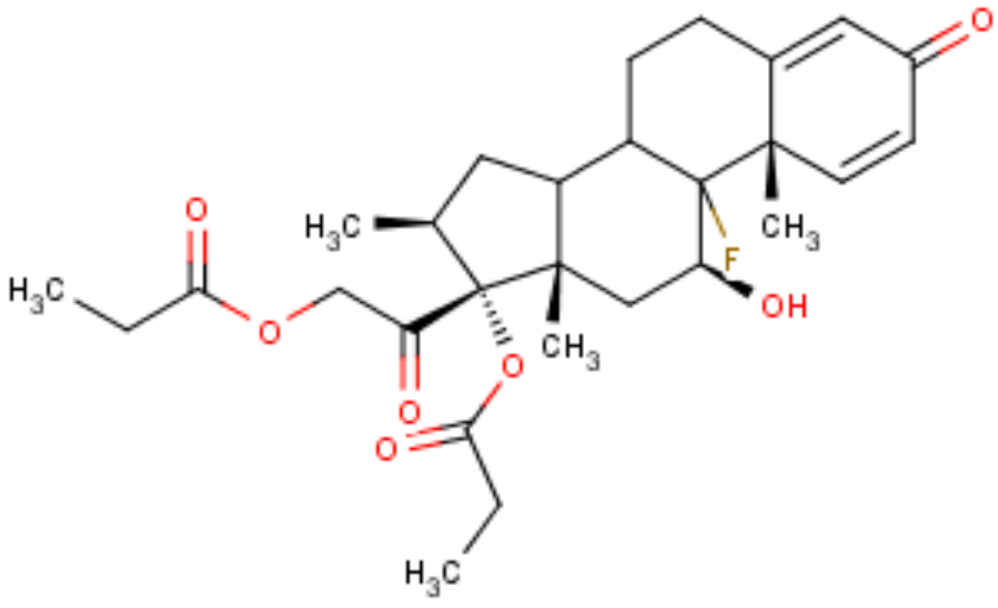
✔ Original target



Failed in Phase 2 clinical trial due to lack of efficacy

Finding Novel Drug Targets

Repurposing failed drugs



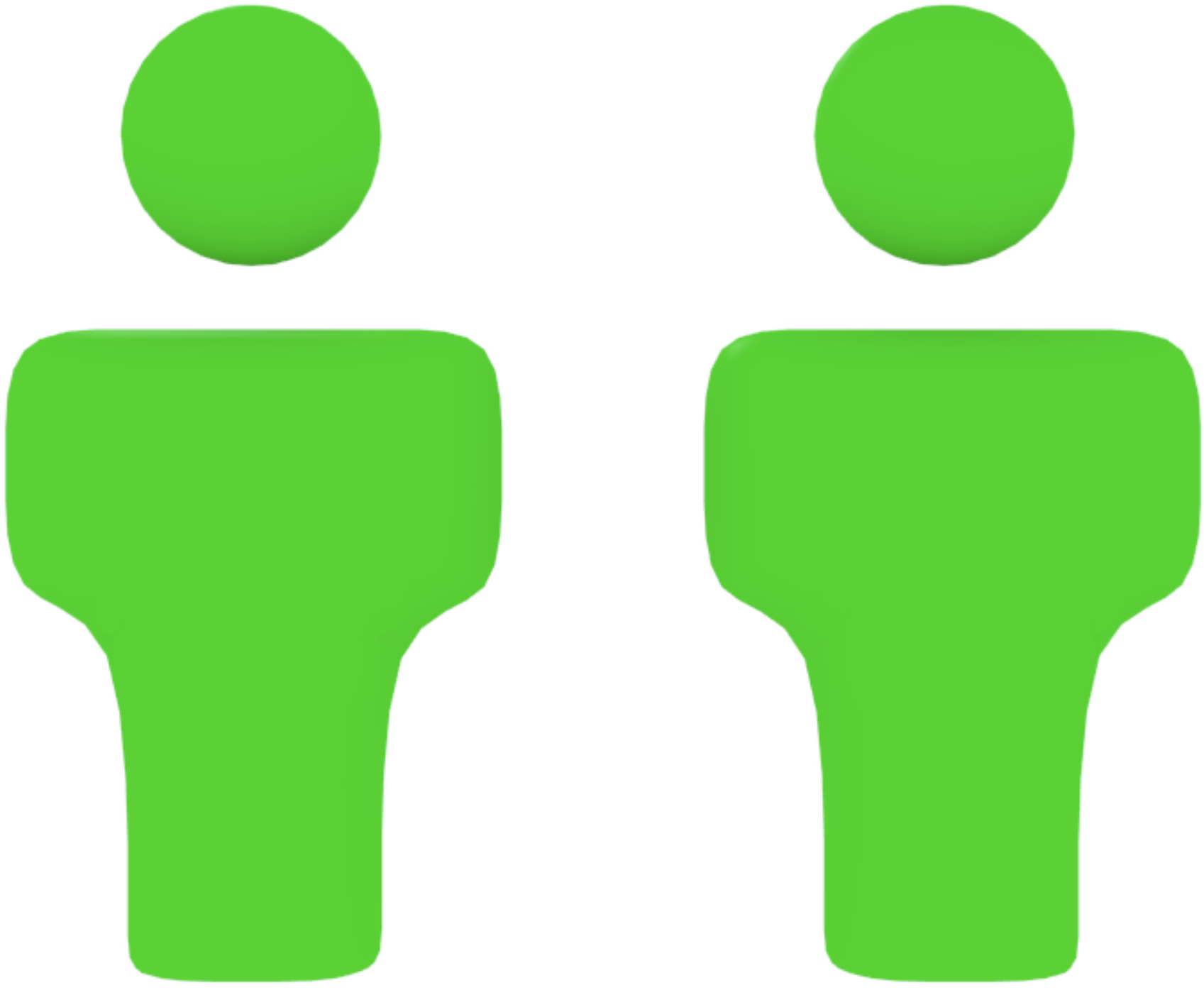
✔ Original target

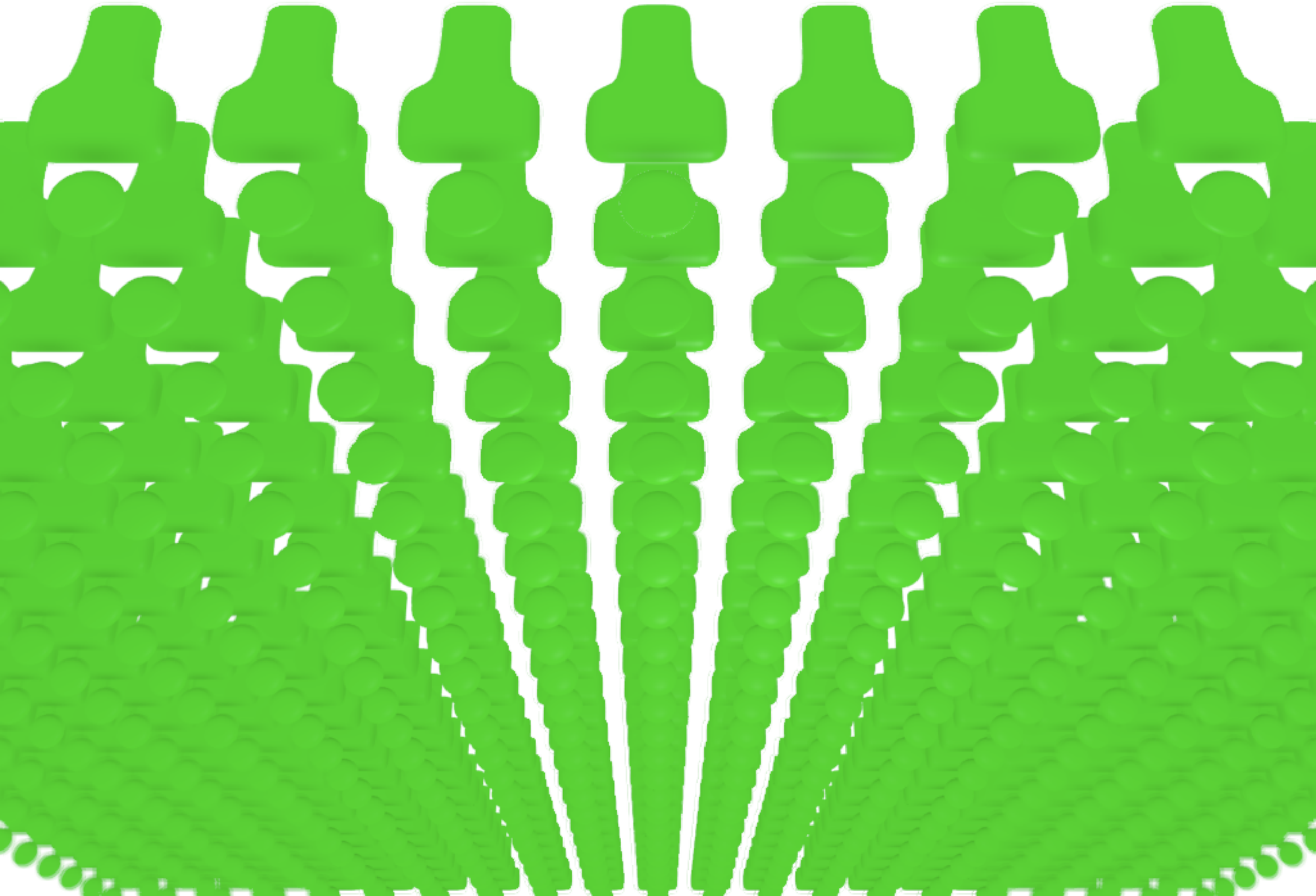
✖ Novel Target A

✔ Novel Target B

✔ Novel Target C

✖ Novel Target D





Acknowledgements

Todd Golub

Core Team: Analysis & Software
Arvind Subramanian
Jacob Asiedu
Larson Hogstrom
Ian Smith
David Lahr
Aravind Subramanian
Josh Gould
Ted Natoli
David Wadden

Core Team: Lab
John Davis
David Peck
Xiaodong Lu
Melanie Donahue
Daniel Lam
Jackie Rosains (Project Manager)

Collaborators
Bang Wong
Steven Corsello (Golub lab)
Jake Jaffe (Proteomics)
David Takeda (Hahn lab)
Pablo Tamayo

Chemistry & Therapeutics
Lucienne Ronco
Josh Bittker
Arthur Liberzon
Mathias Wawer
Paul Clemons

Genetic Perturbation Platform
John Doench
Federica Piccioni
David Root