

Advanced development of an open-source platform for web-based integrative digital image analysis in cancer



David Gutman, MD, PhD
Assistant Professor
Department of Neurology
Emory University
U24 PI



Lee AD Cooper, PhD
Assistant Professor
Department of Biom. Inf.
Emory University
Georgia Institute of Technology
U24 PI



EMORY
UNIVERSITY
SCHOOL OF
MEDICINE



NATIONAL CANCER INSTITUTE
Informatics Technology for
Cancer Research

U24CA194362

The (C)DSA: Developing a platform and infrastructure

The screenshot displays the Cancer Digital Slide Archive (CDSA) web interface. The browser address bar shows the URL `cancer.digitalslidearchive.emory.edu`. The page header includes the "CANCER Digital Slide Archive" logo and navigation links for "Home", "TCGA Resources", and "Help". A secondary navigation bar contains "Apply Filters", "Report Bad Image", "Show Debug Info", "Draw Tools", "Query Metadata", and "Path Report".

On the left side, there is a "Select Patient" dropdown menu with "lgg" selected. Below it, it states "Available Slides: 2642". A vertical list of slide thumbnails is shown, with the middle one highlighted in blue and labeled "TCGA-E1-5305-01A-01-BS1". Below that is another thumbnail labeled "TCGA-DU-5853-01A-01-TS1".

The main area of the interface is a large histology slide viewer showing a high-magnification view of tissue stained with hematoxylin and eosin (H&E). The tissue shows a dense population of cells with purple nuclei and pink cytoplasm/extracellular matrix. A toolbar above the slide includes zoom in (+), zoom out (-), and a refresh icon.

At the bottom of the interface, a status bar displays the text "Current image:TCGA-E1-5305-01A-01-BS1".



<http://www.kitware.com/>

Expertise in building open-source communities

Software process and project management. Design for long-term maintainability and extensibility.



Brian Helba
Project Manager



Deepak Chitajalu
HistomicsTK



Jonathan Beezley
User Interfaces



Zack Mullen
Workflow / DB



Dave Manthey
Visualization



Project Goals:

Infrastructure + algorithms for the management, analysis and integration of digital pathology data

Development philosophy:

Installable, scalable, maintainable, extensible

Open-source community development:

People

12 >



<https://github.com/DigitalSlideArchive>

Project Started May 1st, 2016

Supporting Cancer Research

Digital Slide Archive

TCGA PanCancer Pathology Review (Alex Lazar)

TCGA Analysis Working Groups (All)

Lymphoma Epidemiology of Outcomes Cohort Study (Flowers)

ISIC Melanoma Working Group

Emory Winship Cancer Institute Biobank

Emory Molecular Pathology

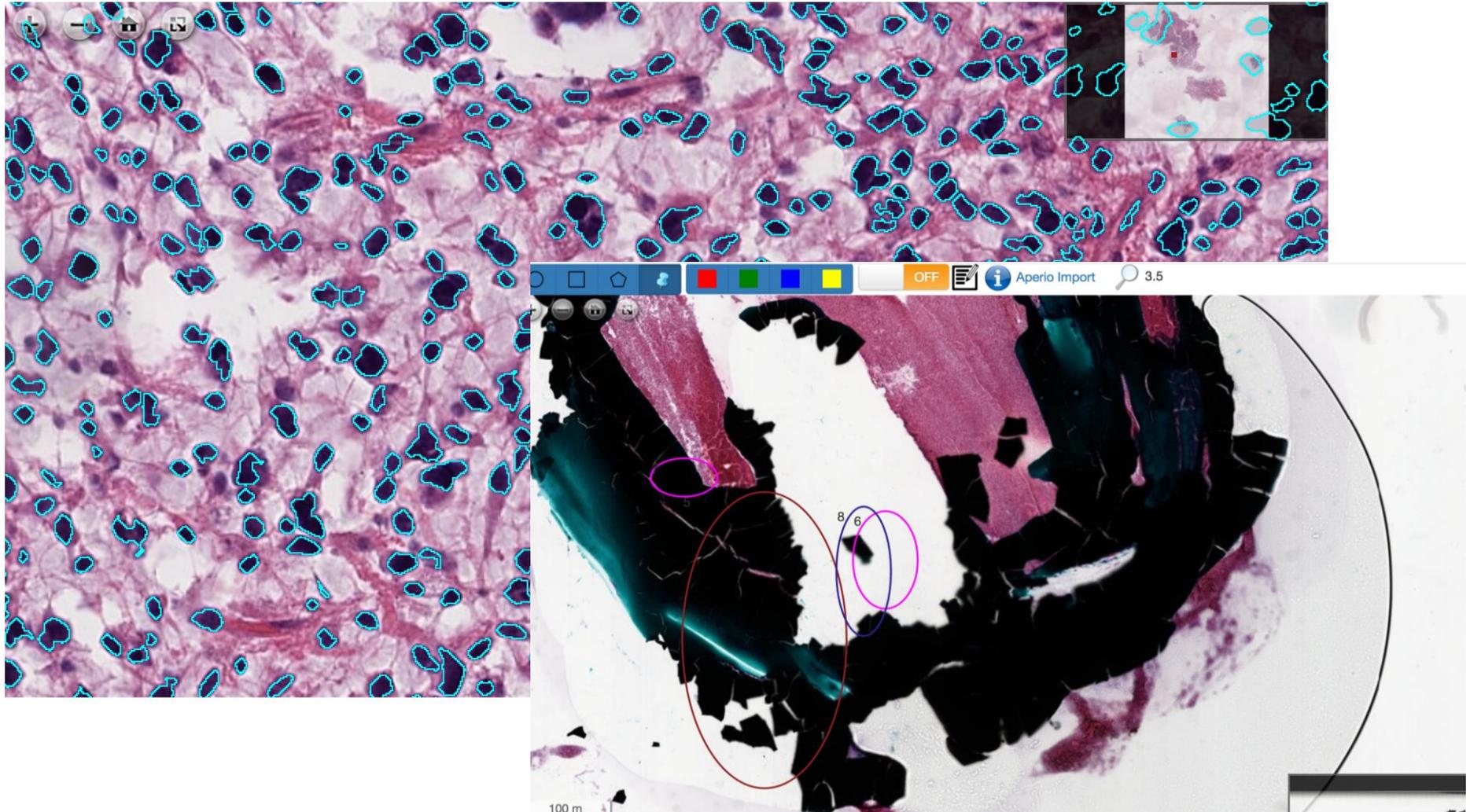
HistomicsTK

TCGA PanCancer Heterogeneity & Evolution (Lazar, Getz)

TCGA Sarcoma AWG (Lazar)

Digital Slide Archive

Visualize Human and Algorithm Generated Results

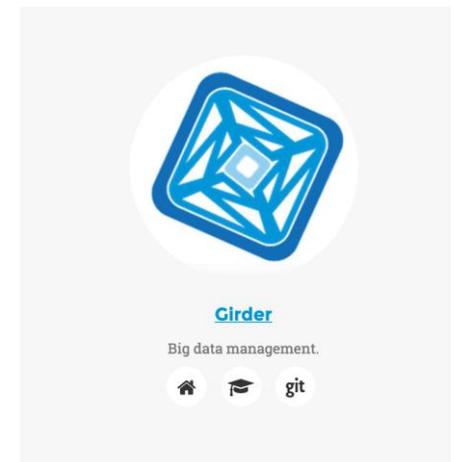


Core Technologies

- Girder (Kitware Platform) for user management/permissions
- Python Based Backend
- MongoDB for Primary Database
- OpenSeadragon Image Viewer
- **Evaluating Docker, CWL and Workflow Tools



mongoDB



(Not) Dealing with PHI

CANCER Digital Slide Archive WINSHIP Digital Slide Viewer Home TCGA Resources Help

Apply Filters Report Bad Image Show Debug Info Draw Tools Query Metadata Query PDF

act Patient
igitalBioBank Available Slides: 6
arch:

Image

2199

EUH_232_TM

33795_TUM

in Status Bar

Document Type: AP Report
Document Date:
Document Status: Auth (Verified)
Document Title: AP REPORT
Encounter info: EUH2199-2

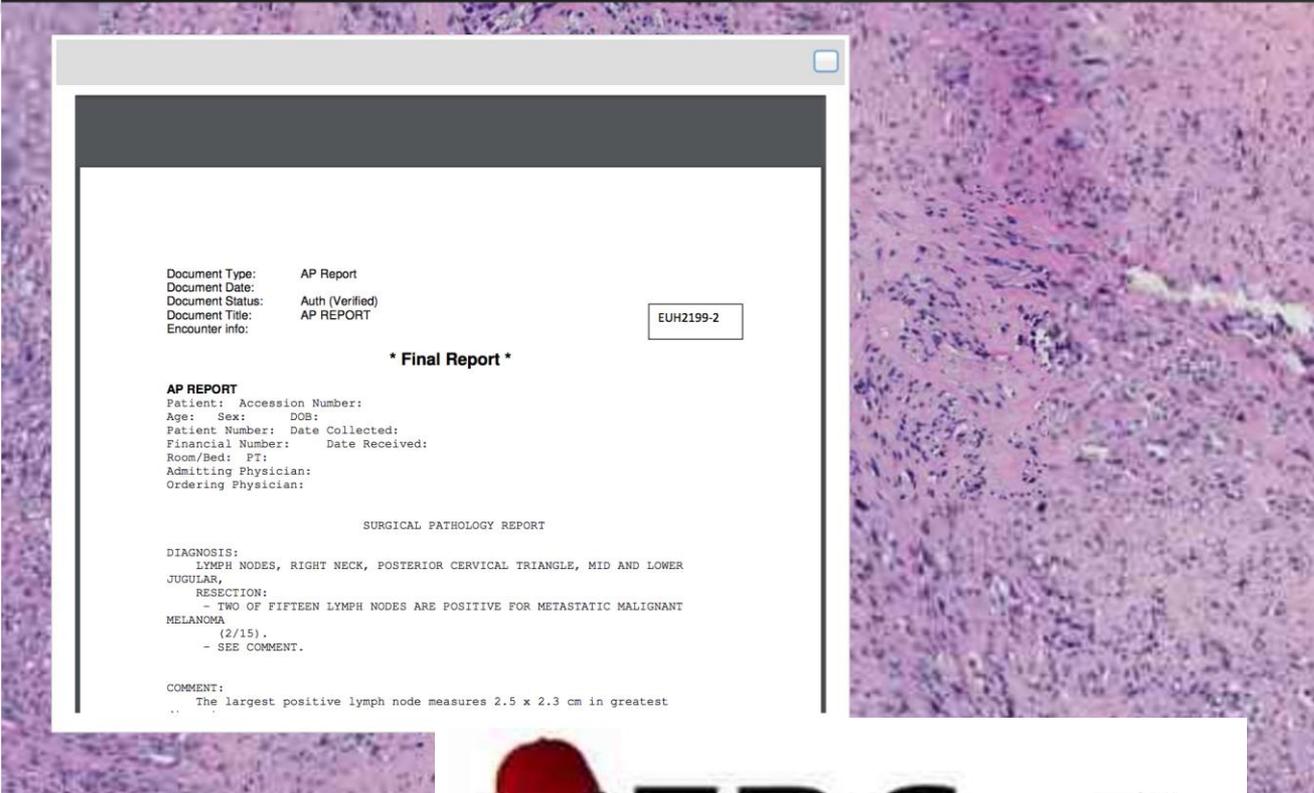
*** Final Report ***

AP REPORT
Patient: Accession Number:
Age: Sex: DOB:
Patient Number: Date Collected:
Financial Number: Date Received:
Room/Bed: PT:
Admitting Physician:
Ordering Physician:

SURGICAL PATHOLOGY REPORT

DIAGNOSIS:
LYMPH NODES, RIGHT NECK, POSTERIOR CERVICAL TRIANGLE, MID AND LOWER JUGULAR,
RESECTION:
- TWO OF FIFTEEN LYMPH NODES ARE POSITIVE FOR METASTATIC MALIGNANT MELANOMA
(2/15).
- SEE COMMENT.

COMMENT:
.. The largest positive lymph node measures 2.5 x 2.3 cm in greatest



??

HistomicsTK

HistomicsTK = Infrastructure + Algorithms for Whole Slide Image Analysis

Infrastructure:

- User Interfaces
- APIs
- Extensibility
- Execution and resource mapping (Girder)
- Provenance and results management (Girder)
- Visualization

HistomicsTK = Infrastructure + Algorithms for Whole Slide Image Analysis

Infrastructure:

- **User Interfaces**
- **APIs**
- **Extensibility**
- Execution and resource mapping (Girder)
- Provenance and results management (Girder)
- Visualization

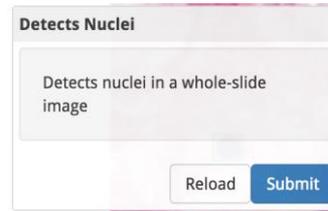


Algorithm Developer

GitHub



DSA UI Widgets



+

Portable algorithm plugin



-REST Interface
-Algorithm
-Dependencies

Step 1. Define Algorithm Interface

Using Slicer Execution Model XML Spec

```
<?xml version="1.0" encoding="UTF-8"?>
<executable>
  <category>HistoricstK</category>
  <title>Performs Adaptive Color Deconvolution</title>
  <description>Uses sparse non-negative matrix factorization to adaptive
  <version>0.1.0</version>
  <documentation-url>https://historicstk.readthedocs.org/en/latest/</doc
  <license>Apache 2.0</license>
  <contributor>Deepak Roy Chittajallu (Kitware)</contributor>
  <acknowledgements>This work is part of the HistoricstK project.</ackno
  <parameters>
    <label>IO</label>
    <description>Input/output parameters.</description>
    <image>
      <name>inputImageFile</name>
      <label>Input Image</label>
```

Step 2. Write Algorithm Source

Python or C/C++

```
import historicstk as htk
import ctk_cli
def run( args ):
    imInput = htk.read_image( args.inputImage )
    ...
if __name__ == "__main__":
    # cmd-line argument parsing and help with ctk_cli
    run( ctk_cli.CLIArumentParser().parse_args() )
```

Step 3. List Algorithms in a JSON

```
{
  "NucleiDetection" : {"type": "python"},
  "CellClassification" : {"type": "c++"}
}
```

Step 4. Write Dockerfile For Containerization

(Generates REST end-points)

```
# Specify root docker image
FROM: dsarchive/historicstk:v0.1.3

# Install system prerequisites
RUN apt-get update && \
    apt-get install -y wget git python
...
# Copy source code into docker image
COPY .
...
# Install dependencies
RUN pip install -r requirements.txt
...
# Use entry-point provided by HistoricstK
ENTRYPOINT ["python", "cli_list_entrypoint.py"]
```



DockerHub

GitHub Auto-build w/ upload to DockerHub

Performs Adaptive Color Deconvolution

Uses sparse non-negative matrix factorization to adaptively deconvolve a given RGB image into intensity images representing distinct stains.

Reload

Submit

Jobs ▾

IO ▲

Input Image

Choose a file...



Initial estimate of color of stain-1

Initial estimate of color of stain-2

Output Image of Stain 1

Choose a file...



Output Image of Stain 2

Choose a file...



Output Image of Stain 3

Choose a file...

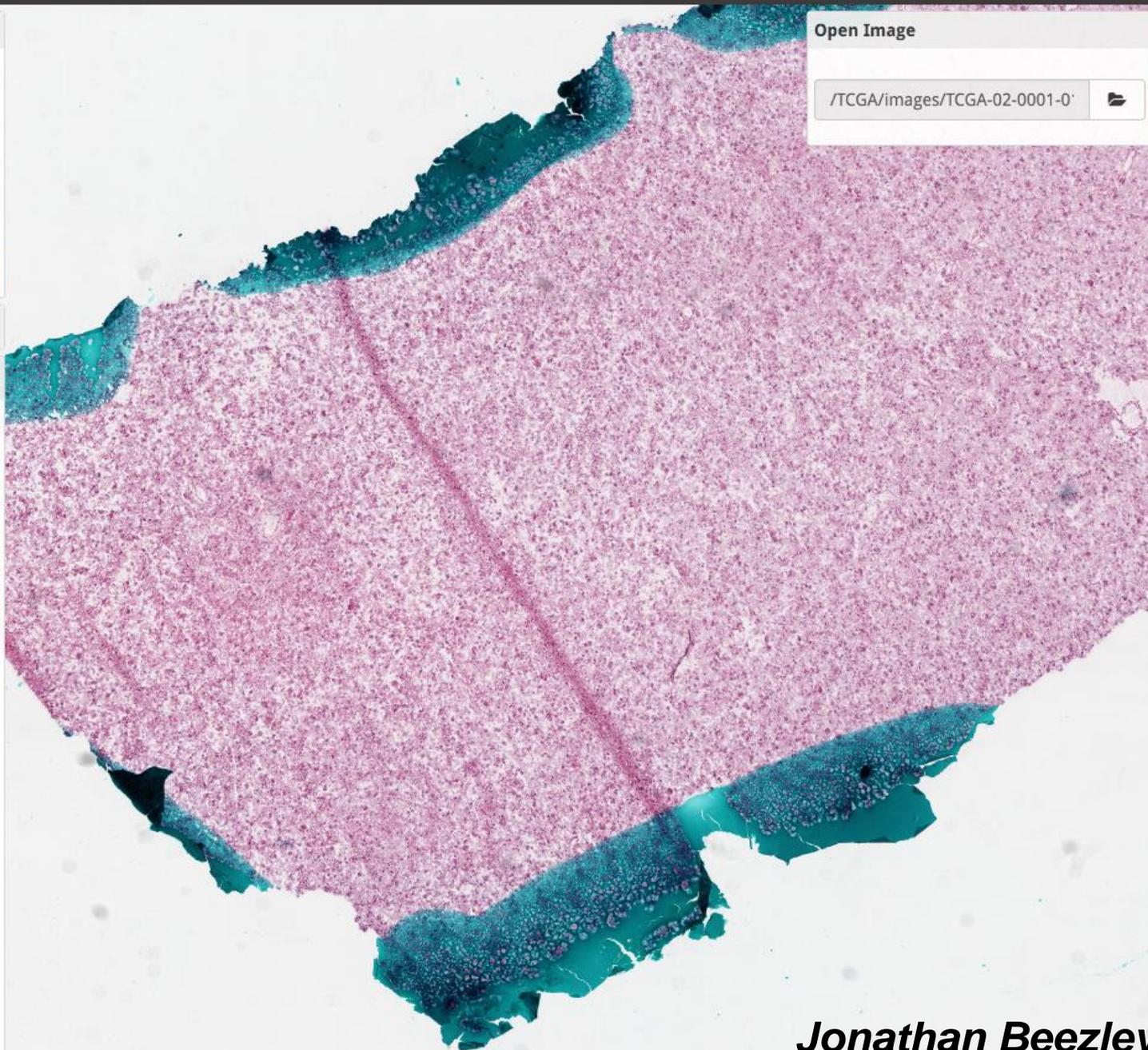


A parameter to control sparsity of stain concentrations

0.5

Open Image

/TCGA/images/TCGA-02-0001-0'



Jonathan Beezley

HistomicsTK Algorithms



Transformations and Color Normalization

ColorConvolution
ColorDeconvolution
ComplementStainMatrix
OpticalDensityFwd
OpticalDensityInv
ReinhardNorm
ReinhardSample
RudermanLABFwd
RudermanLABInv
SparseColorDeconvolution

Feature Extraction

FeatureExtraction

Segmentation

ChanVese
DregEdge
GaussianVoting
GradientFlow
MaxClustering
MergeSinks
SimpleMask

Filtering

Del2
EstimateVariance
GaussianGradient
GradientDiffusion
cLoG
gLoG

Utilities

CondenseLabel
ConvertSchedule
EmbedBounds
FilterLabel
GraphColorSequential
MergeColinear
RegionAdjacencyLayer
Sample
ShuffleLabel
SubmitTorque
TilingSchedule

>[git clone https://github.com/DigitalSlideArchive/HistomicsTK](https://github.com/DigitalSlideArchive/HistomicsTK)

Video

<https://youtu.be/bWv-XrTE5Qc>

Acknowledgement



David Gutman (co-PI)
Michael Nalisnik
Sanghoon Lee
Jun Kong
Pooya Mobadersany
Mohammed Khalilia



Deepak Chittajallu
Jonathan Beezley
Dave Manthey
Brian Helba
Zach Mullen
Charles Law

Clinical Collaborators:

Alexander Lazar
(MD Anderson)
Daniel Brat (Emory)
Christopher Flowers (Emory)
Brian Pollack (Emory)



Funding:
NCI U24CA194362
NLM K22LM011576



Google
Summer of Code