

DIBBs Brown Dog

*A Science Driven Data
Transformation Service*



SMU



UNIVERSITY OF
MARYLAND

Photos Structural Defects Image Stitching
Materials Development Gap Filling Green Infrastructure
Tabular Data Root Tip Tracking Loss of Organ Function
Evolution Underwater Photos Mapping NLP Databases
Video Lidar Hyperspectral Climate Modeling
Historic Maps Color Correction Publications Radar
Flood Plain Analysis Documents River Depth Distribution Hazard Modeling
Cell Tracking Web Sites People Detection/Tracking
Bee Colony Behavior Feedlot Tracking Stream Detection and Sinuosity
Phenomics River Maturity Paleoclimate Bee Detection/Tracking
Pollen Detection/Classification Human Preference Modeling Tissue Classification
Land Cover/Usage Regions in Conflict Satellite/Aerial Photos Disease Detection
Coastline Changes Water Detection (e.g. Lakes, Retaining Ponds) Species Detection/Counting Reef Changes
Large Dynamic Group Behavior River Meander Sentiment Analysis Food Supply
Handwritten Documents Pre-Digital Datasets 3D Data Renal Failure
3D Reconstruction Event Detection Simulations

New Capabilities

- Facilitate, simplify, and make more reproducible the digital **data wrangling** of modern scientific research activities

Data Wrangling

Extractions: Data transformations that create new, often higher level, data from the contents of data (e.g. metadata and other derived products). Typically not reversible. Towards indexing, searching, cleaning, further analyzing the data.



Conversions: Data transformations that largely preserve the entirety of the data (i.e. potentially reversible). Towards accessing, loading, or viewing the data contents.



Conversion



Extraction

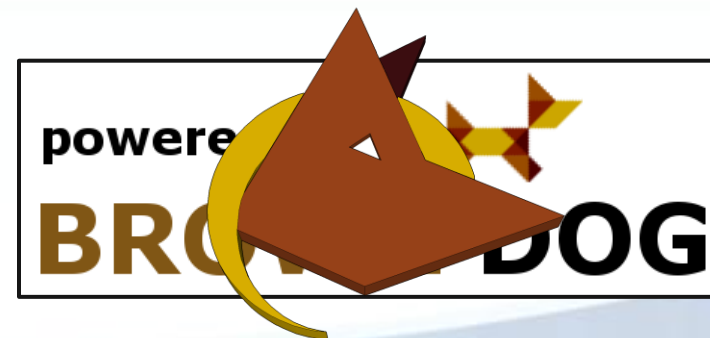
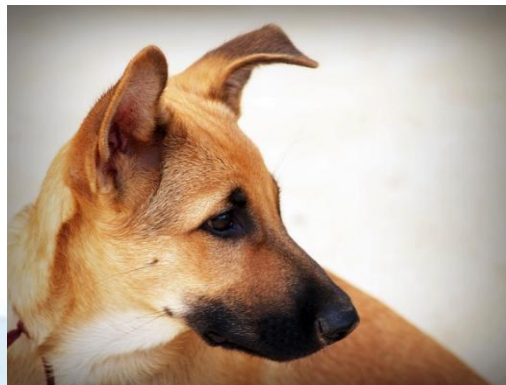
```
{
  "extractor_id": "ncsa.image.exif",
  "Image": "558c3d84e4b00c3a039d5ac5",
  "Format": "JPEG (Joint Photographic Experts Group JFIF format)",
  "Class": "DirectClass",
  "Geometry": "2592x1936+0+0",
  "Resolution": "72x72",
  "Print size": "36x26.8889",
  "Units": "PixelsPerInch",
  "Type": "TrueColor",
  "Endianness": "Undefined",
  "Colorspace": "sRGB",
  "Depth": "8-bit",
  "Channel depth": {
    "red": "8-bit",
    "green": "8-bit",
    "blue": "8-bit"
  },
  "Channel statistics": {
    "Red": {
      "min": "0 (0)",
```

```
{
  "extractor_id": "ncsa.image.ocr",
  "ocr_simple": [
    "EB BROWSER MOSAIC THE FIRST POPULAR BROWSER FOR THE WORLD WIDE  
BY MARC ANDREESSEN BINA THE NATIONAL CENTER COMPUTING APPLICATIONS  
1993 RELEASE TO THE PUBLIC INTERNET USERS EASY ACCESS TO SOURCES OF  
INFORMATION win HAVE TRANSFORMED THE INFORMATION UNIVERSITY OF "
```

```
{
  "id": "558c3d84e4b00c3a039d5ac5",
  "filename": "IMG_0997.JPG",
  "tags": [
    "Human Face Automatically Detected",
    "Person Automatically Detected",
    "Human Eyes Automatically Detected"
  ]
}
```

Brown Dog - A Science Driven Data Transformation Service

- Extensibility
 - Easy to add new transformations (i.e. converters and extractors)
 - Encapsulated transformation software & dependencies
- API
 - Supporting other applications/frameworks to build on top of
- Support for diverse usage (i.e. clients, languages, community tools & applications)
- Scalability, Distributed, Data Movement, Provenance with File Validation & Information Loss, Tool Preservation & Publication, **Open Source**



<https://en.wikipedia.org/wiki/Mongrel>

https://bd-api.ncsa.illinois.edu

 swagger

Authorize

Brown Dog API

Documentation for the [Brown Dog API](#)

Created by Brown Dog Support

See more at <https://opensource.ncsa.illinois.edu/confluence/questions/topics/41156618/brown-dog>

[Contact the developer](#)

[University of Illinois/NCSA Open Source License](#)

Authorization

Show/Hide | List Operations | Expand Operations

POST	/keys	Create a new api key
POST	/keys/{key}/tokens	Create a new access token from api key
DELETE	/keys/{key}	Delete api key and all related access tokens
DELETE	/tokens/{token}	Delete token
GET	/tokens/{token}	

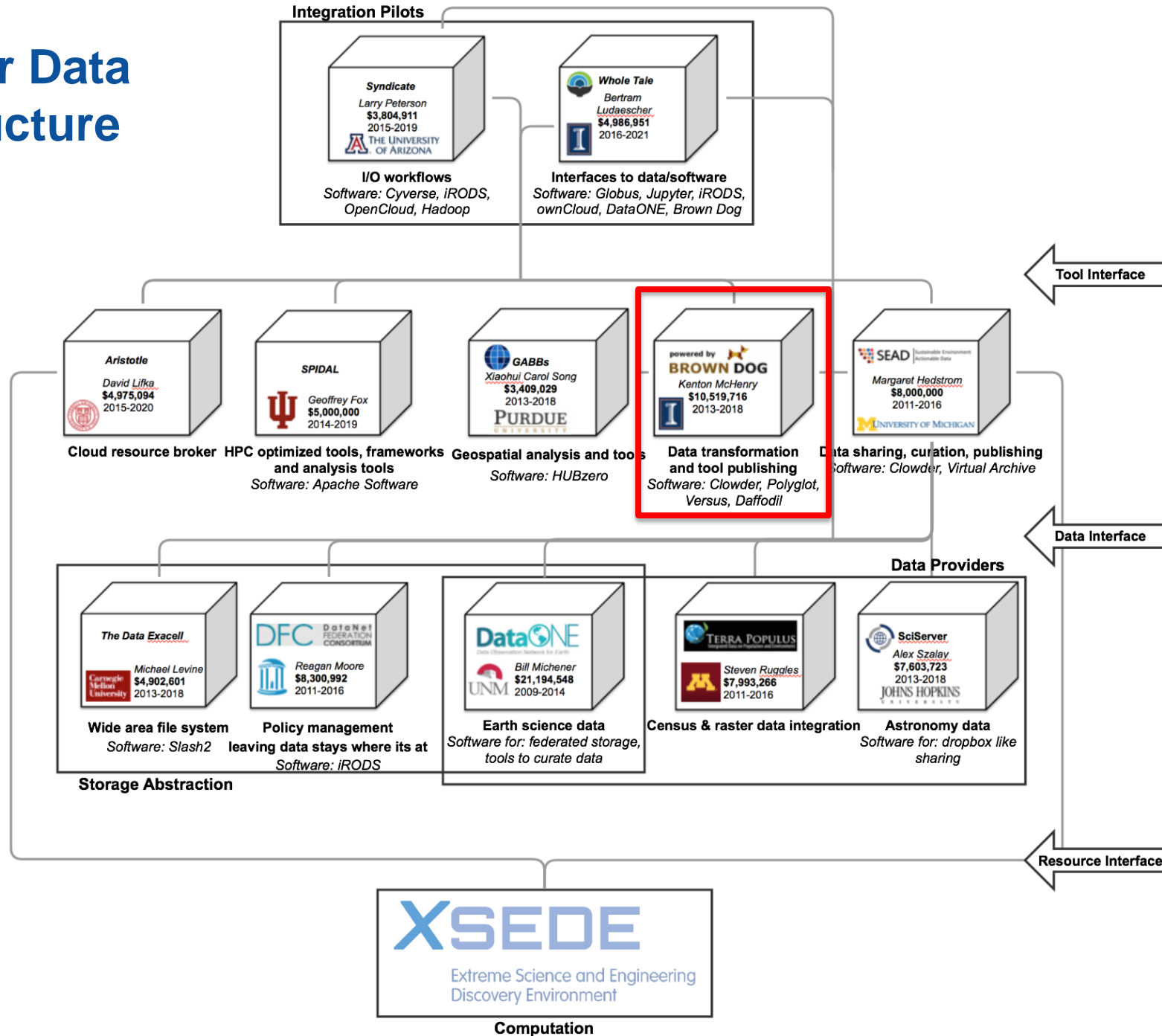
Conversions

Show/Hide | List Operat

GET	/dap/outputs	List all output formats that can be reached
-----	--------------	---

powered by 
BROWN DOG

Enhance Other Data Cyberinfrastructure Components



Enhance your Data
Cyberinfrastructure
Components

re3data.org
REGISTRY OF RESEARCH DATA REPOSITORIES

EZID
Long-term identifiers made easy

AXES

Fedora™

OpenAIRE

zenodo

HYDROSHARE

Meta
Cat



Web
Lab

SciDrive

Rappture



Tool Interface



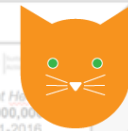
ownCloud

BisQue



TwoRavens

OpenCloud



ckan

SLASH2



EARTH CUBE

swift

Software: HUBzero

Agave

Data sharing, curation, publishing
and tools

face

DataONE

DRYAD



figshare



pegasus

Girder



RDA

RODS

RODS

CYVERSE™

yt



Jupyter

Storage Abstraction

Policy man
leaving data stay

RESEARCH DATA ALLIANCE

Galaxy

globus

The
Dataverse
Project



hubzero

PLATFORM
FOR SCIENTIFIC
COLLABORATION

dSPACE



SciServer

BitCurator

DFC
DataNet
FEDERATION
CONSORTIUM

SEAD

Status & Dissemination



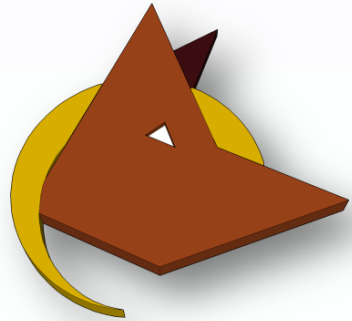
CZO | **CRITICAL ZONE OBSERVATORIES**
U.S. NSF NATIONAL PROGRAM



WEST BIG DATA INNOVATION HUB

Midwest Big Data Hub

Accelerating the Big Data Innovation Ecosystem



ILLINOIS DATA BANK

SOUTH



BDHUB



NORTHEAST BIG DATA INNOVATION HUB



DIBBs / DataNets



XSEDE

Extreme Science and Engineering
Discovery Environment

Research Data Services



EARTH CUBE





SEAD

Sustainable Environment
Actionable Data



Brown Dog Dev Collections Datasets Files Tags API Login Administration Search

Created by **Anonymous User**
Created on Apr 08, 2015

All Rights Reserved Anonymous User

Created by DW WF 2015-04-08T11:39:58

Geospatial Layers

Geospatial Layers

- overlay.tif
- overlay.tif
- dap-streamclip....
- dap-demclip.tif
- overlay.tif
- overlay.tif
- floodplain.zip
- Basemap:

File Types

- image/tiff
- application/zip

Collections containing the dataset

Overlay of floodplains extracted from lidar data and historic river locations from digitized maps



Predictive Ecosystem Analyzer (PEcAn)



Selected Site
Set parameters for the run.

PFT*
populus
temperate.coniferous
temperate.deciduous

Start Date*
2004/01/01

End Date*
2004/12/31

Sipnet.climna*
Use Ameriflux

Email

Use **BrownDog**

Edit pecan.xml

Edit model config

Advanced setup

Missouri Ozark Site/BREA (US-MOz)
Ashland, MO, US

powered by **BROWN DOG**

The PEcAn project is supported by the National Science Foundation (ABI #1062547, ARC #1023477) and the Energy Biosciences

Dataset / model conversions



Novel (On the Fly) Data Products

```
1 library(shiny)
2 library(leaflet)
3
4 # Define UI
5 ui <- shinyUI(fluidPage(
6   # Application title
7   titlePanel("Generate xml file"),
8
9   sidebarLayout(
10    sidebarPanel(
11      selectInput("type", "Type", c("AmeriFlux", "FLUXNET", "NARR")),
12      uiOutput("agreementUI"),
13      textInput("start_date", "Start year", placeholder="2001"),
14      textInput("end_date", "End year", placeholder="2001"),
15      textInput("token", "BrownDog Token"),
16      conditionalPanel(
17        condition = "input.token != ''",
18        selectInput("model", "Model", c("ameriflux.zip", "clim", "dalec")),
19      )
20    ),
21    mainPanel(
22      fluidRow(
23        leafletMap(
24          "map", "100%", 400,
25          initialTileLayer = "://{s}.tiles.mapbox.com/v3/jcheng.map-5e",
26          initialTileLayerAttribution = HTML('Maps by <a href="http://mapbox.com">Mapbox</a>'),
27          options=list(
28            center = c(37.45, -93.85),
29            zoom = 4,
30            maxBounds = list(list(17, -180), list(59, 180))
31          ),
32          style='padding:10px;'
33        ),
34        verbatimTextOutput("xmltext"),
35        fluidRow(
36          column(3,
37            conditionalPanel(
38              condition = "input.agreement == true",
39              downloadButton('downloadXML', 'Download XML')
40            ),
41          column(9,
42            conditionalPanel(
43              condition = "input.token != '' && input.agreement == true",
44              downloadButton("downloadData", "Download Data")
45            )
46          )
47        )
48      )
49    )
50  )
51  )
```

Generate xml file

Type: AmeriFlux

I agree to AmeriFlux license.

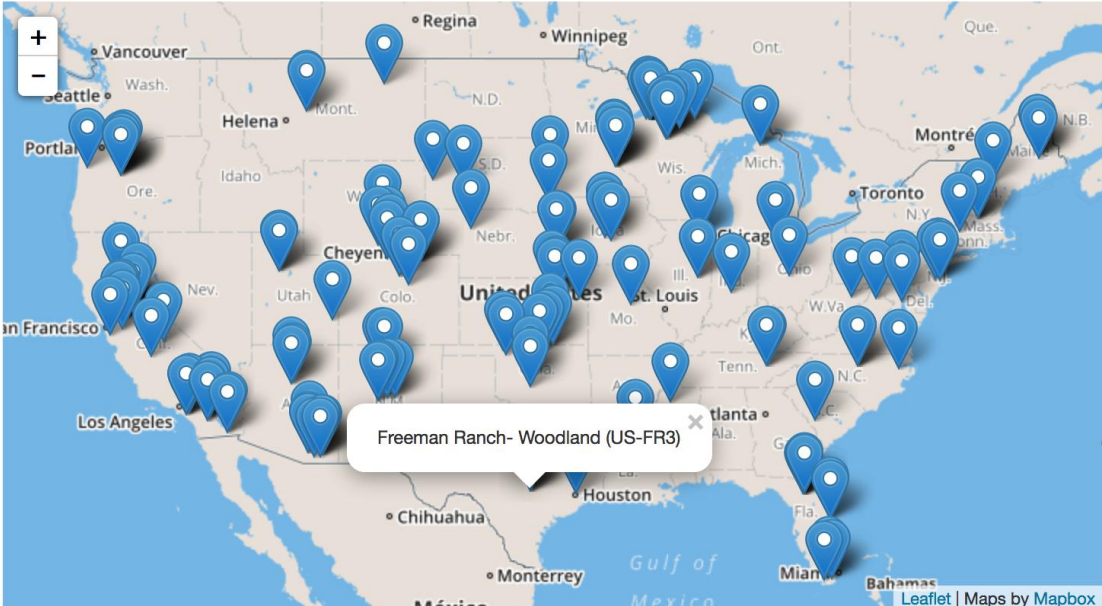
Start year: 2001

End year: 2001

BrownDog Token: c8ac077a-0cd8-4486-8be6-d2bb70c14abb

Model: ed.zip

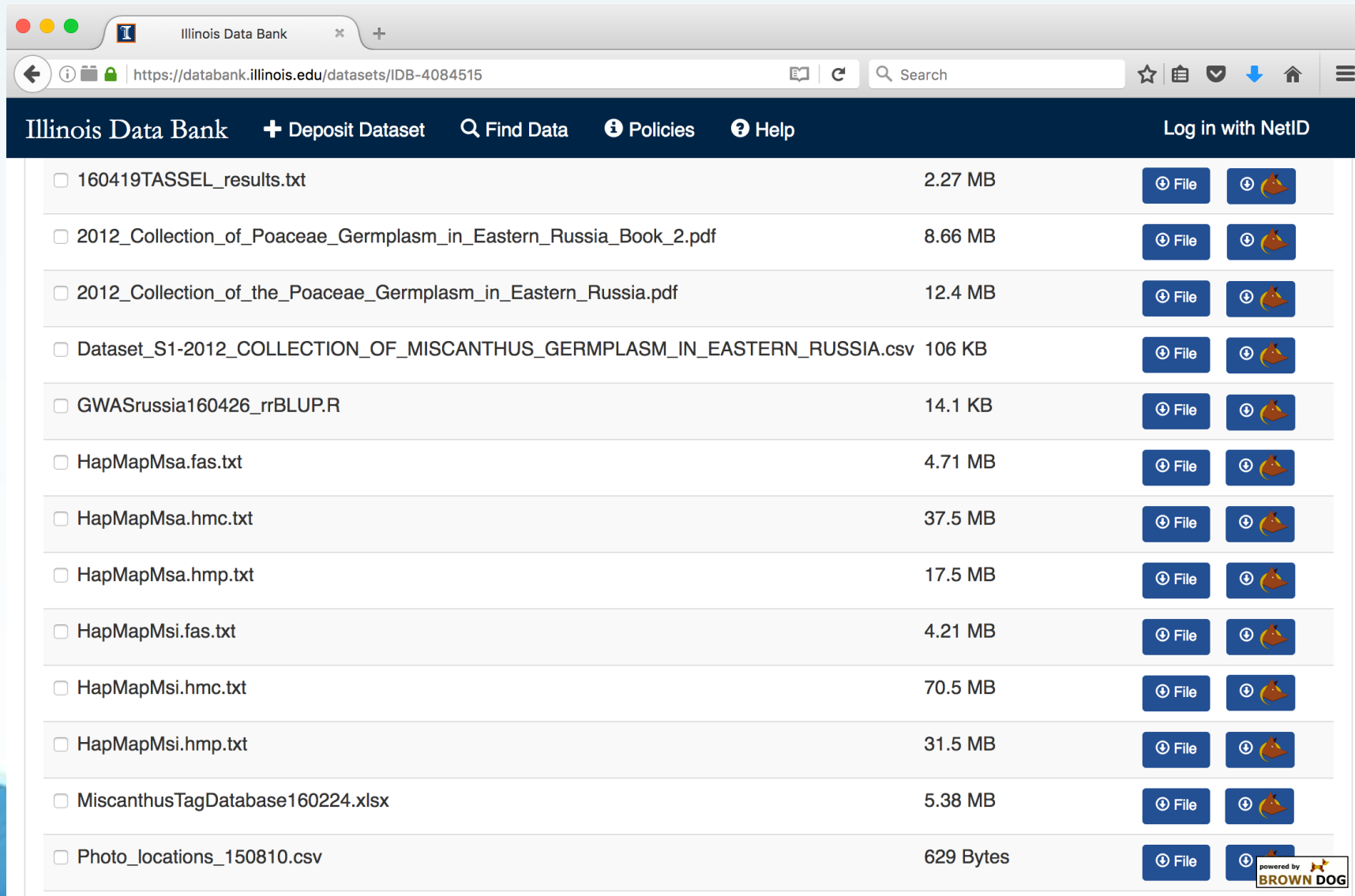
- ameriflux.zip
- clim
- dalec
- ed.zip
- linkages
- pecan.nc
- pecan.zip



```
<input>
  <type>AmeriFlux</type>
  <site>Freeman Ranch- Woodland (US-FR3)</site>
  <lat>29.94</lat>
  <lon>-97.99</lon>
  <start_date>2001-01-01 00:00:00</start_date>
  <end_date>2001-12-31 23:59:59</end_date>
</input>
```

Download XML Download Data

Research Data Services



The screenshot shows a web browser window with the URL <https://databank.illinois.edu/datasets/IDB-4084515>. The page header includes the Illinois Data Bank logo and navigation links: Deposit Dataset, Find Data, Policies, and Help. A "Log in with NetID" button is also present. The main content is a table of datasets, each with a checkbox, filename, size, and two action buttons: "File" and a download icon.

File Name	Size	File	Download
<input type="checkbox"/> 160419TASSEL_results.txt	2.27 MB	File	Download
<input type="checkbox"/> 2012_Collection_of_Poaceae_Germplasm_in_Eastern_Russia_Book_2.pdf	8.66 MB	File	Download
<input type="checkbox"/> 2012_Collection_of_the_Poaceae_Germplasm_in_Eastern_Russia.pdf	12.4 MB	File	Download
<input type="checkbox"/> Dataset_S1-2012_COLLECTION_OF_MISCANTHUS_GERMPLOSM_IN_EASTERN_RUSSIA.csv	106 KB	File	Download
<input type="checkbox"/> GWASrussia160426_rrBLUP.R	14.1 KB	File	Download
<input type="checkbox"/> HapMapMsa.fas.txt	4.71 MB	File	Download
<input type="checkbox"/> HapMapMsa.hmc.txt	37.5 MB	File	Download
<input type="checkbox"/> HapMapMsa.hmp.txt	17.5 MB	File	Download
<input type="checkbox"/> HapMapMsi.fas.txt	4.21 MB	File	Download
<input type="checkbox"/> HapMapMsi.hmc.txt	70.5 MB	File	Download
<input type="checkbox"/> HapMapMsi.hmp.txt	31.5 MB	File	Download
<input type="checkbox"/> MiscanthusTagDatabase160224.xlsx	5.38 MB	File	Download
<input type="checkbox"/> Photo_locations_150810.csv	629 Bytes	File	Download

Show 25 per page

Show / hide columns Copy CSV Excel Print

Search

Search

Clark, Lindsay V.; Dzyubenko, Elena; Dzyubenko, Nikolay; Bagmet, Larisa; Sabitov, Andrey; Chebukin, Pavel; Johnson, Douglas A.; Kjeldsen, Jens Bonderup; Petersen, Karen Koefoed; Jørgensen, Uffe; Yoo, Ji Hye; Heo, Kweon; Yu, Chang Yeon; Zhao, Hua; Jin, Xiaoli; Peng, Junhua; Yamada, Toshihiko; Sacks, Erik J. (2016): Data from: "Ecological characteristics and in situ genetic associations for yield-component traits of wild *Miscanthus* from eastern Russia". University of Illinois at Urbana-Champaign. https://doi.org/10.13012/B2IDB-4084515_V1

Datasets and R scripts relating to the manuscript "Ecological characteristics and in situ genetic associations for yield-component traits of wild *Miscanthus* from eastern Russia" published in *Annals of Botany*, 10.1093/aob/mcw137. [...]

Keywords: *Miscanthus sacchariflorus*; *Miscanthus sinensis*; Russia; germplasm; RAD-seq; SNP

Donovan, Brian; Mori, Alec; Agrawal, Nimit; Meng, Yalan; Lee, Jong; Work, Daniel (2016): New York City Hourly Traffic Estimates (2010-2013). University of Illinois at Urbana-Champaign. https://doi.org/10.13012/B2IDB-4900670_V1

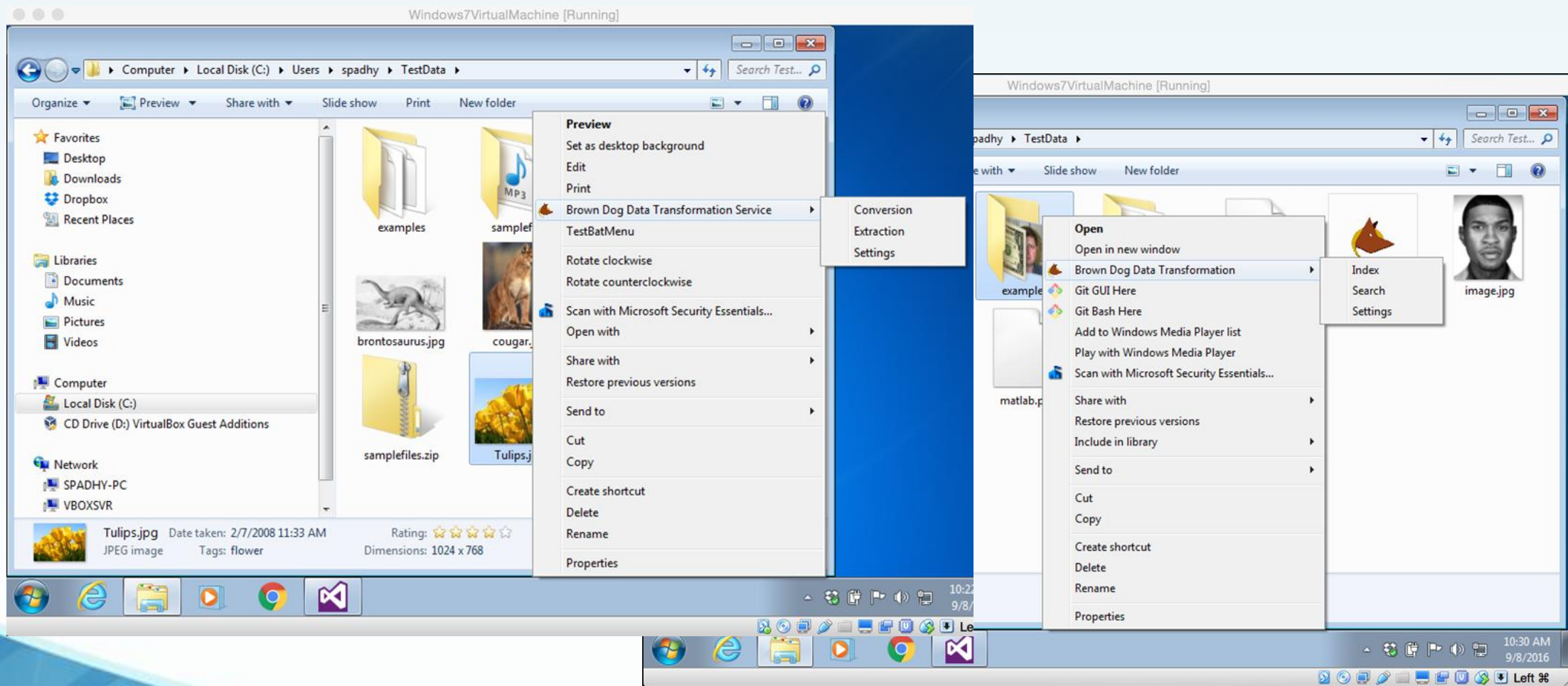
This dataset contains hourly traffic estimates (speeds) for individual links of the New York City road network for the years 2010-2013, estimated from New York City Taxis.

Keywords: traffic estimates; traffic conditions; New York City

Donovan, Brian; Work, Dan (2016): New York City Taxi Trip Data (2010-2013). University of Illinois at Urbana-Champaign. <https://doi.org/10.13012/J8PN93H8>

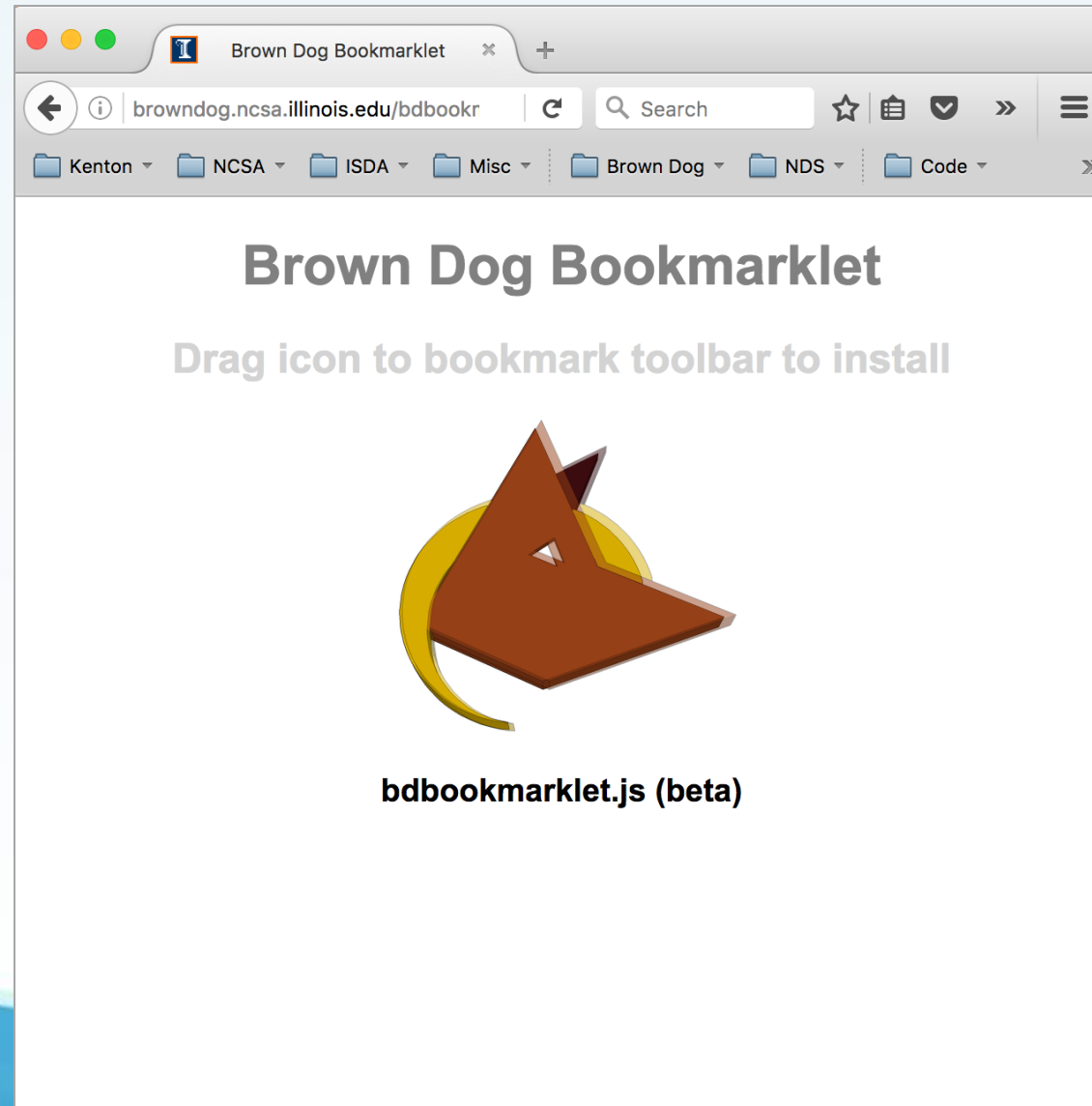
This dataset contains records of four years of taxi operations in New York City and includes 697,622,444 trips. Each trip records the pickup and drop-off dates, times, and coordinates, as well as the metered distance reported by t[...]

Windows & Command Line Clients



```
kmchenry@kgm-d3:~/git/bdcli$ bd -o png examples/photo_cd/IMG0001.PCD | bd -v
```


BD Bookmarklet



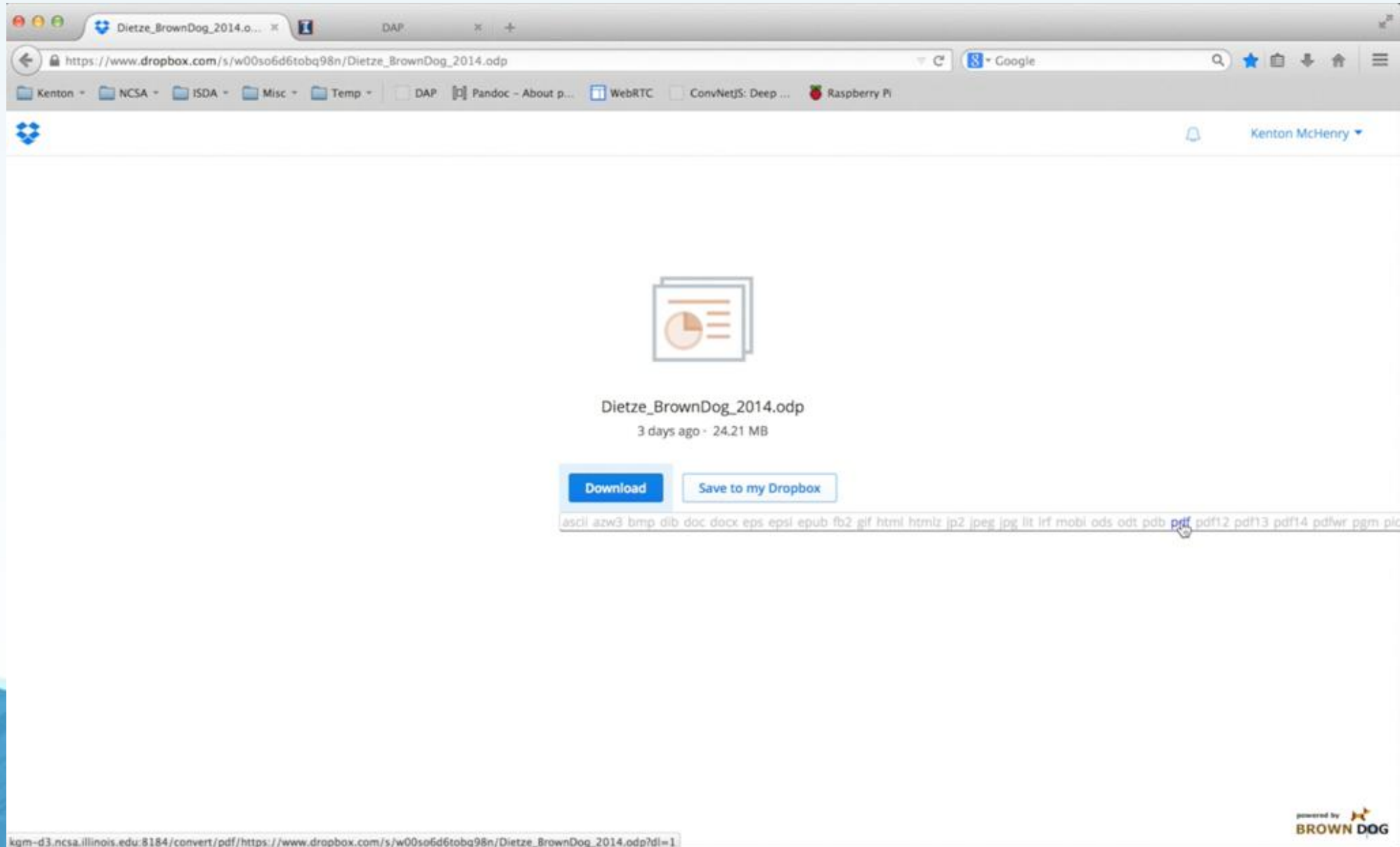
BD Bookmarklet

	<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
	Parent Directory		-	
	0005.jp2	2014-07-03 13:49	633K	
	0094.jp2	2014-07-03 15:15	7.6M	
	105_1.sid	2014-07-03 13:47	3.7M	
	aac aif aiff ascii asf avi azw3 bmp clim dalec dib doc docx ed.zip eps epsi epub			
	City Limits.7z	2014-10-01 12:43	763	
	Human_knownGene_chr22_1-51304566.bed	2015-04-14 13:57	260K	
	IMG_0997.jpg	2016-07-03 23:22	2.1M	
	US-Dk3-2001-2003.xml	2015-07-08 22:31	191	
	alice.txt	2014-12-27 23:06	141K	

Conversion menus added to links



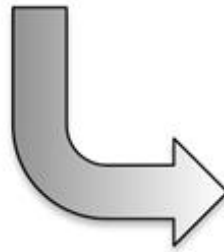
BD Bookmarklet



The screenshot shows a web browser window with the address bar displaying the URL: `https://www.dropbox.com/s/w00so6d6tobq98n/Dietze_BrownDog_2014.odp`. The browser's address bar also shows a search engine (Google) and several open tabs, including 'DAP', 'Pandoc - About p...', 'WebRTC', 'ConvNetJS: Deep ...', and 'Raspberry Pi'. The Dropbox interface shows the file 'Dietze_BrownDog_2014.odp' with a file icon, the name, and the date '3 days ago' and size '24.21 MB'. Below the file information are two buttons: 'Download' and 'Save to my Dropbox'. At the bottom of the page, there is a list of file format options: `ascii azw3 bmp dib doc docx eps epsi epub fb2 gif html htmlz jp2 jpeg jpg lit lrf mobi ods odt pdb pdf pdf12 pdf13 pdf14 pdfwr pgm pic`. The 'pdf' option is highlighted with a mouse cursor. In the bottom right corner, there is a logo for 'powered by BROWN DOG' and a small icon of a dog. The bottom left corner of the browser window shows the URL: `kgm-d3.ncsa.illinois.edu:8184/convert/pdf/https://www.dropbox.com/s/w00so6d6tobq98n/Dietze_BrownDog_2014.odp?dl=1`.

BD Bookmarklet


Press Ctrl+F to index and search data on page




Search Data ✕


Queryable Data Create Index 0 0 0

Search query Search

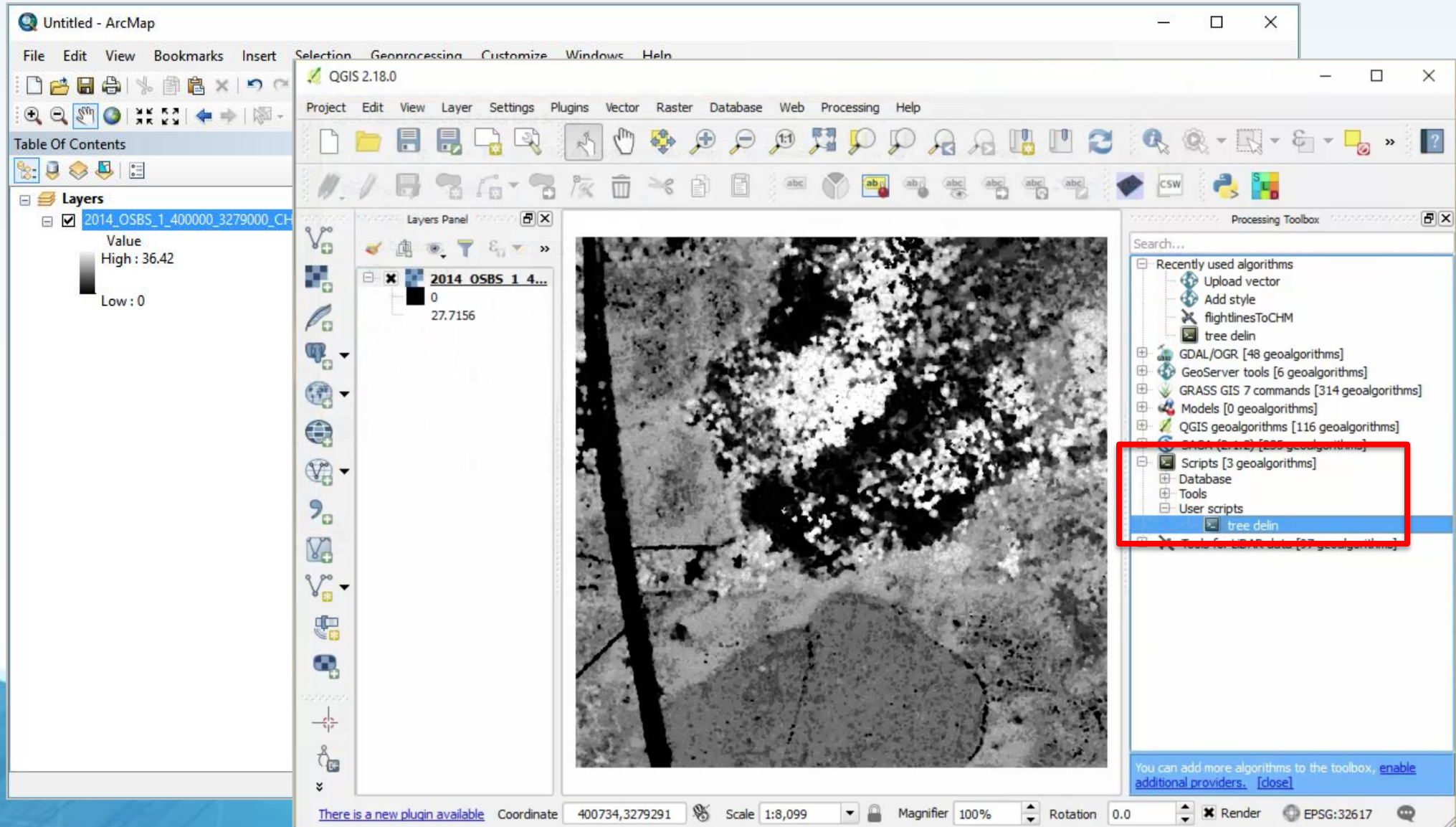
 **Image size**
2592 x 1936

 **Image size**
480 x 640

<http://brandon.sas.uillinois.edu/examples/brandon.sas> 1 link

 Close

Community Software



Community Software

The screenshot displays a Google Sheet titled 'example-path' with a red box highlighting a data table. The table has the following structure:

Route	Lat	Long	Green Index	Orientation
1	40.11123	-88.17377	30.8	0
1	40.11123	-88.17377	37.7	90
1	40.11123	-88.17377	38.9	180
1	40.11123	-88.17377	27.6	270
1	40.11202	-88.17488	32.1	270
1	40.11202	-88.17488	20.9	180
1	40.11202	-88.17488	32.2	0
1	40.11202	-88.17488	47.1	90
1	40.11289	-88.17517	31.6	0
1	40.11289	-88.17517	16.9	90
1	40.11289	-88.17517	19.9	270
1	40.11289	-88.17517	24.1	180

To the right of the sheet is a 'Brown Dog' panel titled 'Green Route Index'. It contains the following instructions:

1. Provide an Access Token:
2. Select cells in the sheet as a 2 column list of latitude, longitude pairs along a path.
3. Submit to Brown Dog service. A new sheet will be added to the workbook with the results once they are ready.

Below the instructions is a 'Submit Selection' button. The status below the button reads: 'Submitted 6 rows and 2 columns' and 'Adding metadata to sheet'. A link for 'Original File Metadata' is provided, followed by a list of 12 derived file metadata links:

- [Derived File 0 metadata](#)
- [Derived File 1 metadata](#)
- [Derived File 2 metadata](#)
- [Derived File 3 metadata](#)
- [Derived File 4 metadata](#)
- [Derived File 5 metadata](#)
- [Derived File 6 metadata](#)
- [Derived File 7 metadata](#)
- [Derived File 8 metadata](#)
- [Derived File 9 metadata](#)
- [Derived File 10 metadata](#)
- [Derived File 11 metadata](#)


BD Fiddle

Firefox File Edit View History Bookmarks Tools Window Help

BDFiddle

browndog.ncsa.illinois.edu/bdfiddle/


BDFiddle Conversions API Extractions API

 **Matlab**
bd.m

```
bd_funcs = bd_client;

metadata = bd_funcs.extract('https://bd-api.ncsa.illinois.edu/dts/api/extractions/upload_file', 'http://browndog.ncsa.illinois.edu/examples/IMG_0997.jpg', '8653dfc5-450b-4433-9bb9-def8fcaec9c6');
```

Copy To Clipboard Download Code

 **JavaScript**
bd.js

```
<html>
<head>
<script src="https://opensource.ncsa.illinois.edu/bitbucket/projects/BD/repos/bd.js/browse/bd.js?raw"></script>
</head>
<body>

<a href="http://browndog.ncsa.illinois.edu/examples/IMG_0997.jpg" dts="">File</a><br>

</body>
</html>
```

Copy To Clipboard Download Code

NCSA I BOSTON UNIVERSITY UNIVERSITY OF MARYLAND SMU

Lines of Code	273
Other files	Model Image Sample
Dependencies	numpy, argparse, glob, cv2, cPickle, random, h5py, skimage, sklearn, scipy
Difficulties	Install OpenCV (cv2)



```

import argparse
import glob
import cv2
import pickle

exposure

and parse the arguments

", required=True,

"size",

help="Flag indicating whether or not to visualize each iteration")
ap.add_argument("-p", "--path",
                help="path to gi_detector")
args = vars(ap.parse_args())

path_1 = '/Users/ankitrai/Dropbox/ppao_VM/gi_detector/'
def pyramid(image, scale=1.5, minSize=(55,55)):
    # yield the original image
    yield image

    # keep looping over the pyramid
    while True:
        # compute the new dimensions of the image and resize it
        w = int(image.shape[1] / scale)
        image = imutils.resize(image, width=w)

        # if the resized image does not meet the supplied minimum
        # size, then stop constructing the pyramid
        if image.shape[0] < minSize[1] or image.shape[1] < minSize[0]:
            break

        # yield the next image in the pyramid
        yield image
def sliding_window(image, stepSize, windowSize):
    # slide a window across the image
    for y in xrange(0, image.shape[0], stepSize):
        for x in xrange(0, image.shape[1], stepSize):
            # yield the current window
            yield (x, y, image[y:y + windowSize[1], x:x + windowSize[0]])
def detect(image, winDim, winStep=4, pyramidScale=1.5, minPr

    # initialize the list of bounding boxes and associat

    # loop over the image pyramid
    pyramid_layers = pyramid(image, scale=1.5, minSize=(100,100))
    for layer in pyramid_layers:
        # determine the current scale of the pyramid
        scale = image.shape[0] / float(layer.shape[0])
        # loop over the sliding windows for the current pyramid layer
        for (x, y, window) in sliding_window(layer, winStep, winDim):
            (winH, winW) = window.shape[:2]
            if winH == winDim[1] and winW == winDim[0]:
                # extract HOG features from the current window and classify

```

Total Code from 2 Files

Lines of Code	47
Other files	None
Dependencies	bd, requests, os, glob, argparse, time, json, PIL
Difficulties	

```

# import the necessary packages
import requests
import os
import glob
import argparse
import time
import json
import PIL
from PIL import Image, ImageDraw
from bd import extract

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-t", "--token", required=True, help="API token")
ap.add_argument("-i", "--images", help="Path to the image directory")
ap.add_argument("-b", "--bdapi", help="Brown Database API")
args = vars(ap.parse_args())

# parse the arguments
token = args["token"]
if args["images"] != None:
    image_path = args["images"]
else:
    image_path = "./images"
if args["bdapi"] != None:
    bdapi = args["bdapi"]
    if bdapi[-1] != "/":
        bdapi = bdapi + "/"
else:
    bdapi = "https://bd-api.ncsa.illinois.edu/"

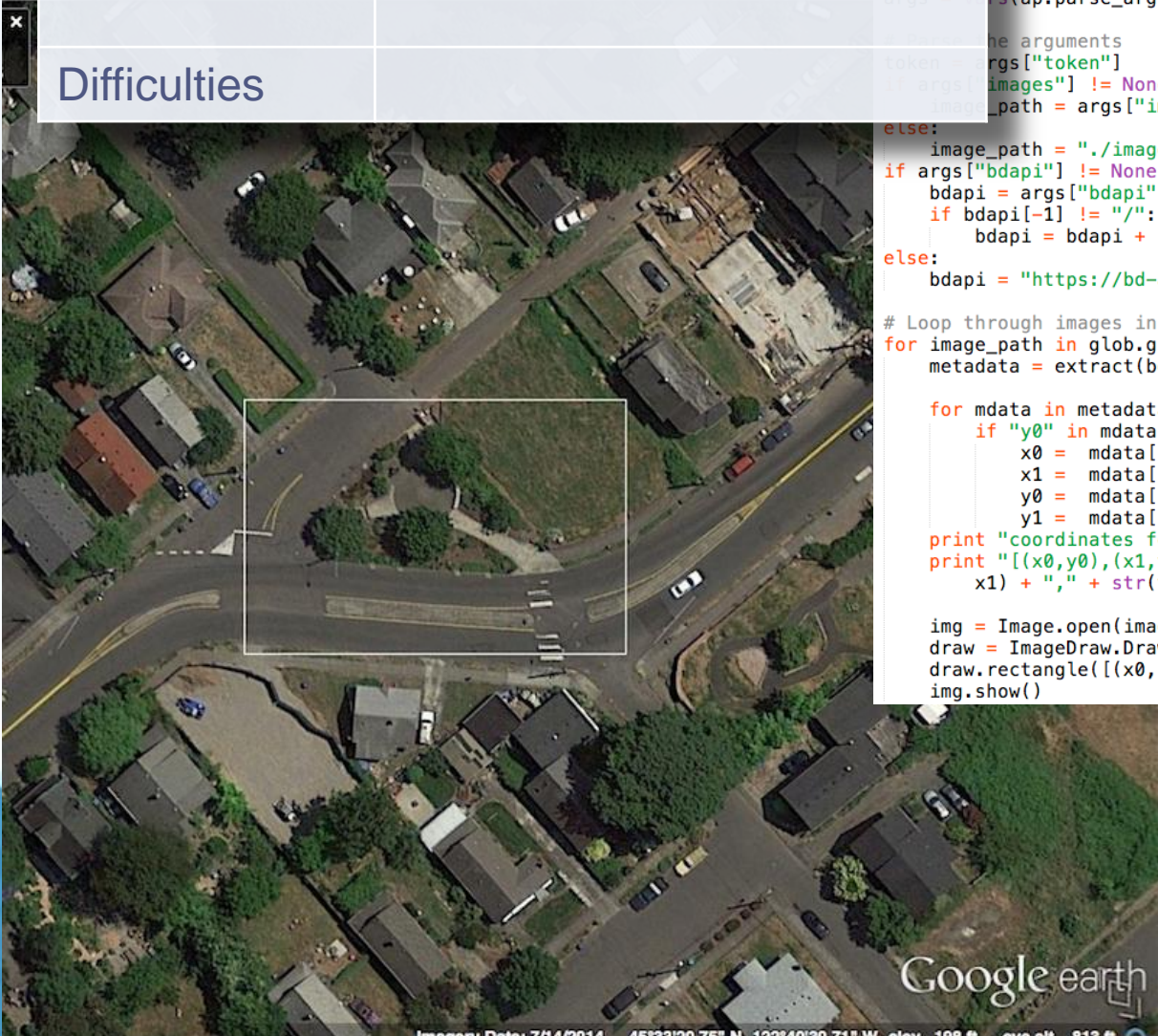
# Loop through images in image directory
for image_path in glob.glob(image_path + "/*.jpg"):
    metadata = extract(bdapi, image_path, token)['metadata.jsonld']

    for mdata in metadata:
        if "y0" in mdata["content"]:
            x0 = mdata["content"]["x0"]
            x1 = mdata["content"]["x1"]
            y0 = mdata["content"]["y0"]
            y1 = mdata["content"]["y1"]
            print "coordinates for bioswale bounding box in" + image_path
            print "[(x0,y0),(x1,y1)] = [{" + str(x0) + "," + str(y0) + "}, {" + str(x1) + "," + str(y1) + "}]

    img = Image.open(image_path)
    draw = ImageDraw.Draw(img)
    draw.rectangle([(x0, y0), (x1, y1)])
    img.show()

```

Total Code from 1 File



Allow Researchers to Better Share/Publish Data Transformation Tools

The screenshot shows a web browser window with the address bar displaying "browndog.ncsa.illinois.edu:9000". The page title is "Browse the Tool Catalog". The navigation bar includes "BD Tool Catalog", "BD Fiddle", "Contribute", "Admin", a search box, and a "Submit" button. A user is logged in as "Kenton McHenry".

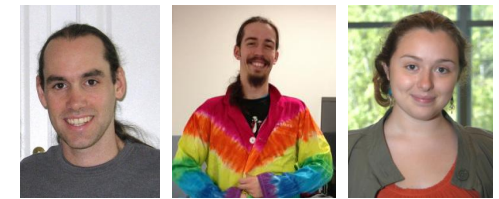
Welcome to the Brown Dog Tools Catalog, a community resource for publishing data transformation tools and making them available for use by the scientific community. To add a new tool click "Contribute".

Tools:

Filter by tool types: All Types Filter by tool levels: All Levels
Displaying 1 to 24 of totally 42 available:

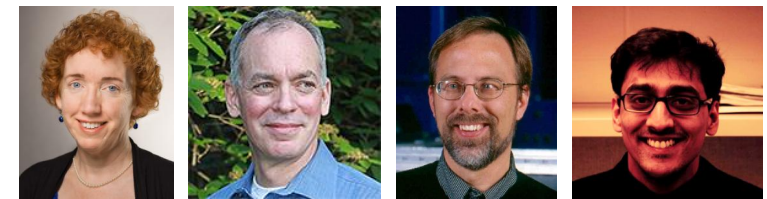
- The Predictive Ecosystem Analyzer**
Convert data sources to formats required by community built ecological models
Tool type: Converter
Tool level: Level 4
Deployments: bd-api bda-api-dev
Submitter: mchenry
Last modified : Nov 23, 2016
- GI Detector**
Identify GI types in satellite images
Tool type: Extractor
Tool level: Level 4
Deployments: bd-api-dev
Submitter: mchenry
Last modified : Nov 22, 2016
- FFmpeg**
Convert between various video formats
Tool type: Converter
Tool level: Level 4
Submitter: mchenry
Last modified : Nov 20, 2016
- Daffodil**
Convert the contents of arbitrary files into an XML representation
Tool type: Converter
Tool level: Level 4
Deployments: bd-api bda-api-dev
Submitter: mchenry
Last modified : Nov 20, 2016
- Ghostscript**
Convert between various document formats
Tool type: Converter
Tool level: Level 4
Deployments: bd-api bda-api-dev
Submitter: mchenry
Last modified : Nov 20, 2016
- OpenJPEG**
Convert JPEG2000 images
Tool type: Converter
Tool level: Level 4
Deployments: bd-api-dev
Submitter: mchenry
Last modified : Nov 20, 2016
- Kobaia**

Project Supported Transformations



Domain	Tool	Type	Description
<i>Biology</i>	netcdf	Converter	Convert from binary netcdf to text
	PEcAn (Ameriflux)	Converter	Convert Ameriflux data to PEcAn's netcdf CF format
	PEcAn (DALEC)	Converter	Convert PEcAn's netcdf CF format to the format required by the DALEC model
	PEcAn (ED2)	Converter	Convert PEcAn's netcdf CF format to the format required by the ED model
	PEcAn (LINKAGES)	Converter	Convert PEcAn's netcdf CF format to the format required by the LINKAGES model
	PEcAn (Sipnet)	Converter	Convert PEcAn's netcdf CF format to the format required by the Sipnet model
	PEcAn 2.0	Converter	Convert AmerifluxBNL, FLUXNET2015 data to PEcAn's netcdf CF format
	PEcAn 2.0	Converter	Convert PEcAn's netcdf CF format to the format required by the FACE, PALEON, NLDAS, CURNCEP, GLDAS, GFDL, BIOCRO, CLM, GDAY, JULES, LPJ-GUESS, MAAT, MAESPA, PRELES models
	PlantCV	Extractor	Extract plant height, area, and color distribution from photographs.

Project Supported Transformations



Domain	Tool	Type	Description
Green Infrastructure	Body of Water Detector	Extractor	Land coverage, extract locations of bodies of water from satellite data
	GI Identification	Extractor	
	Human Preference Score	Extractor	Assign a model derived human preference score to a given image of an urban environment
	Route Greenness	Extractor	Derive the green index of a city route
	Social Media GI Preferences	Extractor	
	Stanford CoreNLP - Sentiment	Extractor	Assign a sentiment score to a piece of text
	TUV Triaxus	Extractor	

Project Supported Transformations



Domain	Tool	Type	Description
<i>Hydrology</i>	Advection Diffusion	Extractor	Solve a general advection-dispersion equation
	Chemical Mean Age	Extractor	Determine the mean age of chemical constituents with inputs of chemical dynamics
	Document Tables Extractor	Extractor	Extract tables from documents
	GDAL	Extractor	
	Historical River Extractor	Extractor	Extract the river networks from the ancient hand-drawing maps and compare them with current river networks
	Normalized Difference Vegetation Index	Extractor	
	River Chi Index	Extractor	Identify the river dynamics in a river basin and evaluate human activities' influences through Chi index in the streams
	River Sinuosity	Extractor	Study the maturity and equilibrium conditions of a stream through the sinuosity index.
	Soil Moisture Change	Extractor	Determine role of hydraulic redistribution in AZ (riparian site / upland site) by studying soil moisture change throughout different seasons
	Species Classifier	Extractor	SAM based Species Classification from Hyperspectral data, Hyperspectral Indices, NDVI, SAVI, MSAVI, etc.

Project Supported Transformations

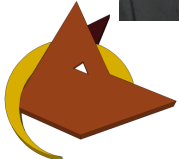


Domain	Tool	Type	Description
<i>Hydrology</i>	TerEx	Extractor	Identify the flat polygons and the heights inside a river valley.
	Topographic Depressions	Extractor	Identify topographic depressions (TDs) and their distribution on landscape (Number, location, area, volume of TDs).
	Tree Delineation	Converter	Tree-wise voxelization of waveform data for lidar metrics that describes canopy structure (max intensity, height, etc...). Individual tree delineation, tree leaf area density to describe vertical leaf distributio.
	Valley Safety Zones	Extractor	Estimate submerging areas and water depths under extreme floods and map the safety zones in a river valley.
	Vegetation Indices	Extractor	Calculating vegetation indices such as NDVI and Surface Temperature from Landsat 7 and 8 satellite data.

Project Supported Transformations

Domain	Tool	Type	Description
S C M C C	General	Tika	Extractor Document extractions such as language identification, ...
		txt2html	Converter Convert text documents to HTML
		Versus – Color Distribution	Extractor Generate a distribution of color values within an image to be used for comparing how similar two images are
		VLFeat	Extractor Classify images as to whether they contain objects from the Caltech101 dataset (e.g. people, airplanes, motorcycles, cougars, ...)
		Zip	Converter Unzip zip archives
	Siegfried	Extractor Extract information about a given file relevant to identifying its type and validating its format	
	Stanford CoreNLP	Extractor Natural Language Process extractions such as parts of speech, named entities, language, etc.	
	Tesseract	Extractor Object Character Recognition (OCR) to extract text from images containing text	

82 transformations to be part of beta ...



<http://browndog.ncsa.illinois.edu>

 @NCSABrownDog