
dcaf Documentation

Release 0.0.1-25-g3b5bd75

Cory Giles and Mikhail Dozmorov

November 07, 2013

Contents

1 Overview	1
2 Contents:	3
2.1 Installation	3
2.2 API Documentation	3
3 Indices and tables	5
Python Module Index	7

Overview

`dcaF` is a Python package containing various bioinformatics code developed in the Wren Lab at the [Oklahoma Medical Research Foundation](#).

This package contains code and scripts for various bioinformatics tasks:

- Wrappers for sequence and variant analysis, principally RNA-seq
- Tools for managing a PostgreSQL database of taxon, gene, expression, ontology, and textual (MEDLINE) data
- Readers for various bioinformatics file formats, including [OBO](#), [SOFT](#), and [BBI](#) (BigWig/BigBED)
- Text mining, enrichment analysis, and other aspects of expression analysis
- Meta-analysis of gene expression data, including gene function prediction and imputation of experimental sample parameters such as tissue, age, and gender

The principal developers are [Cory Giles](#) and [Mikhail Dozmorov](#).

Contents:

2.1 Installation

1. Install the `cython` and `sphinx` Python packages:

```
$ pip install sphinx cython
```

2. Install `dcaf` directly from git:

2.2 API Documentation

2.2.1 `dcaf.db` - Manage a SQL database of genomic data

This module contains utilities to build and query a database containing:

- Expression profiles from GEO
- Text articles from PubMed/MEDLINE
- Metadata from Entrez Gene and RefSeq

Todo

- Add expression profiles from SRA
 - Various genomic feature tables from UCSC
 - Genomic coordinates for genes
-

2.2.2 `dcaf.genome` - Generic manipulation of genomic intervals

2.2.3 `dcaf.util` - Utility functions

Utility functions.

class `dcaf.util.Proxy` (*wrapped*)

A class that wraps an object and forwards all attribute accesses to that object.

Pointless on its own, this class is intended to be subclassed, and subclass methods will be called before the wrapped object's attributes.

`__init__` (*self, wrapped*)

`dcaf.util.coo_to_df` (*triplets*)

Create a SparseDataFrame from a sequence of (row,col,value) triplets.

`dcaf.util.entry_point` (*fn*)

Register a function as a global entry point.

Setuptools will create and install a wrapper script for this function. The function is assumed to take one argument, a list of command-line arguments.

`dcaf.util.find_configuration` ()

Search sys.path for a 'dcaf.cfg' file. If found, return an open ConfigParser for it. Otherwise, return a ConfigParser with the default configuration.

`dcaf.util.multimap` (*pairs*)

Given an iterable of pairs, construct a dict mapping the first elements to sets of second elements.

`dcaf.util.partition` (*n, seq*)

Partition an iterator or collection into chunks of size n.

<p>Warning: This function yields an <i>iterator</i> for each chunk, and as such, will not give correct results if each chunk is not completely realized before calling for the next chunk.</p>

`dcaf.util.which` (*exe*)

Find the full path to an executable (calls the shell command).

2.2.4 dcaf.expression - Manipulate transcript expression data

2.2.5 dcaf.io - Readers/writers for various file formats

dcaf.io.bbi - Read-only API for BigWig and BigBed files

Indices and tables

- *genindex*
- *modindex*
- *search*

Python Module Index

d

`dcaf.util`, 3