PIBASE installation guide. ver 2010

Fred P. Davis, HHMI-JFRC
davisf@janelia.hhmi.org
http://pibase.janelia.org

September 19, 2010

Abstract

This document describes how to set up a local PIBASE installation.

PIBASE can be installed locally by (1) downloading a compiled version of the database from `http://pibase.janelia.org/pibase2010/download.html`, or (2) downloading the PIBASE software package and compiling your own version of the database. Both routes lead to a complete installation that can be accessed through the mysql, perl, or web interfaces.

Part I
Installing the precompiled PIBASE v2010

1   Downloading data

All PIBASE files are available for download at `http://pibase.janelia.org/pibase2010/download.html` under the GPL license.

1.1   PIBASE software

Download the PIBASE software package (pibase_src_v2010.tar.gz) and uncompress it in the directory where you want to install the software:

```
cd yourinstalldirectory
tar xvfz pibase_src.20100906.tar.gz
```

1.2   MySQL data dump

Create an empty MySQL database to hold the PIBASE contents, for example:

```
mysql -u root -p
mysql> CREATE DATABASE pibase ;
mysql> GRANT ALL ON pibase.* TO pibaseuser@hostname IDENTIFIED BY pibasepassword;
mysql> FLUSH PRIVILEGES;
```

Download the PIBASE mysql dump (pibase2010_dump.20100918.out.gz) and load it into the database you just created:

```
zcat pibase2010_dump.20100918.out.gz | mysql -u YOURUSERNAME -p YOURDATABASENAME
```

1.3   Data files

Besides the data in the MySQL tables, there are several kinds of compressed text files that are space-prohibitive as MySQL tables. For example the actual stucture files for individual domains, or the list of pairwise contacts at interfaces. The files you need to download depend on what type of queries or data you are intersted in accessing. For example, for a webserver installation, only the bdp_topology_graphs and subset_files are required. If you are only interested in querying properties that are present in the MySQL tables, you don't need any of these at all.

- bdp_topology_graphs (583 MB) - png and eps pictures of complex topology graphs.

- subset_files (23.3 GB) - PDB files of individual domains.

- metatod - meta tables on disk; the *_file MySQL tables point to these files.

  - bdp_residues (592 MB)- residue listing for each BDP file.
  - bdp_sec_strx (483 MB)- secondary structure assignment for each BDP file.
  - bindingsite_contacts (2.99 GB) - contacts between residues in each binding site
  - bindingsite_secstrx_basic_contacts (530 GB) - contacts between secondary structure elements in each binding site
  - interface_contacts (952 MB)- pairwise contacts across each interface
  - interface_contacts_special (161 MB) - "special" contacts (H-bond, salt bridge, disulfide bridges) across each interface
  - interface_secstrx (1.8GB) - secondary structure content at each interface
  - interface_secstrx_basic_contacts (246 MB) - contacts between basic secondary structure elements at each interface
  - interface_secstrx_contacts (252 MB) - contacts between detailed secondary structure elements at each interface
  - patch_residues (318 MB) - residues in each patch (contiguous region of residues in each binding site)
  - subsets_residues (775 MB) - residues in each domain

2   Web interface

Setting up your own web interface to PIBASE requires you to (1) download the software, (2) create a MySQL database, and (3) download and uncompress the data files for bdp_topology_graphs and subsets_files. Next, there are a few lines to edit in pibase.pm to reflect your database specifications and local directory structure. Then, just copy over the html, cgi-bin, and perl library to your webserver, and it should be ready to query.

   In detail:

1. Edit pibase.pm to reflect your MySQL database speccs. (src/perl_api/pibase.pm lines 57-60).

    ```
    db => 'pibasemysqldatabasename',
    host => 'mysqlserverhostname',
    user => 'pibaseusername' ,
    pass => 'pibasepassword',
    ```

2. Edit pibase.pm to reflect the html and cgi-bin directories of your web server. (src/perl_api/pibase.pm lines 220, 221).

    ```
    $pibase_specs->{web}->{html_dir} = "/var/www/websites/pibase/html/" ;
    $pibase_specs->{web}->{cgi_dir} = "/var/www/websites/pibase/cgi-bin/" ;
    ```

3. Edit pibase.pm to reflect the correct URLS for the html and cgi-bin directories of your web server. (src/perl_api/pibase.pm lines 223, 224).

    ```
    $pibase_specs->{web}->{base_url} = "http://localhost/pibase";
    $pibase_specs->{web}->{basecgi_url} = "http://localhost/pibase-cgi";
    ```

4. Edit pibase.pm to reflect the directories where you uncompressed the bdp_topology_graphs and subsets_files tar files. (src/perl_api/pibase.pm lines 226, 227, 230, 231).

    ```
    $pibase_specs->{web}->{bdp_topology_graphs_baseurl} =
        $pibase_specs->{web}->{base_url}.'/data_files/bdp_topology_graphs'

    $pibase_specs->{web}->{subsets_files_basedir} =
        $pibase_specs->{web}->{html_dir}.'/data_files/subsets_files' ;
    ```

5. The static html pages (web/html/*.html) assume that the CGI directory sits ../cgi-bin relative to the html directory; if this isn't true for your webserver, edit the form submit lines in the html pages to reflect the path to your CGI directory

6. Lastly, copy the contents of the html, cgi, and perl_api directories to your webserver

    ```
    cp -r web/html/* yourwebserver/html
    cp -r web/cgi-bin/* yourwebserver/cgi-bin
    cp -r src/perl_api yourwebserver/cgi-bin/perl_lib
    ```

That should be it, the webserver should be functional now.

Part II

Building your own PIBASE

First, edit pibase.pm to specify the PDB (and optionally, PIBASE) file paths for your local directory structure. Then, `perl src/scripts/build_pibase.pl` should run through the build process.