Network Analysis Tools
Web server installation

Sylvain Brohée
sbrohee@ulb.ac.be

Karoline Faust
kfaust@ulb.ac.be

Jacques van Helden
jvhelden@ulb.ac.be

Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe)
Laboratory of Genome and Network Biology
Université Libre de Bruxelles, Belgium
http://www.bigre.ulb.ac.be/

December 1, 2008

Contents

Description

This document describes the installation procedure for the web server of the **Network Analysis Tools** (*NeAT*).

It assumes that you already installed the perl scripts and the genomes, as described in the *RSAT* installation guide and that it is working properly on the command line and as webserver. To this, please refer to the *RSAT* webserver install guide.

Chapter 1

Web server installation

1.1 Installing a local web server

As the Regulatory Sequence Analysis Tools, *NeAT* includes a web server, which offers a user-friendly interface for biologists. The main server is available for academic users at `http://rsat.ulb.ac.be/neat/`. A few additional mirrors have been installed in different countries.

1.1.1 Web server pages

The web pages are located in the directory *rsa-tools/public_html*. This directory contains both the HTML help pages, and the PHP and CGI scripts.

1.1.2 Apache modules

The *NeAT* interface mainly relies on PHP (and CGI only for the roc-stats tool). These modules should be installed on the web server, and activated in the Apache configuration files. The installation and configuration of CGI is described in manual of the web server of *RSAT*.

To perform the following steps, you might dipose of the administrator permissions.

PHP module for Mac OSX

If your server is running under Mac OSX, you need to install a recent version (at least v5) of the php module, which can be found at the following site. `http://www.entropy.ch/software/macosx/php/`

PHP module for LINUX

Generally, PHP5 is included with the Linux distribution or can easyly be installed using the installer (YAST, YUM, etc). Take care that the PHP5 Module for Apache 2.0 (apache2-mod_php5) is installed.

PHP5 can also be installed manually from the PHP website (`http://www.php.net/downloads.php`).

Modification of php.ini

In order for the server to work with *NeAT*, you have to edit the file *php.ini* which is the main configuration file of PHP. Depending on your PHP installation, this file might be in different directory. In computers running Suse 10.3, this file is in the `/etc/php5/apache2/` directory. In this file, you must modify the following fields accordingly.

```
soap.wsdl_cache_enabled=0
max_execution_time = 2400
max_input_time = 6000 ;
memory_limit = 600M
error_reporting  =  E_ALL & ˜E_NOTICE
post_max_size = 300M
upload_max_filesize = 200M
default_socket_timeout = 3000
upload_tmp_dir = "/tmp/php/"
```

You must change permissions so that the directory `/tmp/php/` for temporary upload is writable by eveverybody. To this please type

```
mkdir /tmp/php
chmod 777 php
```

1.1.3   Tomcat

A part of *NeAT* is using axis web services, JSP and Java servlet pages and needs Tomcat or an equivalent servlet engine to run. Tomcat can be easily installed on SUSE with yast and is usually located in */usr/share/tomcat* after installation. On MacOS, it is already installed by default. Make sure to install at least Tomcat version 5. We noticed however that Tomcat version 6 is more stable. We will refer from now on to the Tomcat root directory as $CATALINA_HOME.

Configuration of Tomcat

If you would like to use the Tomcat manager, make sure to configure the file tomcat-users.xml located in $CATALINA_HOME/conf.

Add a manager role with a special login and password, e.g.

```
<tomcat-users>
  <role rolename="tomcat"/>
  <role rolename="manager"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="metheadmin" password="mysecretpassword" roles="manager"/>
</tomcat-users>
```

Make sure that tomcat-users.xml cannot be read by anyone else than tomcat or root.

By default, Tomcat takes a lot of memory. You can reduce this amount by modifying the file tomcat.conf in $CATALINA_HOME/conf. Outcomment and adjust the variable JAVA_OPTS as follows:
JAVA_OPTS="-Xmx250m"

You can start Tomcat on SUSE with
/etc/init.d/tomcat6 start
and stop it with
/etc/init.d/tomcat6 stop

More advice for the installation and configuration of Tomcat on unix systems can be found at: *http://linux-sxs.org/internet_serving/book1.html*

Mod_jk

Mod_jk forwards requests for defined contexts from Apache to Tomcat. Users that have a strict firewall configuration blocking the Tomcat port will be able to access the web services through the Apache default port.

To install and configure mod_jk, you can follow the steps below:

1. Install the mod_jk module from *http://tomcat.apache.org/connectors-doc/* or via installation systems like yast.

2. In the Apache configuration folder (e.g. */etc/apache2/conf.d*), add two configuration files: workers.properties and tomcat.conf

3. Content of workers.properties

```
# Define 1 real worker using ajp13
  # this coud be al list in the format
  # worker.list=worker1, worker2, worker3, worker4
  worker.list=worker1
  # Set properties for worker1 (ajp13)
  worker.worker1.type=ajp13
  worker.worker1.host=localhost
  worker.worker1.port=8009
```

4. Content of tomcat.conf

```
# Update this path to match your modules location
LoadModule jk_module /usr/lib/apache2/mod_jk.so
# Where to find workers.properties
# Update this path to match your conf directory location
JkWorkersFile /etc/apache2/conf.d/workers.properties
# Where to put jk shared memory
# Update this path to match your local log directory
# JkShmFile /var/log/apache2/mod_jk.shm
# Where to put jk logs
# Update this path to match your logs directory location
JkLogFile /var/log/apache2/mod_jk.log
# Set the jk log level [debug/error/info]
JkLogLevel error
# Send everything for context to worker named worker1 (ajp13)
JKMount /be.ac.ulb.bigre.graphtools.server/* worker1
JKMount /metabolicpathfinding/* worker1
```

5. Reload Apache (on SUSE */etc/init.d/apache2* reload).

Preparation of folders for Tomcat

In Linux, a tomcat user is created if Tomcat is installed via yast. Make sure that this user has read, write and execution rights for the following folders:
$RSAT/contrib/REA
$RSAT/contrib/kwalks/bin
$RSAT/public_html/data/KEGG
$RSAT/public_html/data/Stored_networks

   If any of these folders does not yet exist, create them.
   For installation of REA and kwalks see the section on third-party programs in the RSAT install guide.

1.1.4 Installation of Java tools server

The graphtools server contains the java web services of NeAT. It is stored as a war file
in *$RSAT/java/web*

Either place the war file *be.ac.ulb.bigre.graphtools.server.war* in $CATALINA_HOME/webapps
and then start Tomcat or use the Tomcat manager
(*http://localhost:8080/manager/html*) to deploy the war file.

Go to *$CATALINA_HOME/webapps/be.ac.ulb.bigre.graphtools.server/WEB-INF*, open
serverConfig.txt and set the value of RSAT_ROOT to the value of $RSAT.

You are done. You may read the section "Configuration remarks" for additional
information. If you want to install the metabolic pathfinder, continue with section
"Metabolic Pathfinder".

Configuration remarks

The directory *$RSAT/public_html/data/Stored_networks* allows to store graph files for
longer time (remember that *$RSAT/public_html/tmp* should be cleaned regularly).

Organism-specific KGML files are placed in *$RSAT/public_html/data/KEGG* in a
folder with the KEGG organism name (e.g. sce). You may place KGML folders for
organisms yourself or you may let the program download required KGML files on the
fly. The default KGML version might be outdated (check KEGG Release Notes for the
correct version). To update the KGML version,
go to *$CATALINA_HOME/webapps/be.ac.ulb.bigre.graphtools.server/WEB-INF*, open
serverConfig.txt and modify the value of the variable KGML_VERSION.

1.1.5 Metabolic Pathfinder

After successful installation of the Java web services, you may install metabolic pathfinder,
which is a client of the pathfinder web service.

Requirements of metabolic pathfinder

Metabolic pathfinder has a number of additional requirements. It needs dot, which can
be obtained freely from *http://www.graphviz.org/*. Dot is needed to draw graphs.

It also needs postgres (version 8.2 or later) to store KEGG data. Alternatively,
KEGG data could be accessed via its API, but this takes much longer than accessing
locally stored data. Postgres can be obtained freely from *http://www.postgresql.org/*.
For MacOS, you can use Darwinports to install postgres. On SUSE, you may install it
with yast (install postgresql and postgresql-server).

On SUSE, you may start the postgres server with *etc/init.d/postgresql start*. On
MacOS, you may start the postgres server using a command similar to
pg_ctl -D /usr/local/pgsql/data/ -l logfile start.

The KEGG data is packaged as a postgres backup file located in
$RSAT/java/misc/metabolicdb_dump_13_Oct_2008.backup, which can be loaded into
postgres as follows:

1. Start postgres by typing the following on command line:
   psql -U postgres

2. In postgres, do:
   create database "metabolic_database" with owner "karoline" encoding='UTF8';

3. Quit postgres and type on command line:
   pg_restore -d metabolic_database metabolicdb_dump_13_Oct_2008.backup

Installation of metabolic pathfinder

Either place the war file *metabolicpathfinding.war* located in *$RSAT/java/web* in
$CATALINA_HOME/webapps and then start Tomcat or use the Tomcat manager to
deploy the war file.

Configuration of metabolic pathfinder

Go to the folder *$CATALINA_HOME/webapps/metabolicpathfinding/WEB-INF*.
    Run the configuration script with the following command:
./configureWebxml.pl
    Open the web.xml file and set the variable "dotpath" to the location of the dot
binary, e.g. */usr/bin/dot*
    Finally, reload the metabolic pathfinding web application using the Tomcat manager.
    When this is done, you have to restart the apache web server with the following
command.

```
/etc/init.d/apache2 restart
```

1.1.6   Web services

Edit the WSDL file

The web interface consists in web services that are called by the PHP web pages. Your
computer must thus act as web service server.
    Firt, edit the file *RSATWS.wsdl* located in the `\$RSAT/public_html/web_
services/` directory. At the very end of the file, the line

```
<soap:address location="http://rsat.scmbb.ulb.ac.be/rsat/web_services/RSATWS.cgi"/>
```

must be replaced by

```
<soap:address location="url_of_the_cgi_file_on_your_server"/>
```

The URL can be `http://127.0.0.1/rsa-tools/web_services/RSATWS.
cgi`.

Edit the *NeAT* config file

Edit the *RSAT_config.props* present in the main RSAT directory and edit the following
fields so that they correspond to your local configuration.

```
neat_supported=1
neat_ws=web link to the WSDL file on your computer (e.g. http://127.0.0.1/rsa-tools/web_services/RSATWS.ws
```

Change permissions of the temporary files and log files directories

The directories `\$RSAT/public_html/logs/` and `\$RSAT/public_html/`
`tmp/` must be writable. So, change the permissions

```
chmod 777 \$RSAT/public_html/logs/
chmod 777 \$RSAT/public_html/tmp/
```