Haddock: small tutorial

This is a very simplified HADDOCK tutorial based on the official one in HADDOCK web page:
http://www.nmr.chem.uu.nl/haddock2.0/tut_e2a-hpr-csp.html

Now it has been removed from the web, but you can still read it through the "wayback machine" in the internet archive:
https://web.archive.org/web/20080101191054/http://www.nmr.chem.uu.nl/haddock2.0/tut_e2a-hpr-csp.html

All necessary files to complete the full tutorial from scratch can be found in the "**examples**" directory inside the **Haddock2.1** installation dir.

In this tutorial we are going to dock two proteins from the phosphoenolpyruvate sugar phosphotransferase system. The histidine-containing phosphocarrier protein (HPr) and the glucose-specific enzyme IIA.

You will have to edit text files while completing this tutorial, here there are instructions to do it with a text editor software called "vim", if you prefer other text editors you are encouraged to use them, if you prefer to use vim you should be familiarized with its usage (https://www.engadget.com/2012/07/10/vim-how-to/)

Since the computer cluster where our calculations are going to run is not integrated with our current desktop, Download and uncompress all necessary files in your **local** computer.
http://bioinf.uab.es/ocs/MSCB/MOD4/Haddock21_tutorial.tbz2

Make an ssh connection the computer cluster and you can start the tutorial.
```
ssh -oKexAlgorithms=+diffie-hellman-group1-sha1 -oHostKeyAlgorithms=+ssh-rsa (youruser)@cellerui.uab.es
```

Once you are connected to **cellerui** I strongly advice that you execute the following command:
```
screen
```
And continue working as usual.
"`screen`" performs a function very similar to the "`tmux`" command (maybe you know `tmux`). Here you will find a small introduction:
https://www.liquidweb.com/blog/how-to-use-the-screen-command-in-linux/

Before doing anything, we need to make sure that all ENVIRONMENT VARIABLES that haddock needs are defined in the system.
Execute this:
```
vim /home/$USER/.cshrc
```

make sure that:
- this line is present:

```
alias had2.1source 'set path = ( /share/Sdata3/MSCB/INSTALL/EPD/epd_free-7.3-2-rh5-x86_64/bin $path);source /share/Sdata3/MSCB/${USER}/BIOSOFT/HADDOCK/haddock2.1/haddock_configure.csh'
```
(You can close vim by pressing "Esc" and then :q and the enter key.)

- the file exists:

```
less /share/Sdata3/MSCB/${USER}/BIOSOFT/HADDOCK/haddock2.1/haddock_configure.csh
```

- HADDOCK variable points to the **haddock2.1** directory, this directory is in your work directory and it belongs to your user.

```
ls -lad /share/Sdata3/MSCB/$USER/BIOSOFT/HADDOCK/haddock2.1
```

We are also going to need some initial data to work with (protein structures)

make sure that in your work directory

```
cd /share/Sdata3/MSCB/$USER
ls
```

there is a file called:

```
Haddock21_tutorial.tbz2
```

Or a directory called **HADDOCK**

```
ls HADDOCK
```

If **HADDOCK** directory is not present we need to uncompress the **`Haddock21_tutorial.tbz2`** file. Execute:

```
tar jxf Haddock21_tutorial.tbz2
```

if nor "**`Haddock21_tutorial.tbz2`**" nor the directory "**`HADDOCK`**" are present in your working directory /share/Sdata3/MSCB/$USER) you can download it with the following command:

```
wget "http://bioinf.uab.es/ocs/MSCB/MOD4/Haddock21_tutorial.tbz2"
```

Haddock uses a collection of **csh** scripts, so in order to work properly we need to work with **csh** (or **tcsh**).
Execute this:

```
tcsh
```

Now we can load Haddock environment variables; execute:

```
had2.1source
```

Once the file **`Haddock21_tutorial.tbz2`** has been uncompressed it generates a directory called **HADDOCK** and inside it there are two more : **WORK** and **e2a-hpr**, all we are going to use is in **e2a-hpr**.
Inside e2a-hpr we can find **`e2a_1F3G.pdb`** and **`hpr_??.pdb`**, these are the two proteins that we are going to dock.

```
cd /share/Sdata3/MSCB/$USER/HADDOCK/e2a-hpr
ls e2a_1F3G.pdb
ls hpr_??.pdb
```

The structure of **hpr** protein has been solved by NMR so we have an ensemble of structures conveniently separated in independent files and they are listed in a file called :

```
hpr-files_n.list
```

When we visualize it :

```
less hpr-files_n.list
```

we can see that files are not listed with our path, so we have to correct it.
Execute this:

```
cd /share/Sdata3/MSCB/$USER/HADDOCK/e2a-hpr
ls -1v $PWD/hpr*pdb | xargs -i echo '"{}"' >hpr-files_n.list
```

Due to the fact that haddock needs some direction to perform the docking we need to know some important residues that are going to drive the docking process.

There are two kinds of important residues: active and passive ones. Active are those that you know from some sort of experimental data (or bioinformatics predictions, but…) and Passive are all their surrounding neighbors with an accessibility surface higher than 40-50 % (depending on the case, the researcher, etc)

In Our case we know that for **e2a** these residues are important:
38 40 45 46 69 71 78 80 94 96 141

And for **hpr** these ones:
15 16 17 20 48 51 52 54 56

So we need to find the passive ones. We have multiple ways to do it, but the easiest is to look at the protein and see which ones are close. But first of all we need to know the accessibility surface of all residues.
We can do this with a program called **nacces**:
DO NOT execute this:
```
naccess e2a_1F3G.pdb
```

To increase the speed of the practice that has already been executed and results are in a file called "**e2a_1F3G.rsa**", we can see them using any text editor:
Execute this:
```
less e2a_1F3G.rsa
```

So now we are ready to define passive residues for e2a
Execute this **in your local machine**:
```
rasmol e2a_1F3G.pdb -script e2a_rasmol_active.script
```

This opens a molecular visualization program called rasmol where we can see the protein in white with the active residues colored red, now we click on one of the surrounding neighbors to see which residue it is and then we go to the **`e2a_1F3G.rsa`** file to see if it is surface accessible. A residue is considered surface accessible if it has a **REL**ative surface accessibility over 40-50 for **All-atoms** or for the Side chain ones (**Total-Side**).

That has already been done, to see the final residues close the actual rasmol and start it again with the following command:
```
rasmol e2a_1F3G.pdb -script e2a_rasmol_active-passive.script
```
In green you will see the selected passive residues.

We should repeat this with the **hpr** protein, but, in this case, due to the fact that **hpr** was obtained by NMR so we have an ensemble of structures, we have to work with surface accessibility average values. To obtain such values we would use the script: (DO NOT EXECUTE IT NOW)

```
$HADDOCKTOOLS/calc_ave_rsa.csh *.pdb
```

All data generated by this script can be seen in the previously mentioned WORK directory, but we are only interested i the final file that contains the average surface accessibility values for each residue:

```
less /share/Sdata3/MSCB/$USER/HADDOCK/WORK/rsa_ave.lis
```

After all of this is complete we have the passive residues for both molecules:
E2A passive: P37 V39 E43 G68 E72 E97 E109 K132 (37 39 43 68 72 97 109 132)
HPr passive: N12 Q21 K24 L47 K49 Q57 E85 (12 21 24 47 49 57 85)

Now we can generate the AIR file using this webpage:
http://www.nmr.chem.uu.nl/haddock2.0/generate_air.html

Also not working anymore, you can download and use this perl script:
http://bioinf.uab.es/ocs/MSCB/MOD4/generate_air.pl

We have already done it and it is called "`e2a-hpr_air.tbl`"
Execute this to see it:

```
less e2a-hpr_air.tbl
```

Now we have everything we need to start the docking process. To prepare all necessary files needed to run the docking Haddock reads a file called **new.html**, we need to edit it and configure our data:

```
vim new.html
```

it must end up like this:

```
<html>
<head>
<title>HADDOCK - start</title>
</head>
<body bgcolor=#ffffff>
<h2>Parameters for the start:</h2>
<BR>
<h4><!-- HADDOCK -->
AMBIG_TBL=/share/Sdata3/MSCB/YOUR_USER_HERE/HADDOCK/e2a-hpr/e2a-hpr_air.tbl<BR>
HADDOCK_DIR=/share/Sdata3/MSCB/YOUR_USER_HERE/BIOSOFT/HADDOCK/haddock2.1<BR>
N_COMP=2<BR>
PDB_FILE1=/share/Sdata3/MSCB/YOUR_USER_HERE/HADDOCK/e2a-hpr/e2a_1F3G.pdb<BR>
PDB_FILE2=/share/Sdata3/MSCB/YOUR_USER_HERE/HADDOCK/e2a-hpr/hpr_1.pdb<BR>
PDB_LIST2=/share/Sdata3/MSCB/YOUR_USER_HERE/HADDOCK/e2a-hpr/hpr-files_n.list<BR>
PROJECT_DIR=/share/Sdata3/MSCB/YOUR_USER_HERE/HADDOCK/e2a-hpr<BR>
PROT_SEGID_1=A<BR>
PROT_SEGID_2=B<BR>
RUN_NUMBER=1<BR>
submit_save=Save updated parameters<BR>
</h4><!-- HADDOCK -->
</body>
</html>
```

With vim you can do it pressing ":" and later:

```
%s/data1\/2055583/Sdata3\/MSCB\/YOUR_USER_HERE/g
```

We can check that the file locations we have changed exists with this command:

```
grep "/share" new.html | cut -d "=" -f 2 | cut -d \< -f 1 | xargs -i ls -ld {}
```

After this execute:

```
haddock2.1
```

HADDOCK has just copied all necessary files in the **run1** directory, let's go there and check it:

```
cd run1
ls -l
```

One of these directories is "**`structures`**", here is where your docking results will be. You can also see a file called "**`run.cns`**", in this file is where all your docking parameters are defined. Let's customize it:

```
vim run.cns
```

then we press "**:**", type "**set number**" and then press enter.

```
:set number
```

Now we should be seeing the line numbers.

Change line 1271 (number of structures for rigid body docking) to this:

```
{===>} structures_0=200;
```

line 1274 (number of structures for refinement)

```
{===>} structures_1=100;
```

line 1278 (structures to be analyzed)

```
{===>} anastruc_1=100;
```

Line 1419 (number of structures for the explicit solvent refinement)

```
{===>} waterrefine=100;
```

Make sure that line 1540 (submission script) looks like this:

```
{===>} queue_1="/share/data0/txino/BIOSOFT/HADDOCK/PERL/had_sge_wrap_multiuser.pl";
```

line 1541 (the **cns** location) like this:

```
{===>} cns_exe_1="/share/data0/txino/BIOSOFT/CNS/cns_solve_1.3/intel-x86_64bit-linux/bin/cns";
```

and line 1542 (number of possible simultaneous jobs) like this:

```
{===>} cpunumber_1=20;
```

Haddock2.1 by default uses cns1.2 with topology version 5.3 (like haddock2.0) but we are going to use cns1.3 so we must change topology version to 5.4. For this purpose, in vim enter in mode "execute" (":") and execute this (without typing the starting ":")

:%s/allhdg5.3/allhdg5.4/g

to replace all mentions of 5.3 topology to 5.4

Once we finish editing "`run.cns`" we save it and close vim pressing ":" and typing "wq"

:wq

After this, it is a good moment to look in detail the run.cns file and see what parameters can be customized and what information can be provided.

Now everything is ready to start the docking run, so we execute:

```
haddock2.1
```

This will take some time, once it is done we should analyze the results, this process is very well explained in the tutorial:
http://www.nmr.chem.uu.nl/haddock2.0/tut_e2a-hpr-csp.html#analysis
Again “wayback machine”:
https://web.archive.org/web/20080101191054/http://www.nmr.chem.uu.nl/haddock2.0/tut_e2a-hpr-csp.html#analysis

It consists in executing various scripts in order to clusterize the generated structures according to their structural similarity (rmsd). As reported by haddock developers the best structure is the one with the lowest score (“energy”) from the cluster with the lowest score, but this is not always like this.

This can be accomplished executing the following scripts:

All of this has already been done and the results can be found inside the following directory:
`/share/Sdata3/MSCB/$USER/HADDOCK/e2a-hpr/run1_OSCAR`

Inside the directory **structures/it1/water** execute:
```
$HADDOCKTOOLS/ana_structures.csh
less -S structures_haddock-sorted.stat
```

Then with xmgrace generate a chart relating haddock score and rmsd to the lowest score structure:
```
$HADDOCKTOOLS/make_ene-rmsd_graph.csh 3 2 structures_haddock-sorted.stat
xmgrace ene_rmsd.xmgr #execute it in your local workstation (not in cellerui)
```

Observing this last plot: How many clusters do you expect? Which one do you think is the best one?

Now you have to clusterize the structures. Move to directory **structures/it1/water/analysis** and execute:
```
$HADDOCKTOOLS/cluster_struc e2a-hpr_rmsd.disp 7.5 4
```

To visualize this command output in the run1_OSCAR directory execute:
`cat cluster7.5.out`

7.5 is the rmsd cutoff and 4 is the minimum number of members to define a cluster. It is highly recommended that at least 50% of the generated structures are clusterized.

It shows which structures belong to which cluster, numbers are the structures position in the file “file.list” which is a sorted list of the generated structures from the one with the best score (lowest) to the worst. So, THEORETICALLY, the cluster with the lowest numbers is the best cluster and its structure with the lowest number is the best. The first number of each cluster is the central structure in the cluster, not the best.

Just observing **cluster_struct** output, do you think you can choose a structure?

This output can be used to analyze the solutions on a cluster basis and obtain cluster average values. Execute the following commands:
```
$HADDOCKTOOLS/cluster_struc e2a-hpr_rmsd.disp 7.5 4 >cluster7.5.out
cd ..
```

```
$HADDOCKTOOLS/ana_clusters.csh analysis/cluster7.5.out
```

Once it is finished you can check which cluster has the lowest averaged score (so theoretically, the best one) :

```
less -S clusters_haddock-sorted.stat
```

You can check if this is correct looking inside run1_OSCAR/structures/it1/water/analysis/CLUS_1, water/analysis/CLUS_2, etc

Was your choice correct?

Inside this directories you can find the “real” structure for this complex solved by “NMR” experiments (**`e2a-hpr_1GGR.pdb`**) and scripts to compare it to any generated structure present in this directory (**`compare_vs_reference.sh`, `pfzonBBW.pl`**)

If you want you can rerun the docking process using the recommended number of structures for every step (this has also been done and results can be found in run2_OSCAR):

create a run2 directory (modifying new.html) and in run.cns keep the following values (remember to do all necessary changes as explained before)

Line 1271 (number of structures for rigid body docking)

```
{===>} structures_0=1000;
```

line 1274 (number of structures for refinement)

```
{===>} structures_1=200;
```

line 1278 (structures to be analyzed)

```
{===>} anastruc_1=200;
```

Line 1419 (number of structures for the explicit solvent refinement)

```
{===>} waterrefine=200
```

And do it following the full tutorial:
https://web.archive.org/web/20080101191054/http://www.nmr.chem.uu.nl/haddock2.0/tut_e2a-hpr-csp.html

Any question or problem you have, please, do not hesitate to mail me:
Oscar.Conchillo@uab.cat