ypkpathway

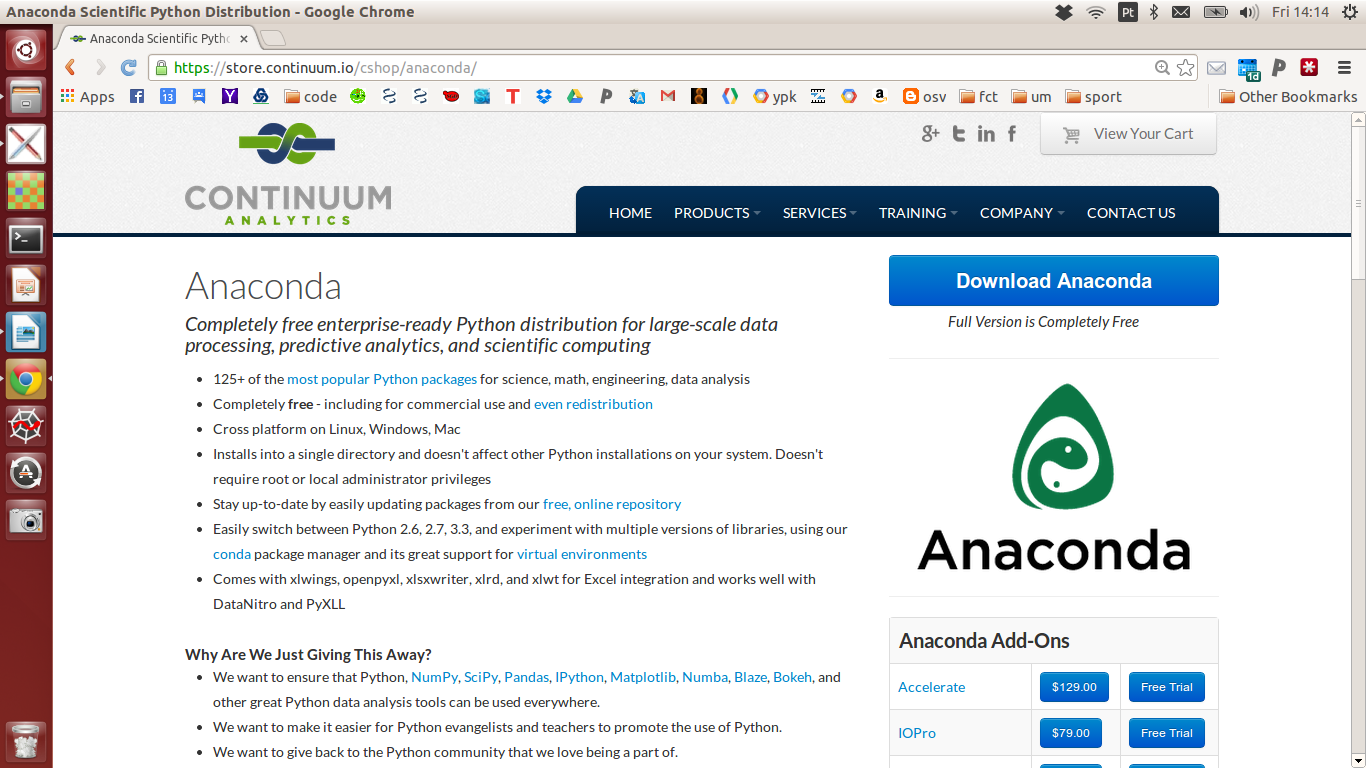
Ypkpathway is a software tool for the automated planning of metabolic pathway assemblies using the Yeast Pathway Kit (Pereira et. al. 2014).

## Installation

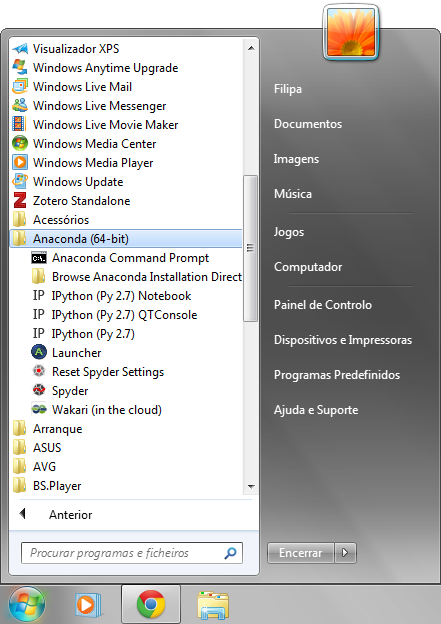
Python 2.7 and the python packages **pydna, networkx**, **biopython**, and **docutils** are required to run ypkpathway. The easiest and most efficient way to install ypkpathway is by first installing the free Anaconda Scientific Python distribution from Continuum analytics. It is a large download, but it installs cleanly in one folder in the users directory (regardless of the operating system) and is easily removable if necessary. Anaconda is available for Windows, Mac and Linux.

## Anaconda installation

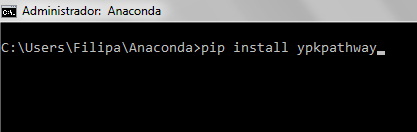
Go to the website of Anaconda at <https://store.continuum.io/cshop/anaconda/> (Fig1).

Fig 1

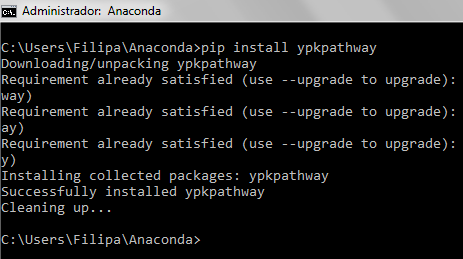
Download Anaconda installation file for your operating system and follow the installation instructions. Once Anaconda is installed, find and start the Anaconda Command Prompt (see Fig2)

Fig 2 Finding the Anaconda Command Prompt on MS Windows

The Anaconda Command Prompt starts a terminal window. Write “pip install ypkpathway” at the command prompt followed by return (Fig 3). This command will download ypkpathway and install it and all needed dependencies in one go, so make sure you are connected to the internet.

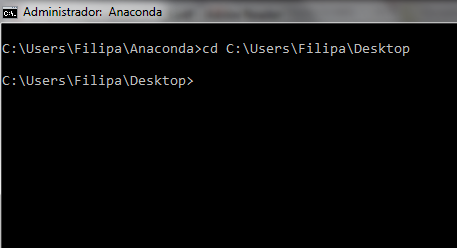
Fig 3

The installation process will generate some text output, important is that it ends with “Successfully installed ypkpathway Cleaning up...” (Fig 4).

Fig 4

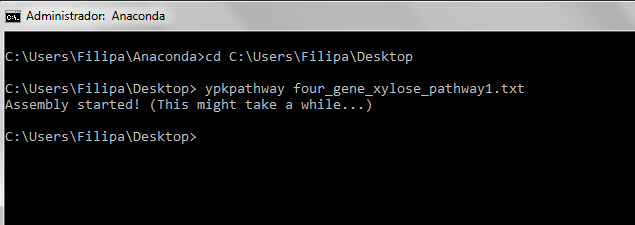
# Use

Ypkpathway is meant to be used in the terminal. It is most practical to navigate the terminal to where you have the data file that ypkpathway should process. Issue the commands (“cd” means change directory) in Fig 5 to specify the working directory where the program will read the data files and and write the results. In this case we chose the Desktop (cd C:\Users\Filipa\Desktop), the actual name of the folder will be different on another computer.

Fig 5

The syntax is very simple as ypkpathway takes only one argument which is the name of the data file to be processed, in this case four\_gene\_xylose\_pathway1.txt (Fig 6) which is a datafile accompanying this document.

Fig 6

Fig 7

The data file can have any name as long as it is a text file containing the sequences to be assembled. See next section “Indata” for proper formatting of this file. The result is a new folder created in the same directory called ypk\_assembly containing the assembly report (Fig 7).

# Indata

The data file is simply a list of the TPs and genes that should be assembled in a text format (either FASTA or Genbank). The datafile.txt file (which can have a different name) can have the structure depicted in Fig 8. The sequences in Fig 8 are truncated for clarity and could also be given in Genbank format or a mix of FASTA and Genbank formats. The sequences could be linear fragments (as in the example four\_gene\_xylose\_pathway1.txt file accompanying this document).

Fig 8 Two sets of three sequences, each forming a tp gene tp cassette. Dots symbolize sequence not shown for clarity.

The Yeast Pathway Kit was designed for the reuse of the genetic parts, especially terminator-promoter and genes cloned in the pYPKa. These plasmids are named pYPKa\_Z\_XXXN and pYPKa\_E\_XXXN, where XXXN represent the actual identifier of the tp or gene. Once constructed, terminator-promoter plasmids can be reused for other pathways, in which case they do not need to be constructed again.

Sequences can also be given to the ypkpathway program in the form of the entire pYPKa\_Z\_XXXN, pYPKa\_A\_ XXXN or pYPKa\_E\_ XXXN sequences, typically generated from a previous assembly experiment. These will be recognized by the ypkpathway algorithm and the assembly report will indicate that these were given and not constructed (Fig 9).

Fig 9 The pYPKa sequences will not be assembled by the algorithm.

>pYPKa\_Z\_TEF1tp

tcgcgcgttt...ACAATGC...AAA...ctttcgtc

>pYPKa\_A\_gene1

tcgcgcgttt...atgatc...taa...ctttcgtc

>pYPKa\_E\_TDH3tp

tcgcgcgttt...ATAAAAA...AAA...ctttcgtc

>pYPKa\_Z\_TDH3tp

tcgcgcgttt...ATAAAAA...AAA...ctttcgtc

>pYPKa\_A\_gene2

tcgcgcgttt...atgcac...tag...ctttcgtc

>pYPKa\_E\_TPI1tp

tcgcgcgttt...TGTTTAA...AAA...ctttcgtc

In the same way sequences can be supplied as pYPK0\_tp\_gene\_tp sequences (Fig 10).

Fig 10

>pYPK0\_TEF1tp\_gene1\_TDH3tp

tcgcgcgttt...ACAATGC...AAA...atgatc...taa...ATAAAAA...AAA...ctttcgtc

>pYPK0\_TDH3tp\_gene2\_TPI1tp

tcgcgcgttt...ATAAAAA...AAA...atgcac...tag...TGTTTAA...AAA...ctttcgtc

The ypkpathway algorithm also permits the use of data files with any valid combination of the three kinds of sequences (Fig 11). In the example in Fig 10 two pYPKa sequences were supplied, one for the tp1 and one for the gene1. The tp2 was never cloned before, so it was given as a linear sequence. The pYPK0\_TDH3tp\_gene2\_TPI1tp vector was made in a previous experiment and was also given.

Fig 11

>pYPKa\_Z\_TEF1tp

tcgcgcgttt...ACAATGC...AAA...ctttcgtc

>pYPKa\_A\_gene1

tcgcgcgttt...atgatc...taa...ctttcgtc

>TDH3tp

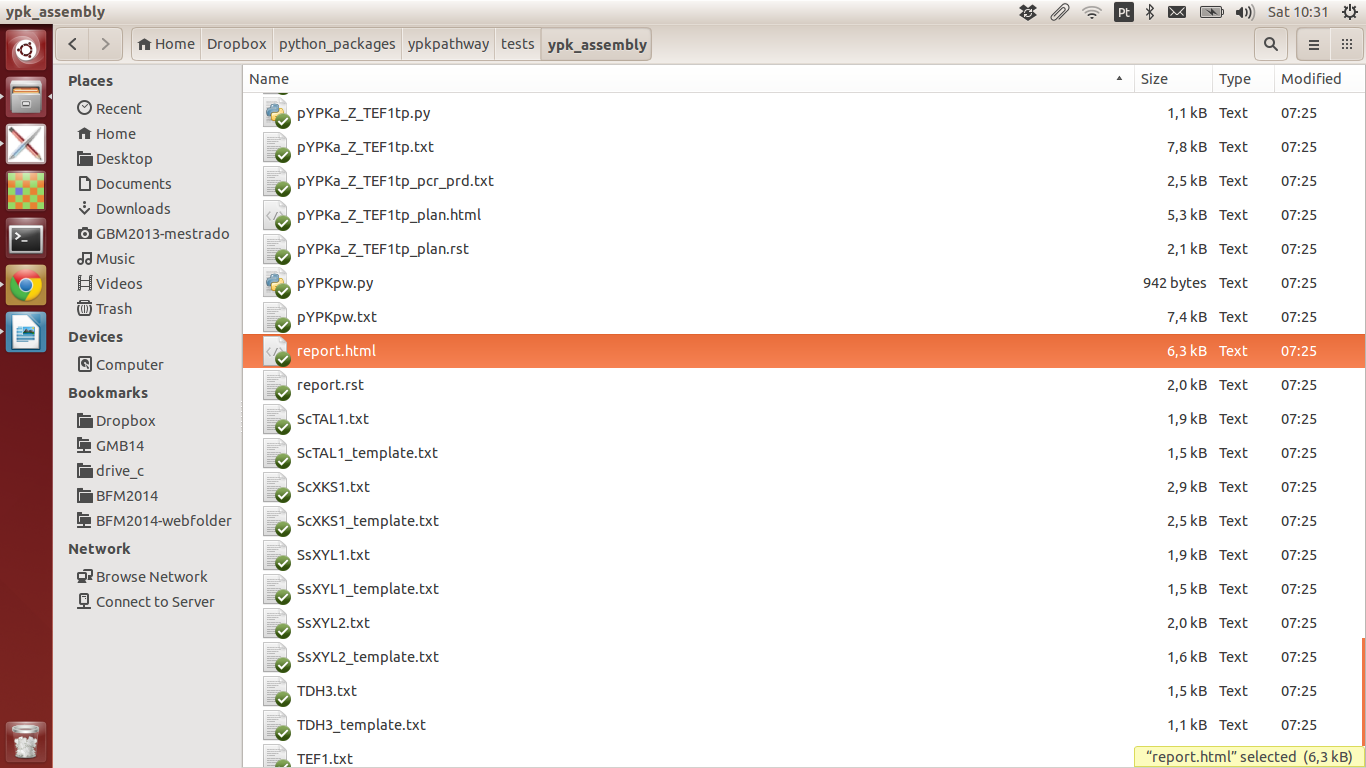
ATAAAAA...AAA

>pYPK0\_TDH3tp\_gene2\_TPI1tp

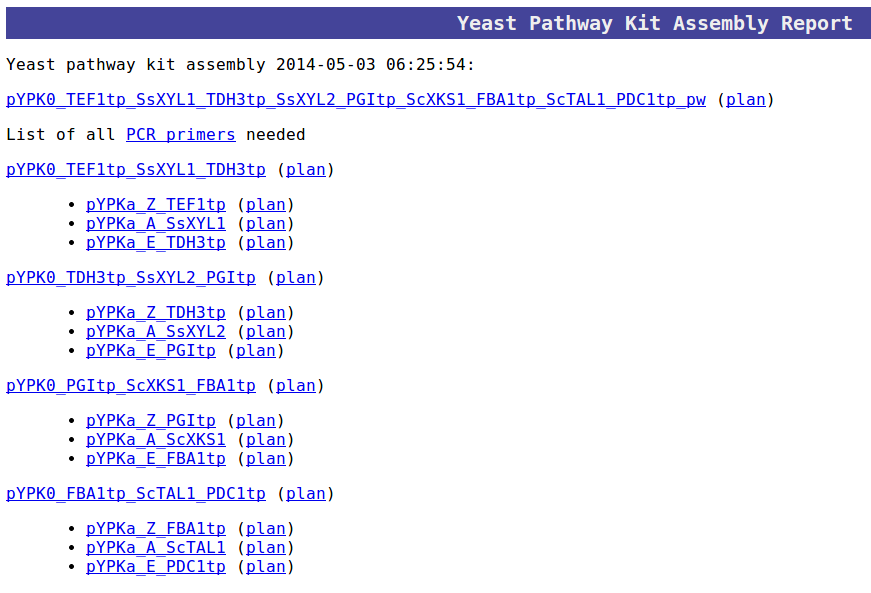
tcgcgcgttt...ATAAAAA...AAA...atgcac...tag...TGTTTAA...AAA...ctttcgtc

# Output

The ypkpathway program creates a folder in the current working directory (the directory from which ypkpathway was called). The folder is called “ypk\_assembly”. This folder will be overwritten by ypkpathway if it already exists.

Fig 12

Open this folder (Fig 12) and open the file “report.html” in the web browser (Fig 11). A web page will be created in the browser looking like Fig 13.



1

2

3

4

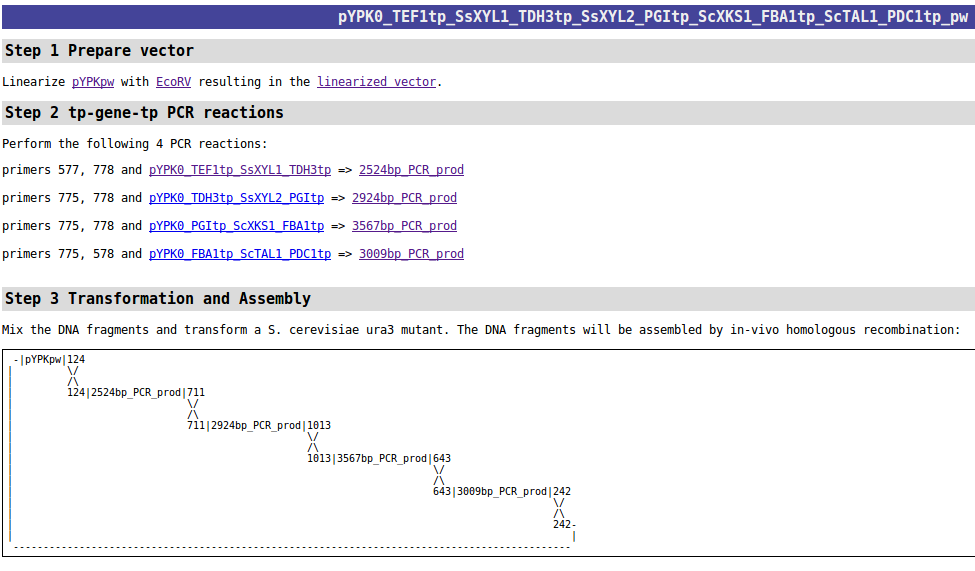
Fig 13:

Clicking on the first link “pYPK0\_TEF1tp\_SsXYL1\_TDH3tp\_SsXYL2\_PGItp\_ScXKS1\_FBA1tp\_ScTAL1\_PDC1tp\_pw”

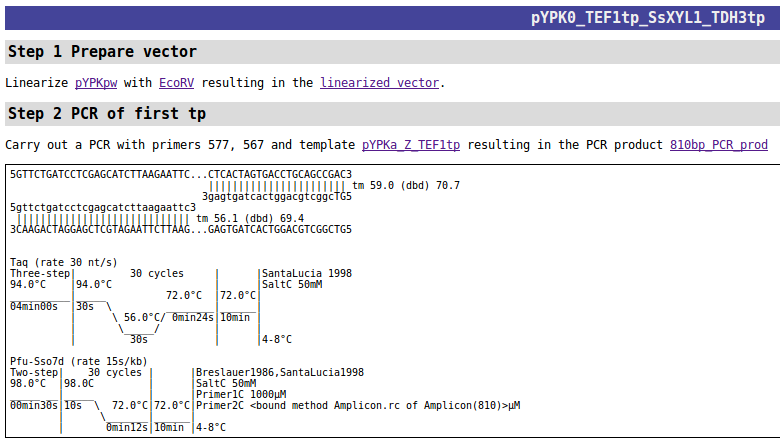
(Fig 13-1) will display the final sequence of the pathway in the browser, a 14800 bp sequence in this case (Fig 14).

Fig 14

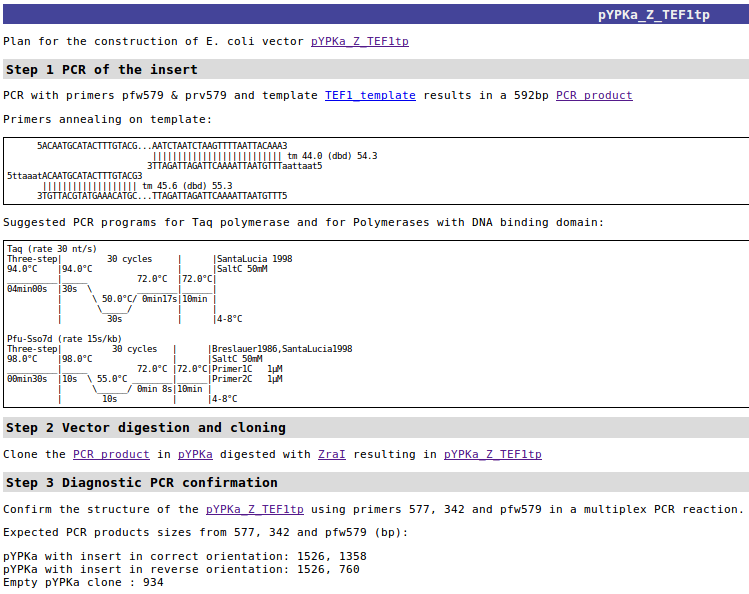
The “(plan)” link (Fig 13-2) displays a small representation of how the final sequence was assembled (Fig 15). The image shows how four PCR fragments were assembled from PCR products derived from pYPK0 tp\_gene\_tp clones and linearized pYPKpw sequence to form the final circular construct.

Fig 15

The second “(plan)” link (Fig 13-3) shows a plan for the construction of the first pYPK0\_tp\_gene\_tp clone. These clones are assembled from three pYPKa derived pcr products for each element and linearized pYPKpw (Fig 16).

Fig 16

The last “(plan)” link (Fig 12-4) show a plan for the construction of the first pYPKa clone (Fig17). These clones are made from pYPKa vectors linearized with ZraI, AjiI or EcoRV and a linear PCR product.

Fig 17

All PCR primers needed for the construction and verification of the pathway can be found under the “PCR primers” link on the report page (Fig 18). Primers are devided between specific and general primers. The specific primers are generated whenever new genes and tps are to be cloned in the pYPKa vector. General primers are vector specific primers used in any assembly project.

Specific Primers:

pfw579 ttaaatACAATGCATACTTTGTAC

prv579 taattaaTTTGTAATTAAAACTTAGATTA

pfw957 aaATGCCTTCTATTAAGTTGAA

prv957 TTAGACGAAGATAGGAATCTT

pfw698 ttaaatATAAAAAACACGCTTTTTC

prv698 taattaaTTTGTTTGTTTATGTGTGTT

pfw1092 aaATGACTGCTAACCCTTC

prv1092 TTACTCAGGGCCGTCA

pfw1000 ttaaatAATTCAGTTTTCTGACTGA

prv1000 taattaaTTTTAGGCTGGTATCTTG

General primers:

577 gttctgatcctcgagcatcttaagaattc

578 gttcttgtctcattgccacattcataagt

468 gtcgaggaacgccaggttgcccact

467 ATTTAAatcctgatgcgtttgtctgcacaga

567 GTcggctgcaggtcactagtgag

568 GTGCcatctgtgcagacaaacg

775 gcggccgctgacTTAAAT

778 ggtaaatccggatTAATTAA

342 CCTTTTTACGGTTCCTGGCCT

>-gene-->

>-TP--> \ / >-TP-->

\ / \ / \ /

\ / \ / \ /

| | |

577> 775>| 468> <567 | 568> <467 |<778 <578 <342

||| ||| | ||| ||| | ||| ||| | ||| ||| |||

--------------Z-----------------A-----------------E---------------------

| r j c |

| a i o |

| I I R |

| V |

| |

--------------------- pYPKa tp gene tp ---------------------------------

Fig 18