

Guia de instalação do Python e uso do script de tratamento de curvas de fluxo

Karl Jan Clinckspoor

22 de janeiro de 2018

Sumário

1	Configuração do RheoWin	1
1.1	Exportar medidas existentes para ASCII	1
1.2	Exportar para ASCII automaticamente	2
2	Configurando o Python	4
3	Baixando o script	5
4	Instalando os pacotes necessários	6
5	Rodando o script	8

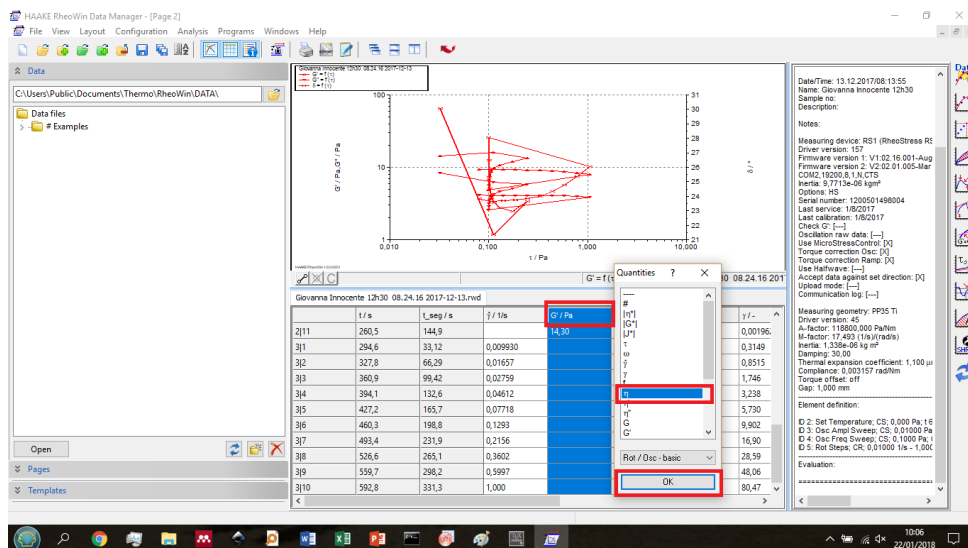
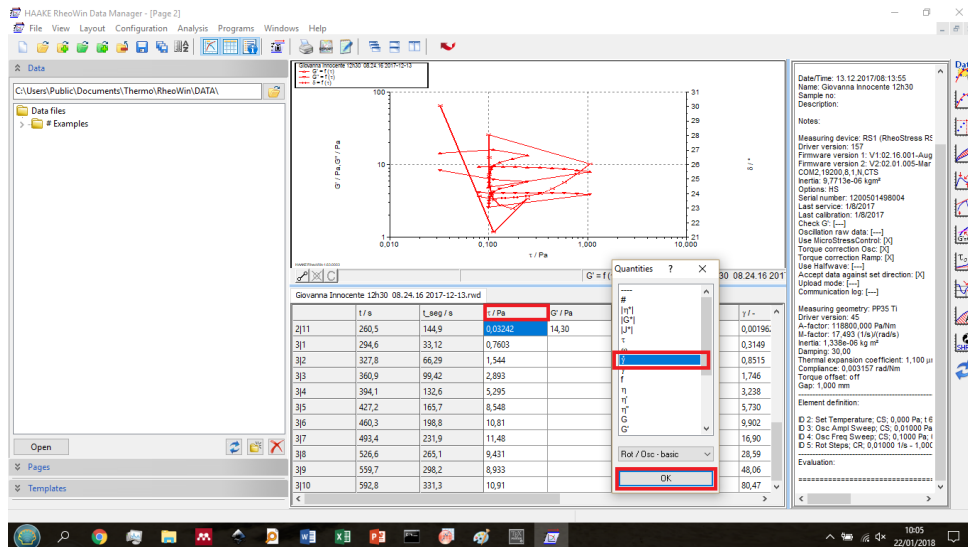
1 Configuração do RheoWin

O script necessita que os dados estejam em .txt. Isso pode ser feito de duas maneiras.

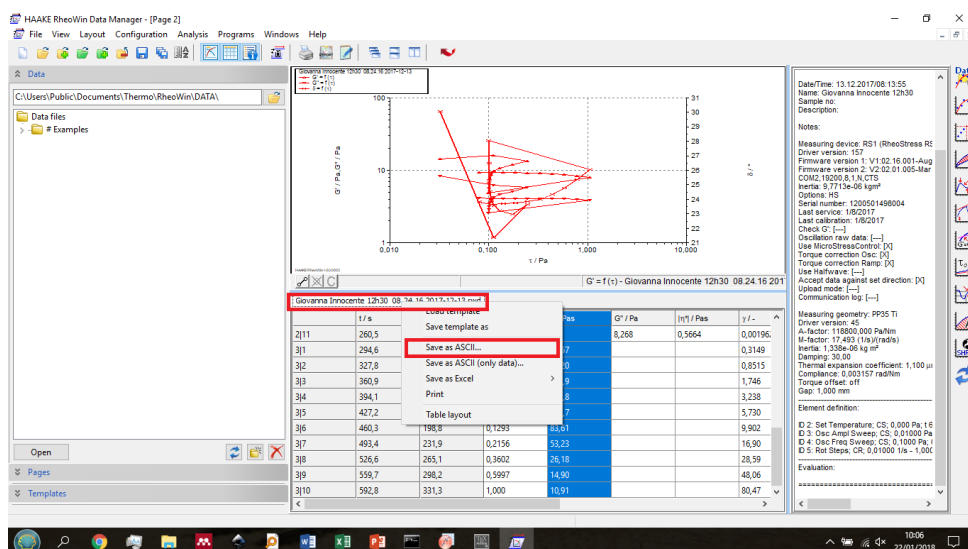
1. Exportando medidas já existentes pelo Data Manager
2. Configurando o Job Manager para automaticamente exportar os dados após a medida ter terminado

1.1 Exportar medidas existentes para ASCII

Abra um arquivo no Data Manager. Configure as colunas para mostrarem $\dot{\gamma}$ (taxa de cisalhamento) e η (viscosidade). Isso é feito clicando com o botão direito na coluna e depois escolhendo a medida relevante. Não importa a ordem ou se há medidas anteriores, desde que não haja mais de um local com valores de $\dot{\gamma}$ e η .



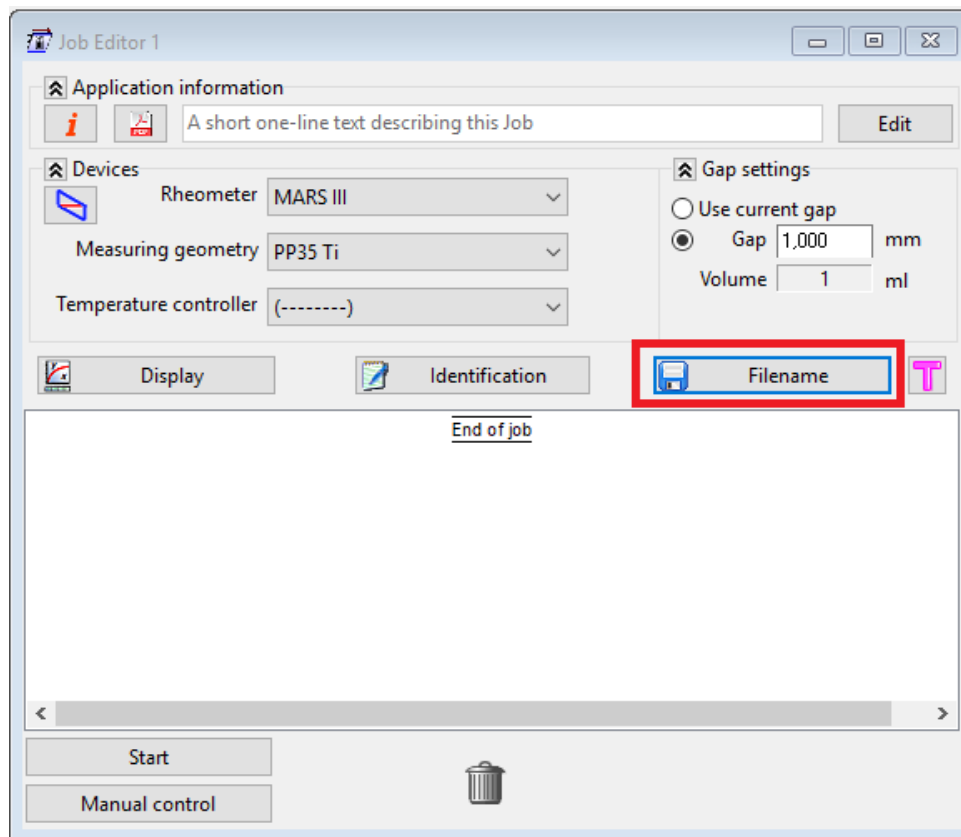
Após alterar as colunas, clique com o botão direito na aba com o nome da medida e clique em **Save to ASCII...**



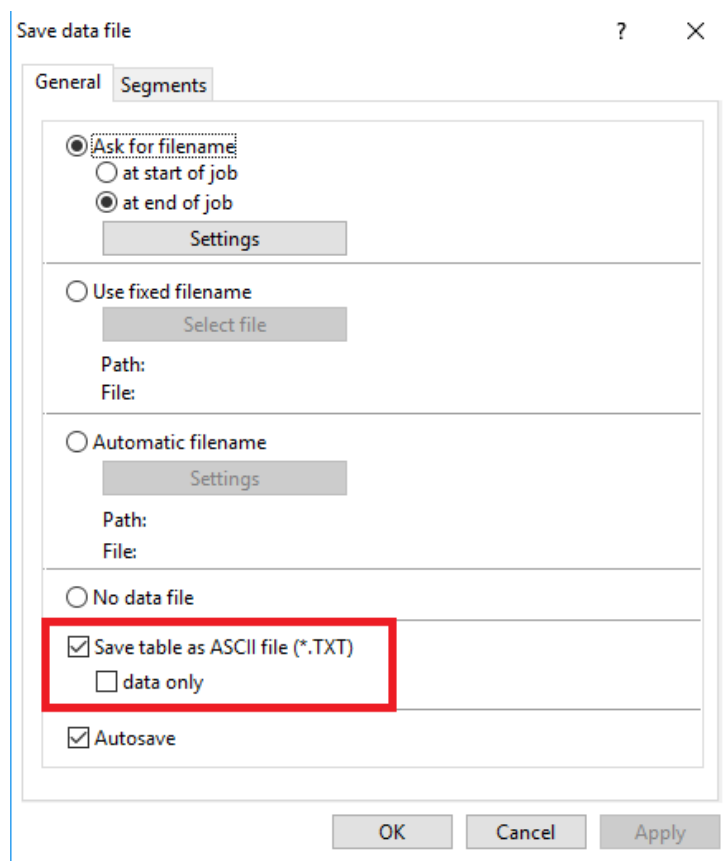
Salve num local apropriado.

1.2 Exportar para ASCII automaticamente

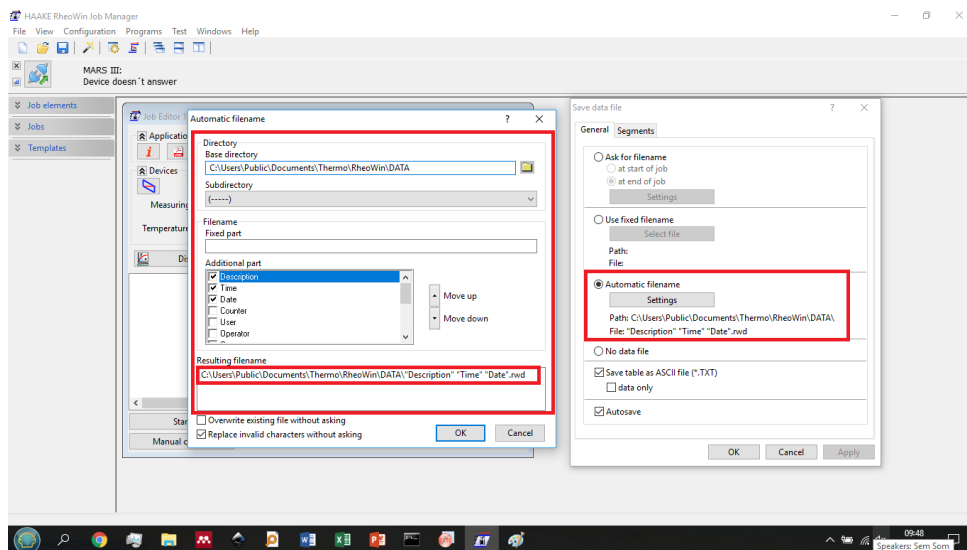
Para novas medidas, é conveniente configurar o Job Manager para automaticamente converter as medidas em ASCII depois de terminada a análise. Isso é feito clicando-se, no Job, em **Filename**.



Depois clique em Save Table as ASCII file (*.txt).

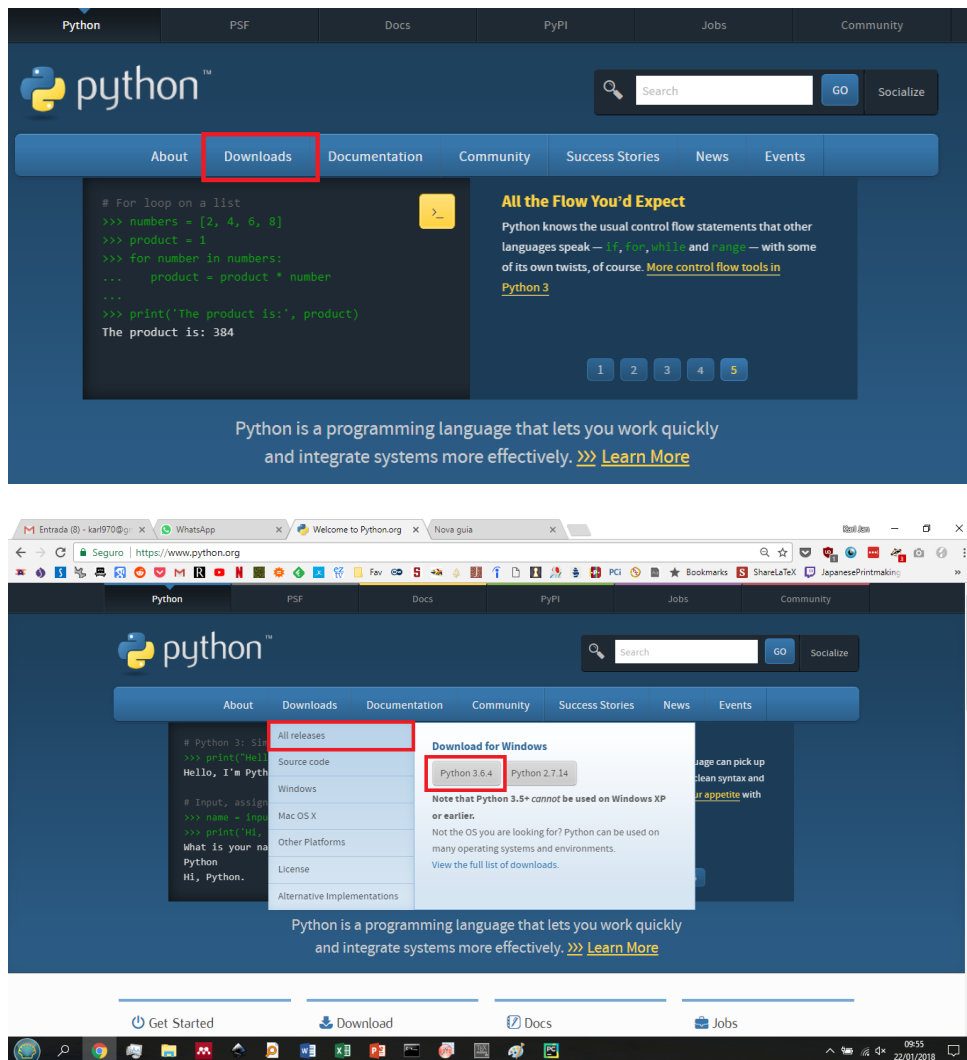


Também é possível configurar o Job Manager para dar um nome automaticamente às amostras, com o horário e a data do dia, clicando em **Automatic Filename**, escolhendo o diretório base e configurando os parâmetros que irão no nome. É possível configurar para que o nome dado à amostra no início de cada medida se tornar o nome do arquivo no final, evitando de se escrever duas vezes o nome.



2 Configurando o Python

Primeiramente, baixe o Python (3.6) do python.org. Clique em Downloads e depois selecione a versão mais nova para o Windows.



Instale o Python. Selecione a instalação customizada e certifique-se do seguinte:

- pip está sendo instalado
- python está sendo adicionado às variáveis de ambiente (*Add Python to environment variables*).

Caso tenha dúvidas, siga **esta página**.

3 Baixando o script

O script está no GitHub. Os arquivos estritamente necessários são:

- RheoFC.py

- requirements.txt

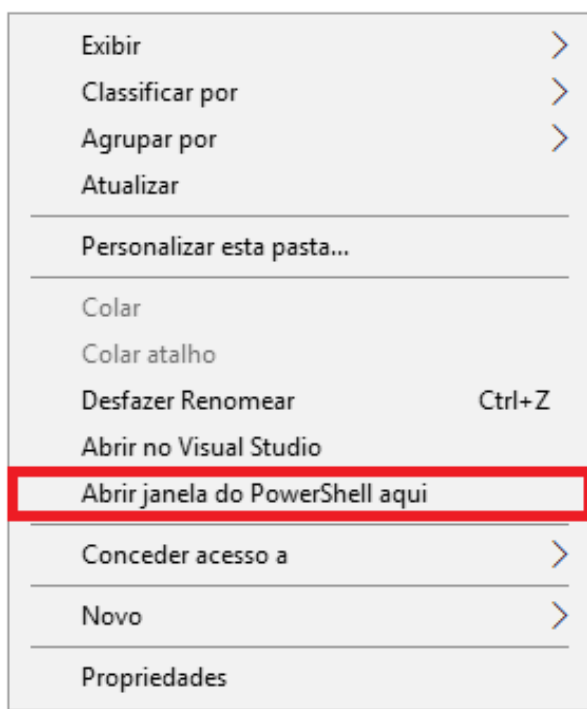
É recomendado que baixe também os arquivos:

- help
- Readme.md

Coloque esses arquivos em uma pasta específica para o script. Depois que tudo está instalado, é necessário mover o arquivo `RheoFC.py` para a pasta com os arquivos `.txt` que contém os dados.

4 Instalando os pacotes necessários

Após o programa ter instalado, abra uma nova janela de comando ou do *PowerShell* na pasta do script. Nessa pasta deverá haver o programa (`RheoFC.py`) e o arquivo `requirements.txt`. Abrir uma nova janela de comando na pasta atual pode ser feito clicando-se com o botão direito no Windows Explorer enquanto Shift está pressionado e depois clicar na opção de abrir uma janela de comando.



Outra maneira de fazer isso é abrindo Executar (Com Windows + R), digitando `cmd` ou `powershell`, e depois navegando até o diretório desejado usando comandos `cd`.

Na pasta com o arquivo `requirements.txt`, digite:

```
pip install -r requirements.txt
```

O pip começará a instalar os arquivos necessários. Caso apareça um erro, dizendo que o comando não foi reconhecido, siga as instruções contidas aqui, aqui ou aqui.

Esta é a cara do script em andamento.

```
C:\WINDOWS\system32\cmd.exe - pip install -r requirements.txt

(tutorial-env) C:\Users\Karl\Desktop\Python>pip install -r requirements.txt
Collecting scipy==1.0.0 (from -r requirements.txt (line 1))
  Downloading scipy-1.0.0-cp36-none-win_amd64.whl (30.8MB)
  100% |#####| 30.8MB 17kB/s
Collecting uncertainties==3.0.2 (from -r requirements.txt (line 2))
  Downloading uncertainties-3.0.2.tar.gz (229kB)
  100% |#####| 235kB 252kB/s
Collecting sympy==1.1.1 (from -r requirements.txt (line 3))
  Downloading sympy-1.1.1.tar.gz (4.6MB)
  100% |#####| 4.6MB 128kB/s
Collecting numpy==1.14.0 (from -r requirements.txt (line 4))
  Downloading numpy-1.14.0-cp36-none-win_amd64.whl (13.4MB)
  100% |#####| 13.4MB 54kB/s
Collecting pandas==0.20.1 (from -r requirements.txt (line 5))
  Downloading pandas-0.20.1-cp36-cp36m-win_amd64.whl (8.2MB)
  100% |#####| 8.3MB 83kB/s
Collecting matplotlib==2.0.2 (from -r requirements.txt (line 6))
  Downloading matplotlib-2.0.2-cp36-cp36m-win_amd64.whl (8.9MB)
  100% |#####| 8.9MB 93kB/s
Collecting mpmath>=0.19 (from sympy==1.1.1->-r requirements.txt (line 3))
  Downloading mpmath-1.0.0.tar.gz (511kB)
  100% |#####| 512kB 595kB/s
Collecting python-dateutil>=2 (from pandas==0.20.1->-r requirements.txt (line 5))
  Using cached python_dateutil-2.6.1-py2.py3-none-any.whl
Collecting pytz>=2011k (from pandas==0.20.1->-r requirements.txt (line 5))
  Downloading pytz-2017.3-py2.py3-none-any.whl (511kB)
  100% |#####| 512kB 819kB/s
Collecting six>=1.10 (from matplotlib==2.0.2->-r requirements.txt (line 6))
  Downloading six-1.11.0-py2.py3-none-any.whl
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=1.5.6 (from matplotlib==2.0.2->-r requirements.txt (line 6))
  Using cached pyparsing-2.2.0-py2.py3-none-any.whl
Collecting cycycler>=0.10 (from matplotlib==2.0.2->-r requirements.txt (line 6))
  Using cached cycycler-0.10.0-py2.py3-none-any.whl
Installing collected packages: numpy, scipy, uncertainties, mpmath, sympy, six, python-dateutil, pytz, pandas, pyparsing, cycycler, matplotlib
```

Esta é a transcrição de uma nova instalação pelo pip (em um ambiente virtual).

```
(tutorial-env) C:\Users\Karl\Desktop\Python>pip install -r requirements.txt
Collecting scipy==1.0.0 (from -r requirements.txt (line 1))
  Downloading scipy-1.0.0-cp36-none-win_amd64.whl (30.8MB)
  100% |#####| 30.8MB 17kB/s
Collecting uncertainties==3.0.2 (from -r requirements.txt (line 2))
  Downloading uncertainties-3.0.2.tar.gz (229kB)
  100% |#####| 235kB 252kB/s
Collecting sympy==1.1.1 (from -r requirements.txt (line 3))
  Downloading sympy-1.1.1.tar.gz (4.6MB)
  100% |#####| 4.6MB 128kB/s
Collecting numpy==1.14.0 (from -r requirements.txt (line 4))
  Downloading numpy-1.14.0-cp36-none-win_amd64.whl (13.4MB)
  100% |#####| 13.4MB 54kB/s
Collecting pandas==0.20.1 (from -r requirements.txt (line 5))
  Downloading pandas-0.20.1-cp36-cp36m-win_amd64.whl (8.2MB)
  100% |#####| 8.3MB 83kB/s
Collecting matplotlib==2.0.2 (from -r requirements.txt (line 6))
  Downloading matplotlib-2.0.2-cp36-cp36m-win_amd64.whl (8.9MB)
  100% |#####| 8.9MB 93kB/s
Collecting mpmath>=0.19 (from sympy==1.1.1->-r requirements.txt (line 3))
  Downloading mpmath-1.0.0.tar.gz (511kB)
  100% |#####| 512kB 595kB/s
Collecting python-dateutil>=2 (from pandas==0.20.1->-r requirements.txt (line 5))
  Using cached python_dateutil-2.6.1-py2.py3-none-any.whl
Collecting pytz>=2011k (from pandas==0.20.1->-r requirements.txt (line 5))
  Downloading pytz-2017.3-py2.py3-none-any.whl (511kB)
  100% |#####| 512kB 819kB/s
Collecting six>=1.10 (from matplotlib==2.0.2->-r requirements.txt (line 6))
  Downloading six-1.11.0-py2.py3-none-any.whl
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=1.5.6
  (from matplotlib==2.0.2->-r requirements.txt (line 6))
  Using cached pyparsing-2.2.0-py2.py3-none-any.whl
Collecting cycycler>=0.10 (from matplotlib==2.0.2->-r requirements.txt (line 6))
```

```

Using cached cycpler-0.10.0-py2.py3-none-any.whl
Installing collected packages: numpy, scipy, uncertainties, mpmath, sympy, six,
python-dateutil, pytz, pandas, pyparsing, cycpler, matplotlib
Running setup.py install for uncertainties ... done
Running setup.py install for mpmath ... done
Running setup.py install for sympy ... done
Successfully installed cycpler-0.10.0 matplotlib-2.0.2 mpmath-1.0.0 numpy-1.14.0
pandas-0.20.1 pyparsing-2.2.0 python-dateutil-2.6.1 pytz-2017.3
scipy-1.0.0 six-1.11.0 sympy-1.1.1 uncertainties-3.0.2

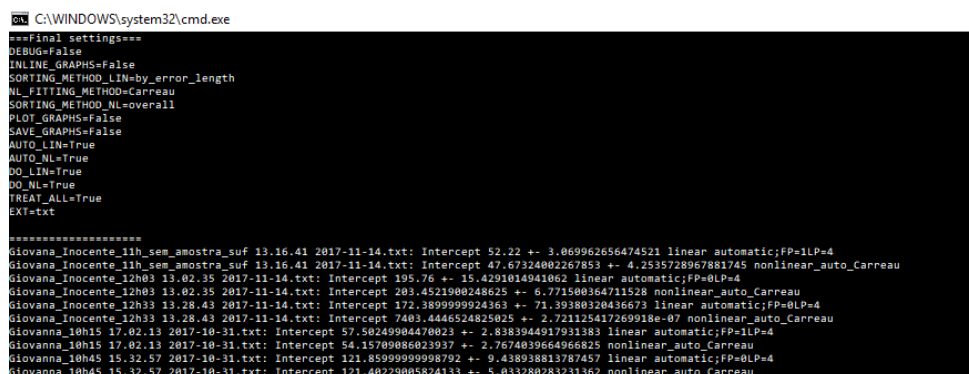
```

(tutorial-env) C:\Users\Karl\Desktop\Python>

5 Rodando o script

Após ter instalado todos os pacotes, rode o programa utilizando o comando `python RheoFC.py`. Ele irá carregar as configurações presentes, e depois você pode escolher o que deseja modificar. Quando terminar de configurar, ele iniciará o processo. Os dados do ajuste linear estarão no arquivo `linear.csv`, os dados do ajuste de Carreau no arquivo `Carreau.csv`, e assim por diante. Mais detalhes podem ser vistos no arquivo `Readme.md` (em inglês).

Exemplo:



```

C:\WINDOWS\system32\cmd.exe
===Final settings===
DEBUG=False
INLINE_GRAPHS=False
SORTING_METHOD_LIN=by_error_length
NL_FITTING_METHOD=Carreau
SORTING_METHOD_NL=overall
PLOT_GRAPHS=False
SAVE_GRAPHS=False
AUTO_LIN=True
AUTO_NL=True
DO_LIN=True
DO_NL=True
TREAT_ALL=True
EXT=txt

=====
Giovanna_Inocente_11h_ses_amostra_suf 13.16.41 2017-11-14.txt: Intercept 52.22 +- 3.069962656474521 linear automatic;FP=1LP=4
Giovanna_Inocente_11h_ses_amostra_suf 13.16.41 2017-11-14.txt: Intercept 47.67324002267853 +- 4.2535728967881745 nonlinear_auto_Carreau
Giovanna_Inocente_12h03 13.02.35 2017-11-14.txt: Intercept 195.76 +- 15.4291014941062 linear automatic;FP=0LP=4
Giovanna_Inocente_12h03 13.02.35 2017-11-14.txt: Intercept 203.4521900248625 +- 6.771500364711528 nonlinear_auto_Carreau
Giovanna_Inocente_12h33 13.28.43 2017-11-14.txt: Intercept 172.3899999924363 +- 71.39380320436673 linear automatic;FP=0LP=4
Giovanna_Inocente_12h33 13.28.43 2017-11-14.txt: Intercept 7403.4446524825025 +- 2.721125417269918e-07 nonlinear_auto_Carreau
Giovanna_10h15 17.02.13 2017-10-31.txt: Intercept 57.50249904478023 +- 2.8383944019931383 linear automatic;FP=1LP=4
Giovanna_10h15 17.02.13 2017-10-31.txt: Intercept 54.15709086023937 +- 2.7674039664966825 nonlinear_auto_Carreau
Giovanna_10h45 15.32.57 2017-10-31.txt: Intercept 121.85999999998792 +- 9.438938813787457 linear automatic;FP=0LP=4
Giovanna_10h45 15.32.57 2017-10-31.txt: Intercept 121.40229005824133 +- 5.033280283231362 nonlinear_auto_Carreau

```

Pronto, parabéns, o script está funcionando. Bom proveito.