

# **Python Pakket en GUI voor FlowSOM**

*Elias Nijs*

Computationele Biologie  
Universiteit Gent

## *ABSTRACT*

Dit verslag presenteert de ontwikkeling en implementatie van het FlowSOM package voor python en de ontwikkeling van een grafische gebruikersinterface (GUI) ervoor. Op deze manier maken we het FlowSOM algoritme toegankelijk voor zowel mensen met als zonder technische achtergrond.

11 June 2023

# Python Pakket en GUI voor FlowSOM

*Elias Nijs*

Computationale Biologie  
Universiteit Gent

## 1. Introductie

Flow cytometry is een techniek die in toenemende mate wordt gebruikt in biomedisch onderzoek. Deze techniek maakt het mogelijk om de fysische en chemische kenmerken van individuele cellen in een stroom van vloeistof te meten, wat tot inzichten in het functioneren van cellen kan leiden. Een van de grootste uitdagingen van flow cytometry is echter de analyse van de geproduceerde data, die vaak uit duizenden tot miljoenen cellen bestaat, elk met meerdere gemeten parameters. Het is een hoogdimensionale dataset, en traditionele analysetechnieken zijn vaak ontoereikend.

In deze context is het FlowSOM-algoritme<sup>1</sup> ontwikkeld. FlowSOM maakt gebruik van Self-Organizing Maps (SOM) en meta clustering om de data te visualiseren en interpretatie mogelijk te maken. Het heeft zich bewezen als een hulpmiddel voor flow cytometry data-analyse. Het blijft echter een uitdaging voor onderzoekers met een minder technische achtergrond om dit algoritme te gebruiken. Daarnaast gebeurt de computationele biologie steeds vaker in python. Het huidige flowsom pakket is echter geschreven voor de taal R.

Het doel van dit werk is om zowel een python pakket als grafische gebruikersinterface (GUI) voor het FlowSOM-algoritme te implementeren. Op deze manier kan de toegankelijkheid en de gebruikerservaring van dit algoritme aanzienlijk verbeteren. Daarnaast kan het de drempel verlagen voor onderzoekers om het FlowSOM-algoritme te gebruiken.

In dit verslag bespreken we onze aanpak voor het implementeren van de GUI, de technische uitdagingen die we hebben overwonnen, en de snelheid en efficiëntie die we hebben bereikt. Tot

slot bespreken we ook mogelijke uitbreidingen voor het pakket en de GUI.

## 2. Methoden

De methodologie van deze studie valt uiteen in twee componenten: de implementatie van het FlowSOM python pakket en de implementatie van de bijbehorende grafische gebruikersinterface (GUI).

### 2.1. FlowSOM Implementatie

De FlowSOM implementatie wordt beschikbaar gesteld in een klasse genaamd FlowSOM, een uitbreiding van de *BaseEstimator* en *ClusterMixin* klassen. Deze klasse bevat een tal aan functies voor het trainen, voorspellen en visualiseren van de data, alsook een functie om een verslag te genereren. Daarnaast houdt het object ook de relevante data zoals de parameters en de tussenresultaten van het trainen bij.

De hoofd functie van het FlowSOM object is de *fit* methode. Dit is de methode die een model zal trainen. Dit trainen bestaat uit drie delen. Het eerste deel is het trainen van de SOM. Hiervoor maken we gebruik van *Minisom*.<sup>2</sup> Ten tweede bouwen van de minimum spanning. Hiervoor maken we gebruik van *SciPi*.<sup>3</sup> Tot slot construeren we de metaclustering. Hiervoor maken we gebruik van Agglomerative clustering van *scikit-learn*.<sup>4</sup> Het resultaat van elk van deze delen wordt ook intern opgeslaan in het object en kan de gebruiker dus later nog verder gebruiken.

We raden aan het Flowsom object op volgende manier te gebruiken:

```
model = FlowSOM(  
    som_param,  
    mst_param,  
    hcc_param).fit(data)
```

We zien hierbij twee zaken: de initialisatie van het object en het oproepen van de *fit* methode. De argumenten van de initialisatie zijn dataklassen die de parameters van elk deel van het algoritme bevatten. Er werd gekozen om deze in dataklassen te steken om de parameters overzichtelijk te structureren. De parameters van de fit functie zijn: een pandas dataframe die de cytometrie data bevat, een optie om logging aan te zetten met `verbose=True`.

De functionaliteit van de FlowSOM-klasse strekt zich verder uit tot data- en visualisatiefuncties waaronder functies om de minimum spanning tree met of zonder metacusters te plotten, een functie om het som grid met of zonder metaclusters te plotten, ... Voor een globaal overzicht van de functies en hun mogelijke parameters verwijzen we naar de meegeleverde documentatie.

Het FlowSOM-pakket omvat ook een uitgebreide rapport functie, deze is in staat is een verslag van de analyse te genereren in markdown syntax. Dit rapport bevat een samenvatting van de FlowSOM-analyse waaronder de model parameters, visualisaties en numerisch resultaten voor elke node en elke metacluster. Er werd gekozen dit in markdown te voorzien zodat de gebruiker kan kiezen in welke stijl hij het verslag wilt renderen.

De implementatie voor het FlowSOM pakket is terug te vinden onder:

1. `src/flowsom/flowsom.py`,
2. `src/flowsom/fs_dataclasses.py`,
3. `src/flowsom/fs_plotting.py`,
4. `src/flowsom/fs_reporting.py`

Hierbij zijn tal van functies geschreven in de `fs_...` bestanden. De functies in deze geschreven, zijn ofwel hulpfuncties, ofwel functies die in het flowsom object, dat zich in `flowsom.py` bevindt, omwikkeld worden. De zaken die in het pakket publiek gesteld worden zijn het FlowSOM object uit `flowsom.py` en de parameter klassen uit `fs_dataclasses.py`.

## 2.2. FlowSOMTool: De GUI

De tweede component van onze methodologie betreft de ontwikkeling van FlowSOMTool, een GUI voor FlowSOM. De GUI streeft ernaar de interactie met het FlowSOM-algoritme zo eenvoudig mogelijk te maken zodat mensen met minder technische achtergrond ook het algoritme

kunnen gebruiken. Deze GUI is een prototype en is bedoeld als platform waar meer geavanceerdere interactie mogelijkheden kunnen op verder gebouwd worden.

FlowSOMTool biedt gebruikers een aantal aan mogelijkheden. Ten eerste geeft het de gebruiker de optie om FlowSOM-bestanden te selecteren op interactieve wijze. Daarnaast stelt het de gebruikers in staat om de parameters voor het algoritme aan te passen, het algoritme uit te voeren, en de resultaten te exporteren voor verdere analyse. Een belangrijk onderdeel van de GUI is het vermogen om de gegenereerde plots te visualiseren. Hierdoor krijgen gebruikers de mogelijkheid om de FlowSOM-analyse op een visueel aantrekkelijke manier te inspecteren.

Het programma is gebouwd aan de hand van de volgende technologieën:

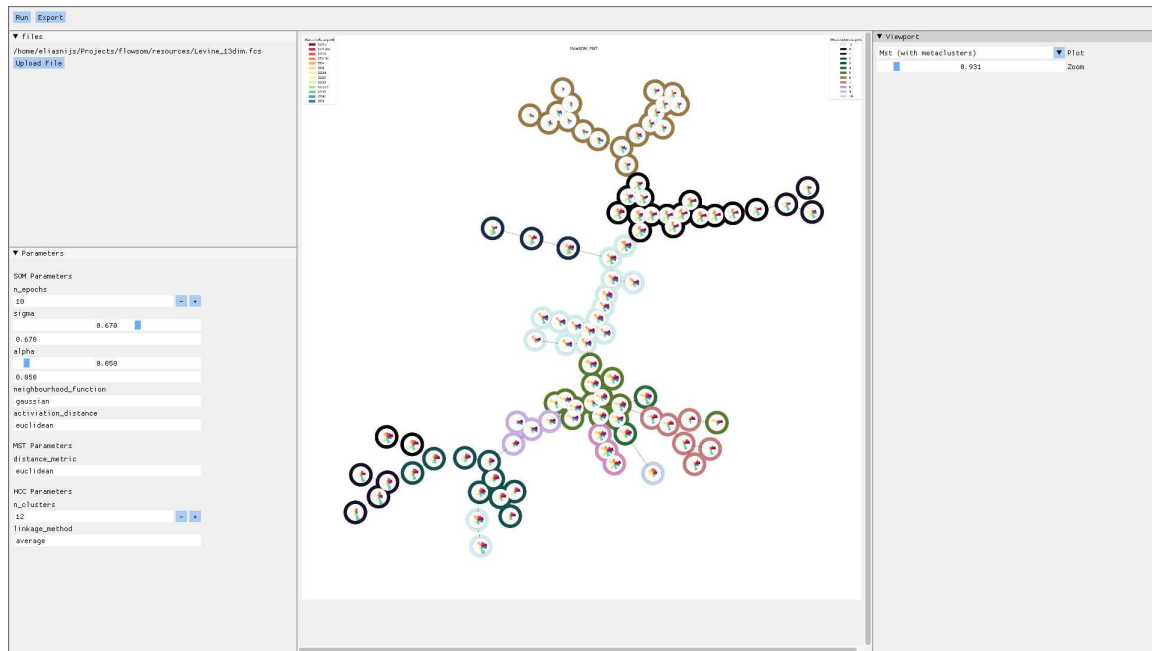
1. ImGui<sup>5</sup>
2. SDL2<sup>6</sup>
3. OpenGL<sup>7</sup>

Het bestaat uit initialisatie, programmalus en opkuis. Tijdens de initialisatie worden de contexten voor ImGui, SDL, OpenGL en Flowsom opgezet. Tijdens de programma lus worden telkens de contexten geupdate, de imgui windows opgezet en tot slot alles weergegeven.

De implementatie voor de FlowSOMTool is terug te vinden onder:

1. `src/flowsomtool.py`,
2. `src/fst_dataclasses.py`,
3. `src/fst_utils.py`,
4. `src/fst_windows.py`

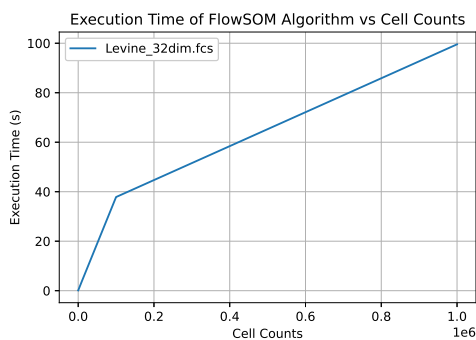
Het lanceren van FlowSOMTool, kan met het script in de `scripts/` folder genaamd `flowsom-tool.sh`.



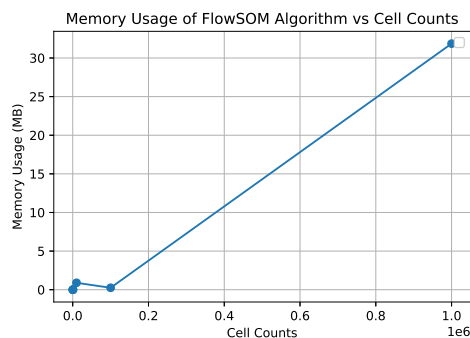
### 3. Resultaten

In deze sectie bespreken we de prestaties van onze implementatie van het FlowSOM python pakket.

We doen eerst een tijdsmeting. We houden hierbij het aantal epochs op 10. We bekomen dan het volgende resultaat:



We bekijken ook het geheugen gebruik:



We zien bij beide een lineair verband tussen de grootte van de input en tijd/geheugengebruik.

### 4. Conclusie

In dit werk hebben we het FlowSOM-pakket geïmplementeerd in Python en een grafische gebruikersinterface (GUI), genaamd FlowSOMTool, ontwikkeld om de toegankelijkheid van het algoritme te vergroten. De ontwikkeling van het Python-pakket is van bijzonder belang, gezien het groeiende gebruik van Python in de computationele biologie, en het feit dat het oorspronkelijke FlowSOM-pakket alleen beschikbaar was in R.

Het FlowSOM Python-pakket is geïmplementeerd als een klasse met een reeks functies voor het trainen, voorspellen, visualiseren van de data en

het genereren van rapporten. Daarnaast is het opgebouwd om een gebruiksvriendelijke en gestructureerde interface te bieden, terwijl het de onderliggende complexiteit van het algoritme maskeert. De FlowSOMTool GUI maakt interactie met het algoritme mogelijk zonder de noodzaak van programmeren, wat bijzonder gunstig is voor gebruikers zonder een technische achtergrond.

Uit de resultaten blijkt dat onze implementatie schaalbaar is en zich efficiënt gedraagt, met een lineair verband tussen de grootte van de input en de tijd/geheugengebruik. Dit is van cruciaal belang voor flow cytometry data, die vaak uit duizenden tot miljoenen cellen bestaat.

We stellen voor om in de toekomst verdere optimalisatie van het pakket en uitbreiding van de GUI te overwegen, om een nog robuuster en veelzijdiger instrument te creëren voor onderzoekers in flow cytometry. We zouden bijvoorbeeld het highlighten van bepaalde clusters kunnen ondersteunen in de GUI.

In essentie hebben we met succes een bruikbaar FlowSOM-pakket en GUI ontwikkeld, die samen een belangrijke bijdrage leveren aan het overbruggen van de kloof tussen geavanceerde data-analyse en gebruikers met uiteenlopende niveaus van technische expertise.

## References

1. Van Gassen, Sofie, Callebaut, Britt, Van Helden, Mary J., Lambrecht, Bart N., De-meester, Piet, Dhaene, Tom, and Saeys, Yvan, "FlowSOM: Using self-organizing maps for visualization and interpretation of cytometry data," *Cytometry Part A* **87**(7), pp. 636-645 (2015).
2. Giuseppe Vettigli, *MiniSom: minimalistic and NumPy-based implementation of the Self Organizing Map* (2023).
3. SciPy developers, *SciPy* (2023). Python library for scientific computing including implementation of minimum spanning tree algorithms.
4. scikit-learn developers, *scikit-learn* (2023). Machine learning library in Python including implementation of Agglomerative Clustering.
5. *PyImGUI contributors* (2023). Python binding for the Dear ImGui C++ GUI library.
6. PySDL2 contributors, *PySDL2* (2023). Python bindings for SDL2 C library.
7. PyOpenGL contributors, *PyOpenGL* (2023). Python bindings for OpenGL 3D graphics library.