

# Pattes de mouches: une modélisation de biophysique utilisant python scientifique

Logo DamCB

**Guillaume Gay**

**DamCB** - Data analysis and modeling for Cell Biology

**e-mail:** [guillaume@damcb.com](mailto:guillaume@damcb.com) (<mailto:guillaume@damcb.com>)

**github / freenode:** glyg

**twitter:** @damcellbio / @elagachado

On travaille sur la mouche de vinaigre, drosophila Melanogaster:



source: wikipedia

Drosophila development

On regarde la formation du petit pli entre les segments 4 et 5...

Vue de *très* près ça ressemble à ça:

Vue apicale

# Le problème biologique: comment la mouche plie ses pattes?

## L'apoptose, ou les cellules kamikases

- Mort **programmée** *via* une cascade biochimique.
  - Nombreuses fonctions:
    - comme la prévention du cancer
    - et la **morphogénèse**
- 

Un exemple de *phénotype* quand il y a un défaut d'apoptose



## Dans le tissu qui donnera la patte

Pour former un pli, ~ 30 apoptoses en anneau autour de l'épithélium



Apical vue of the fold formation on a drosophila leg disk (<http://vimeo.com/109897311>) from glyg (<http://vimeo.com/user12210065>) on Vimeo (<https://vimeo.com>).

Et lorsqu'on supprime l'apoptose...

Des pattes trop droites

Regardons ce qu'il se passe à l'échelle cellulaire:

L'apoptose dans la cellule vivante

## **Le problème biophysique**

Les déformations du tissu sont dues à l'activité d'un ensemble de protéines intra et extra cellulaires.

Le modèle de Farhadifar

Ces protéines forment un maillage entre les cellules, que l'on peut modéliser comme un réseau

Architecture du réseau pour une cellule

- Minimiser localement l'énergie à chaque modification
- Équilibre des forces

Le modèle est donc basé sur la bibliothèque `graph_tool` écrite et maintenue par Tiago P. Peixoto, permettant de manipuler des graphes.



graph-tool | Efficient network analysis

Documentation

Mailing List

Git

Issues

## What is graph-tool?

Graph-tool is an efficient [Python](#) module for manipulation and statistical analysis of [graphs](#) (a.k.a. [networks](#)). Contrary to most other python modules with similar functionality, the core data structures and algorithms are implemented in [C++](#), making extensive use of template [metaprogramming](#), based heavily on

**Download version 2.2.35** 

[See Instructions](#) | [See Changelog](#)

Voici un exemple rapide de cette bibliothèque

In [3]:

```
g = gt.price_network(3000)
pos = gt.sfdp_layout(g)

pos = gt.graph_draw(g, pos=pos, output="graph-draw-sfdp.png")
```

A graph generated by graph-tool

## Les avantages de graph-tool:

1. C'est rapide (C++ / meta-programming, openMP)
2. PropertyMaps interfacées aux ndarray de NumPy.
3. Bonnes E/S
4. Mécanisme de filtrage (*via* des PropertyMaps booléennes appliquées sur le graphe)
5. Bibliothèque complète d'analyse de graphes.

## Le module `leg_joint`

- Objet `Epithelium`:
  - graphe orienté, masques
  - géométrie
  - méthodes de base
- Fonctions:
  - Division cellulaire, apoptose ...
  - Optimisations
  - Représentation graphiques

In [4]: *### Create an epithelium by instanciating the container class*

```
eptm = lj.Epithelium(lj.data.before_apoptosis_xml(),  
                    save_dir='.',  
                    identifier='slides',  
                    copy=True)
```

```
### Scale the tissue globally to approach equilibrium  
eptm.isotropic_relax()
```

A simulated tissue

## **Un exemple: la division cellulaire**

```
In [5]: ### Select cell a vertex

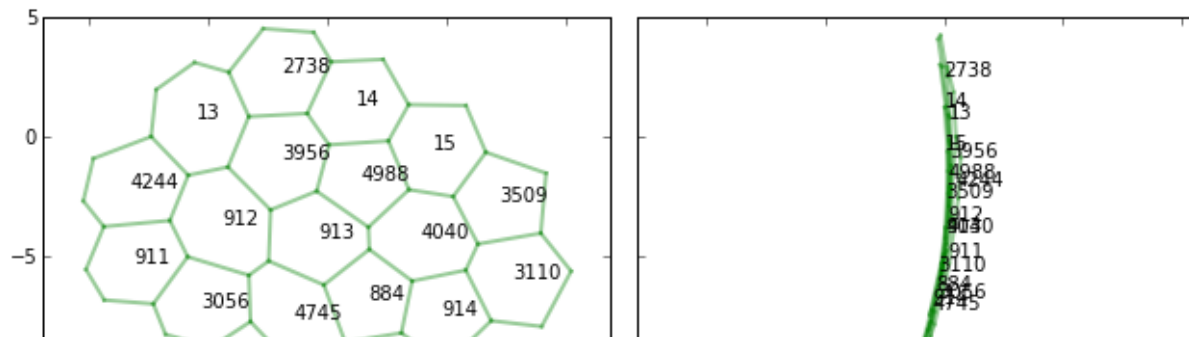
mother_cell = eptm.graph.vertex(913)
print('The vertex {} is a cell: {}'.format(
    mother_cell, bool(eptm.is_cell_vert[mother_cell])))

### Local mask is used to work only on part of the epithelium
eptm.set_local_mask(None)
eptm.set_local_mask(mother_cell, wider=True)
fig, axes=plt.subplots(1, 2, figsize=(9, 4), sharey=True)
axes = lj.plot_2panels(eptm,
                       cell_kwargs={'c_text':True},
                       edge_kwargs={'c':'g', 'lw':2, 'alpha':0.4},
                       axes=axes)
```

The vertex 913 is a cell: True

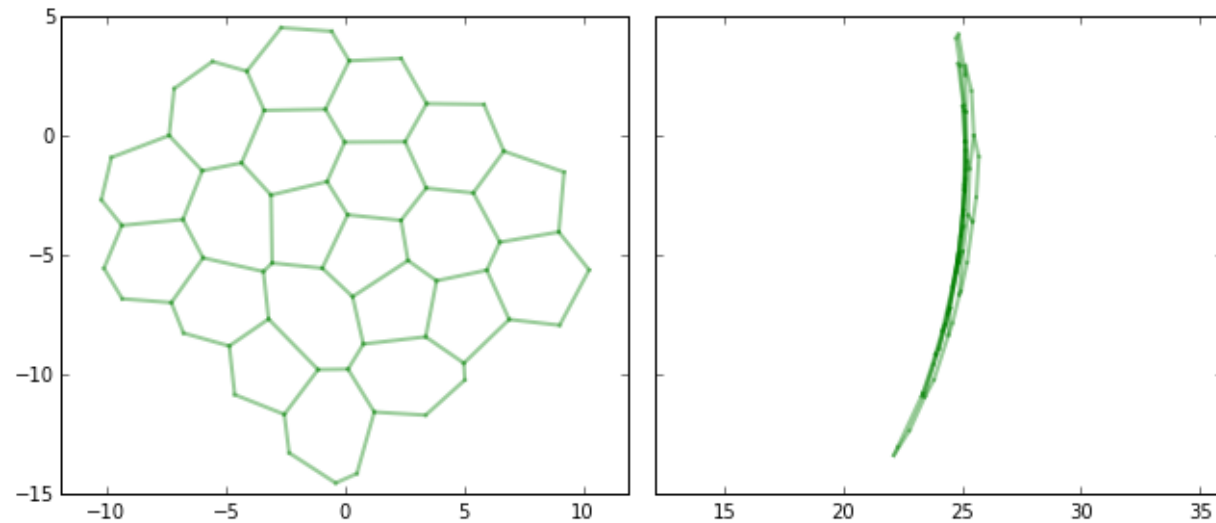
/home/guillaume/anaconda/envs/python3/lib/python3.4/site-packages/matplotlib-1.4.x-py3.4-linux-x86\_64.egg/matplotlib/figure.py:1644: UserWarning: This figure includes Axes that are not compatible with tight\_layout, so its results might be incorrect.

warnings.warn("This figure includes Axes that are not "



```
In [6]: septum = lj.cell_division(eptm, mother_cell, verbose=False)

### Gradient descent energy minimization (`fmin_lbfgs_b` is used)
pos0, pos1 = lj.find_energy_min(eptm)
fig, axes=plt.subplots(1, 2, figsize=(9, 4), sharey=True)
axes = lj.plot_2panels(eptm,
                      cell_kwargs={'c_text':False},
                      edge_kwargs={'c':'g', 'lw':2, 'alpha':0.4},
                      axes=axes)
```



On choisit une cellule apoptotique

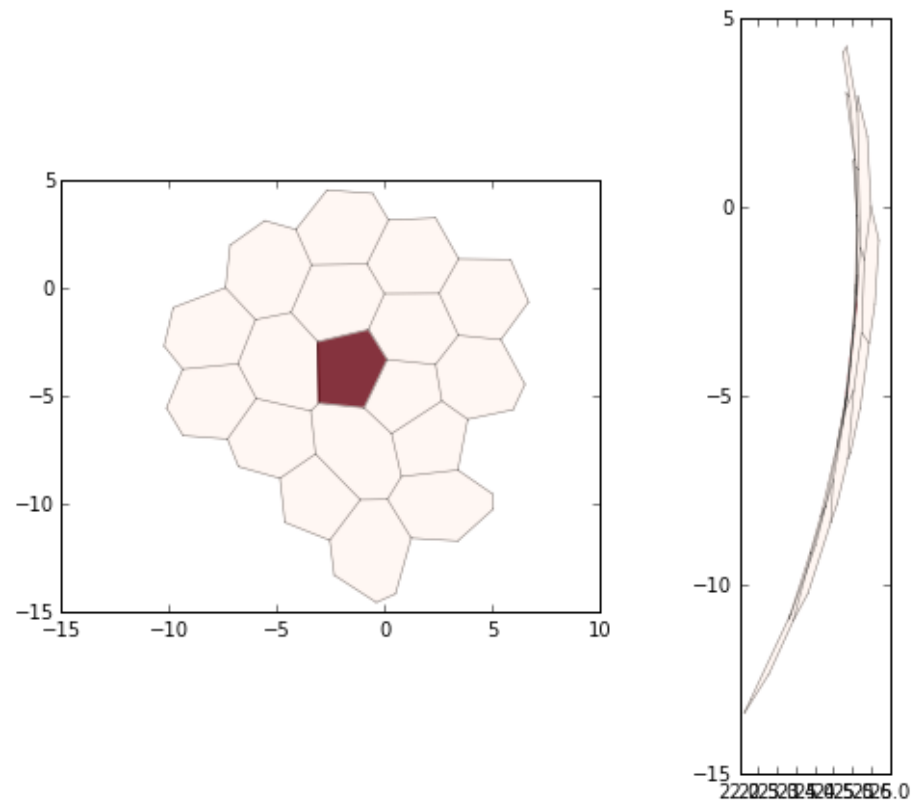
```

In [8]: fig, axes = plt.subplots(1, 2, figsize=(8, 6))

a_cell = mother_cell
eptm.set_local_mask(None)
eptm.set_local_mask(a_cell, wider=True)

ax_zs, ax_xy = show_apopto_surroundings(eptm, a_cell, axes=axes)

```



À 10 reprises, on diminue son volume d'équilibre, et on tire dessus avec une force radiale croissante

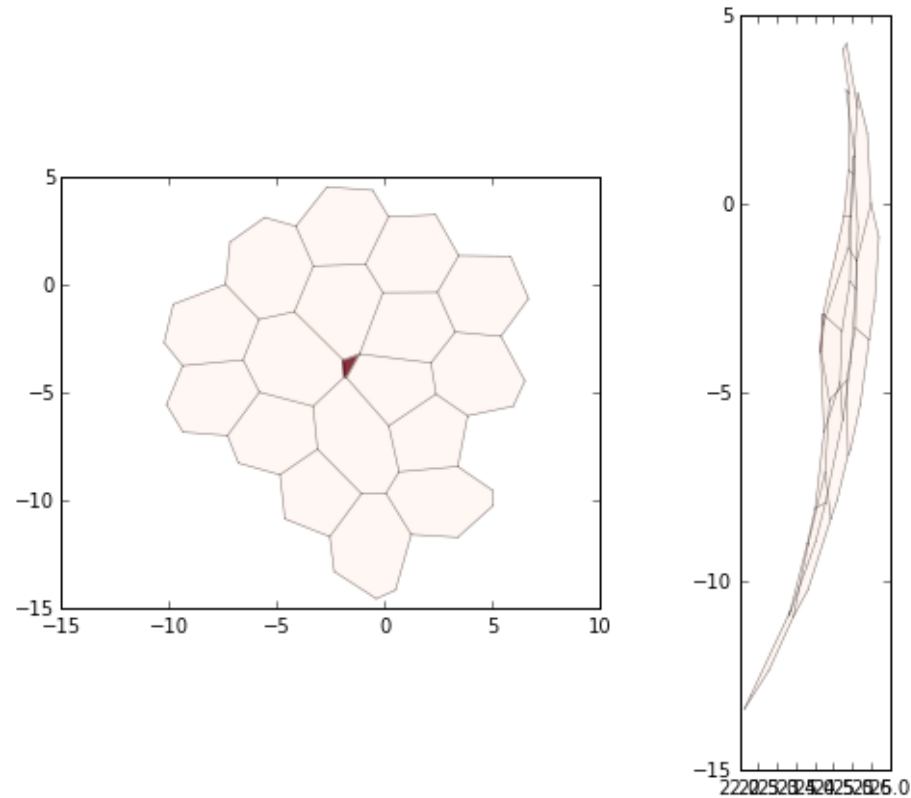
```

In [9]: fig, axes = plt.subplots(1, 2, figsize=(8, 6))

for i in range(10):
    lj.apoptosis_step(eptm, a_cell,
                      vol_reduction=0.7, ## Reduction of equilibrium volume
                      radial_tension=0.1, ## Increase in radial force
                      contractility=1.2, ## Increase in contractility for the apoptotic cell
                      )

ax_zs, ax_xy = show_apopto_surroundings(eptm, a_cell, axes)

```

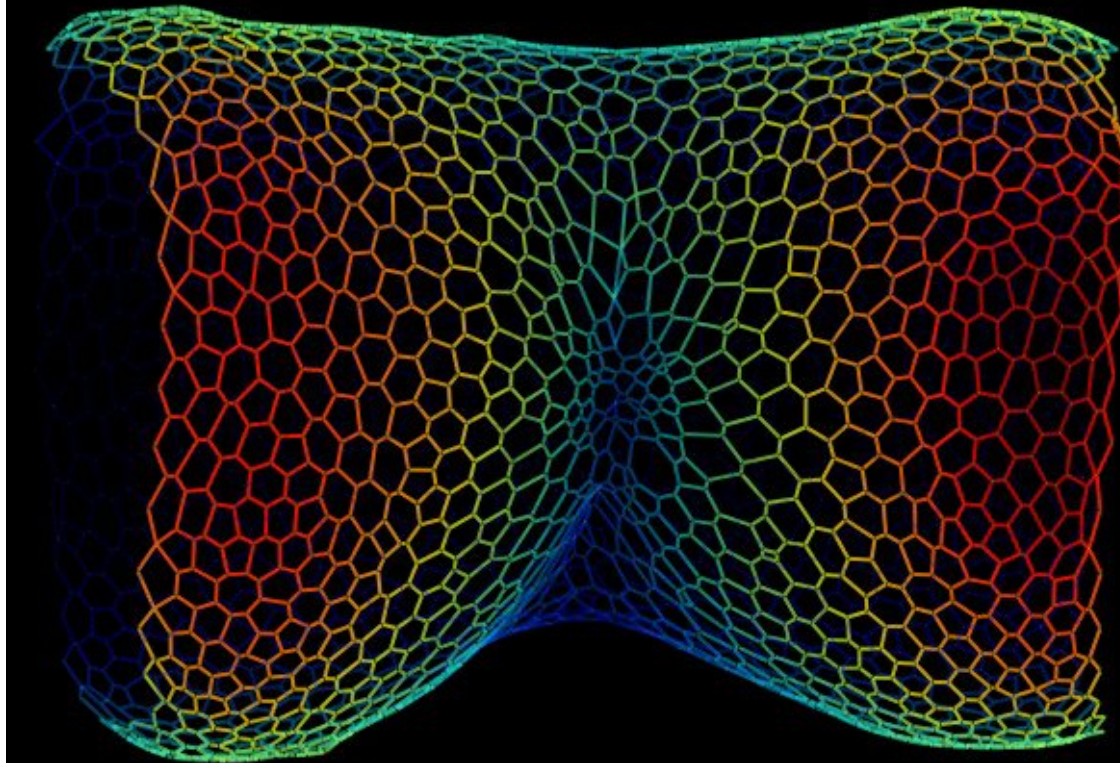


Lorsqu'on fait ça 30 fois, voilà ce qu'il se passe:



Fold formation model

glyg



00:40

| HD

Fold formation model (<http://vimeo.com/107188046>) from glyg (<http://vimeo.com/user12210065>) on Vimeo (<https://vimeo.com>).

Différentes conditions

Pas de force  $\rightarrow$  pas de pli:

**on confirme l'hypothèse de la nécessité d'un rôle actif de l'apoptose dans la morphogénèse**

## Reste à faire

1. Tests unitaires & intégration continue
2. Rendu dynamique 3D avec Blender (en cours)
3. Gros réusinage du code:
  - description basée sur des événements
  - optimiser (y'a du boulot...)

## Et ensuite

1. Généraliser la géométrie, surface basale
2. Plateforme de modélisation de tissus "multi-physique"

## **Les biologistes du LBCMCP (CNRS/Université de Toulouse)**

- Mélanie Gettings
- Bruno Monier
- Sonia Shott
- **Magali Suzanne**

## **L'autre physicien**

- Thomas Mangeat

**Merci!**