



Un package Python open-source et une ressource en ligne
pour faciliter la recherche en biomécanique

Félix Chénier, PhD

- Professeur, Université du Québec à Montréal (UQAM)
 - Chercheur, Centre de recherche interdisciplinaire en réadaptation du Montréal métropolitain (CRIR)
-



1

Traitement de données biomécaniques

- Logiciels très puissants pour traiter et visualiser des données biomécanique :



etc.

- mais parfois:

- Trop complexes pour nos besoins;
- Manque de versatilité (e.g., markerset, conventions d'angles);
- Pas encore de gold-standard, donc pas de module pour réaliser ce qu'on veut;
- Les données du laboratoire ne sont pas au même format que les données attendues par les logiciels.
- etc.

2

Historique de KTK

3

Besoin

Analyser des données
brutes et de développer
des algorithmes maison

4

Besoin

Analyser des données brutes et de développer des algorithmes maison



LIO

Laboratoire d'innovation ouverte
en technologies de la santé

2012-



Algorithmes de traitement de données

Traçabilité des données brutes

Traçabilité des traitements de données

Traçabilité des données traitées

5

Besoin

Analyser des données brutes et de développer des algorithmes maison

Cibler le traitement de données et s'affranchir des pratiques de laboratoire



Laboratoire de recherche en
**MOBILITÉ ET
SYSTÈMES ADAPTÉS**

2014-2019



Algorithmes de traitement de données

Traçabilité des données brutes

Traçabilité des traitements de données

Traçabilité des données traitées

6

Besoin

Analyser des données brutes et de développer des algorithmes maison

Cibler le traitement de données et s'affranchir des pratiques de laboratoire

Accessibilité et partage : Utilisation d'outils ouverts



Laboratoire de recherche en
**MOBILITÉ ET
SYSTÈMES ADAPTÉS**

2019-



Package Python en code ouvert



```
pip install kineticstoolkit
conda install kineticstoolkit
```

7

Besoin

Analyser des données brutes et de développer des algorithmes maison

Cibler le traitement de données et s'affranchir des pratiques de laboratoire

Accessibilité et partage : Utilisation d'outils ouverts

Formation : cibler la connaissance, éviter les boîtes noires.



Laboratoire de recherche en
**MOBILITÉ ET
SYSTÈMES ADAPTÉS**

2019-



Package Python en code ouvert



```
pip install kineticstoolkit
conda install kineticstoolkit
```

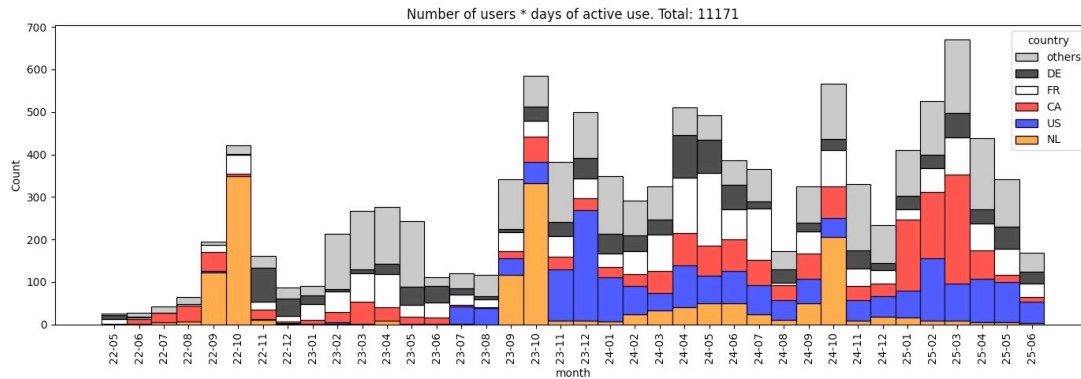
Livre électronique



Biomechanical Analysis using Python and
Kinetics Toolkit

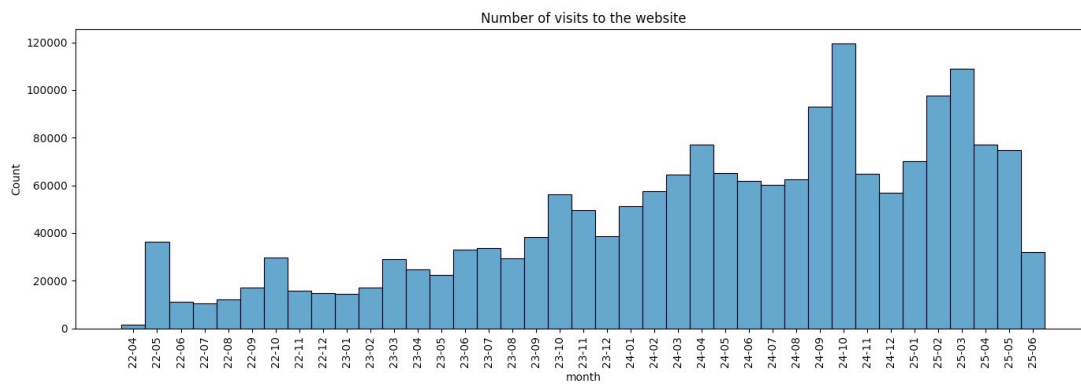
8

(utilisateurs × jours) du package **kineticstoolkit**



9

Nombre de visites sur **kineticstoolkit.uqam.ca**



10

Python pour la biomécanique

11

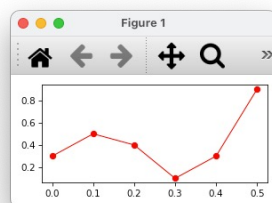
Apprendre Python pour la biomécanique



- Variable
- Arithmétique
- Conditions
- Boucles
- etc.



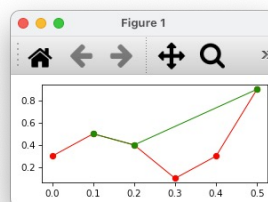
```
import matplotlib.pyplot as plt
time = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5]
data = [0.3, 0.5, 0.4, 0.1, 0.3, 0.9]
plt.plot(time, data, 'o-')
```



```
import matplotlib.pyplot as plt
import numpy as np
time = np.array([0.0, 0.1, 0.2, 0.3, 0.4, 0.5])
data = np.array([0.3, 0.5, 0.4, 0.1, 0.3, 0.9])

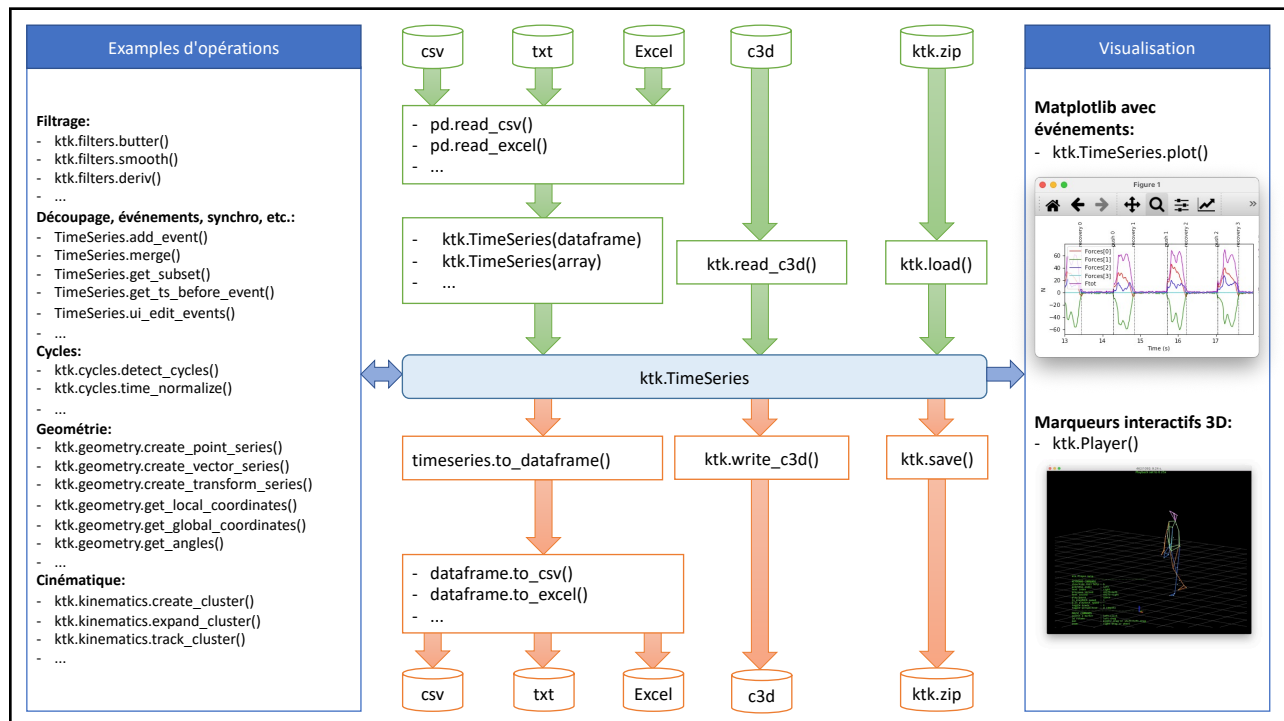
# Keep every data higher than 0.4
new_time = time[data >= 0.4]
new_data = data[data >= 0.4]

plt.plot(time, data, 'o-')
plt.plot(new_time, new_data, 'o-')
```



```
import pandas as pd
df = pd.read_csv("file.csv")
# Some processing
df.to_excel("file.xlsx")
```

12



1. Fondements

- Qu'est-ce qu'une TimeSeries
- Ajouter des données dans une TimeSeries
- Rééchantillonner une TimeSeries
- Ajouter et éditer des événements
- Exporter vers Excel

2. Roues instrumentées

- Analyse spatiotemporelle et cinétique de la propulsion en fauteuil roulant :
 - Enlever les offsets dûs au poids de la main courante
 - Calculer la force totale, la vitesse et la puissance
 - Isoler les poussées manuellement et automatiquement
 - Analyse poussée par poussée
 - Exporter vers Excel
- Analyse cinétique de la propulsion en fauteuil roulant
 - Afficher le cycle de propulsion moyen (F_{tot} , M_z)

15

3. Cinématique



16

<https://kineticstoolkit.uqam.ca/doc/geometry.html>

3. Cinématique

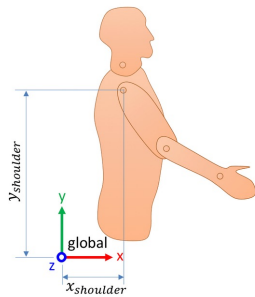


Fig. 9.2 A global coordinate system.

$${}^{\text{global}}p_{\text{shoulder}} = \begin{bmatrix} x_{\text{shoulder}} \\ y_{\text{shoulder}} \\ z_{\text{shoulder}} \\ 1 \end{bmatrix}$$

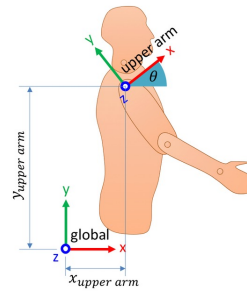


Fig. 9.4 Expressing the position and orientation of the upper arm.

$${}^{\text{global}}_{\text{upper arm}}T = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & x_{\text{upper arm}} \\ \sin(\theta) & \cos(\theta) & 0 & y_{\text{upper arm}} \\ 0 & 0 & 1 & z_{\text{upper arm}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

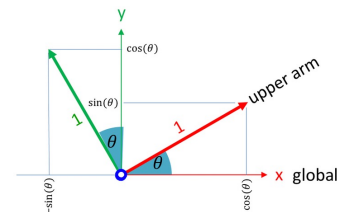


Fig. 9.5 Orientation of the upper arm coordinate system (bold lines) in reference to the global coordinate system (thin lines).

17

<https://kineticstoolkit.uqam.ca/doc/geometry.html>

3. Cinématique

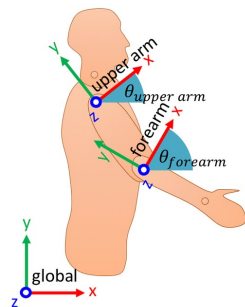
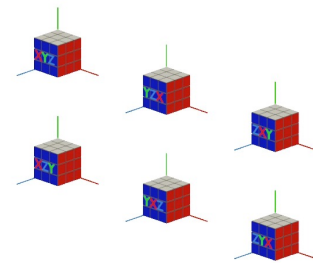
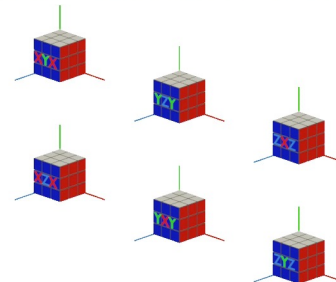


Fig. 9.6 Local coordinates for both the upper arm and the forearm.

$${}^{\text{global}}p_{\text{wrist}} = ({}^{\text{global}}_{\text{upper arm}}T)({}^{\text{upper arm}}_{\text{forearm}}T)({}^{\text{forearm}}p_{\text{wrist}})$$



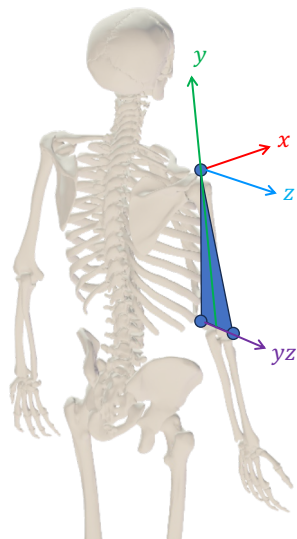
The six sequences of intrinsic Cardan rotations (intrinsic = moving coordinate system)



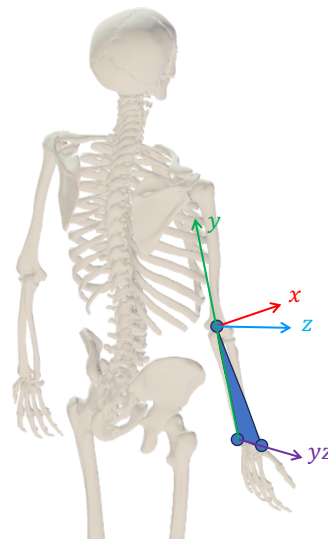
The six sequences of intrinsic Euler rotations (intrinsic = moving coordinate system)

<https://kineticstoolkit.uqam.ca/doc/kinematics.html>

3. Cinématique



1. y
2. $x = y \times yz$
3. $z = x \times y$



1. y
2. $x = y \times yz$
3. $z = x \times y$

19

3. Cinématique

- Fondements de géométrie dans KTK
- Ouvrir un fichier C3D
- Visualiser les points
- Créer le système d'axes du bras et de l'avant-bras
- Déterminer les angles d'Euler pour passer du bras à l'avant-bras
- Afficher une courbe des angles de flexion/*carrying*/pronation

20

JOS
The Journal of Open Source Software

Kinetics Toolkit: An Open-Source Python Package to Facilitate Research in Biomechanics

Félix Chénier^{1, 2}

¹ Department of Physical Activity Sciences, Université du Québec à Montréal (UQAM), Montreal, Canada ² Mobility and Adaptive Sports Research Lab, Centre for Interdisciplinary Research in Rehabilitation of Greater Montreal (CRIR), Montreal, Canada

DOI: [10.21105/joss.03714](https://doi.org/10.21105/joss.03714)

Software

- [Review it](#)
- [Repository it](#)
- [Archive it](#)

Editor: [Marie E. Rognes](#)

Reviewers:

- [@kantar](#)
- [@mdund](#)

Submitted: 10 June 2021
Published: 23 October 2021


License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Summary


Kinetics Toolkit is a Python package for generic biomechanical analysis of human motion that is easily accessible by new programmers. The only prerequisite for using this toolkit is having minimal to moderate skills in Python and Numpy.


While Kinetics Toolkit provides a dedicated class for containing and manipulating data (Time Series), it loosely follows a procedural programming paradigm where processes are grouped as interrelated functions in different submodules, which is consistent with how people are generally introduced to programming. Each function has a limited and well-defined scope, making Kinetics Toolkit generic and expandable. Particular care is given to documentation, with extensive tutorials and API references. Special attention is also given to interoperability with other software programs by using Pandas Dataframes (and therefore CSV files, Excel files, etc.), JSON files or C3D files as intermediate data containers.

Kinetics Toolkit is accessible at <https://kineticstoolkit.uqam.ca> and is distributed via conda and pip.

 kineticstoolkit.uqam.ca

 github.com/kineticstoolkit

 linkedin.com/in/felixchenier

 chenier.felix@uqam.ca

Chénier, 2022. Kinetics Toolkit: An Open-Source Python Package to Facilitate Research in Biomechanics. Journal of Open Source Software.