Python BTK 0.3 Cheat Sheet

(version 0.1)

Learn more on

http://b-tk.googlecode.com

http://biomechanical-toolkit.github.io/

Btk contributor : Fabien Leboeuf (University of Salford, UK)

You want to start programming with python Btk.
You don’t know how to get C3D-embeded data.
= > This cheat sheet gathers all useful methods for you.

Preliminary

Install BTK python

Prequisite :

- Select a python environment (e.g. Anaconda, pythonxy)
- Download corresponding OS BTK Python binary
- Install the BTK package in your Python package folder
- Call BTK from your script by typing :

```
import btk
```

Find Help

Find help on a method, or function

- Use online Doxygen documentation :
- in your script, type

```
help (btk.btkAcquisition.GetAnalog)
```

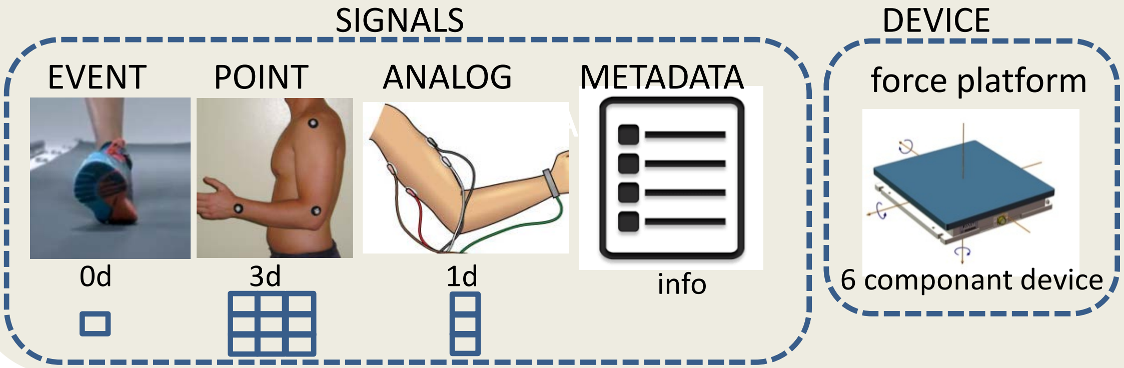
return help on the the method *GetAnalog()* of the *btkAquisition* object

Use support

- Post a message on the BTK Users forum

What can i get from a C3D file?

The C3D file format is a standard widely use to store motion capture system



File I/O

Read

Goal : construct a btkAcquisition Object from a C3D

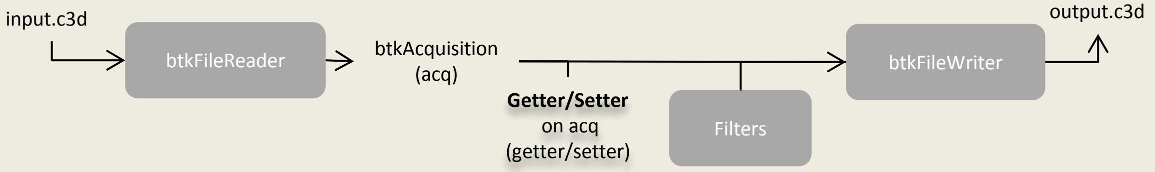
```
reader =  btk.btkAcquisitionFileReader()
reader.SetFilename(“input.c3d")
reader.Update()
acq = reader.GetOutput()
```

Write

Goal : from a btkAcquisition, generate a C3D file

```
writer =  btk.btkAcquisitionFileWiter()
writer.SetInput(acq) # acq = btkAcquisition
writer.SetFileName(“output.c3d")
writer.Update()
```

Pipeline



Object accesible from a btkAcquisition

btkEvent btkPoint btkAnalog btkMetadata

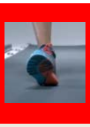
examples

*acq.**GetPoint**(“LASI”), acq.**GetEvent**(0) , acq.**GetAnalog**(“emg1”), acq.**GetMetaData**()*

Force platform accessible through a **Filter**
(see Force platform section)

btkAcquisition basic info

acq.GetFirstFrame()
acq.GetLastFrame()
acq.GetDuration()



btkEvent

Get an event

ev0= acq.GetEvent (0) ⇨ **Return btkEvent**

- Label
- Frame
- Description
- Context

Need an index

See "Collection" to handle ALL Points

A Context maybe General or a Left (right) side event

Convenient getter/setter

ev0.GetLabel()
ev0.setLabel("Toe Off")

ev0.GetFrame()
ev0.SetFrame(200)

ev0.GetContext()
ev0.SetContext(200)

ev0.GetDescrition()
ev0.SetDescription("begin of the swing phase")

btkPoint

Basic info

*acq.**GetPointFrequency**()*
*acq.**GetPointFrameNumber**()*
*acq.**GetPointNumber**()*
*acq.**IsEmptyPoint**()*

Get Value or Values

Get value at frame 10

*LASI.**GetValue**(10) # or*
acq.GetPoint ("LASI").GetValue(10)

Return a numpy array (size : 1,3)

Get all values

*LASI.**GetValues**() # or*
acq.GetPoint ("LASI").GetValues()

Return a numpy array (size : n frames, 3)

Use numpy array indexing to get data

Values = LASI.GetValues()
*Values**[0,2]** # get row 0, col 2*
*LASI.GetValues()**[:,2]** # all rows in col 2*

See also :
http://docs.scipy.org/doc/numpy/user/basics.indexing.html

Find a point

see : iterator

Get a Point (eg LASI or index 0)

*LASI= acq.**GetPoint** ("LASI")*
Pt0= acq.GetPoint (0)

Return btkPoint()

- Values
- Description
- Label
- Residual
- Type

See "Collection" to handle ALL Points

Set a new Value

For Point LASI , Set value (100) at a frame 10 on col 2

*LASI.**SetValue**(10,2, 100) # or*
acq.GetPoint ("LASI").SetValue(10,2, 100)

Set new Values

Change all values of the LASI marker
Howto : pass a numpy.array

***Import numpy as np** # do not forget to import numpy*

Nframe = acq.GetPointFrameNumber()
values = np.zeros((Nframe ,3)) # zeros array (size : Nframe rows, 3 col)
LASI = acq.GetPoint ("LASI")
*LASI.**SetValues**(values)*
# ---or---
acq.GetPoint ("LASI"). SetValues(values)

Convenient getter/setter

*acq.GetPoint(0).**GetLabel**()*
*acq.GetPoint("LASI").**Set**Label("left ASIS ")*

*acq.GetPoint(0).**GetDescription**()*
*acq.GetPoint("LASI").**Set**Description(« left antero-superior ilica spine ")*

Create a New Point and Append to a btkAcquisition

*newpoint = **btk.btkPoint("newLabel", acq.GetPointFrameNumber())***
newpoint.SetValues(values) # values a 3d numpy array
acq.AppendPoint(newPoint)

By default : type is marker

But…A btkPoint is not a marker only !

In Biomechanics, you can find Euler Angles, Moment

btk.btkPointType.Angle
btk.btkPointType.Moment

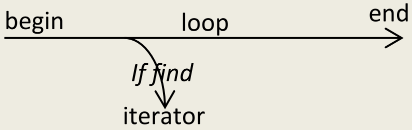
Look out some commercial model add power as a 3d vector ! Then, in BTK, you will find : btk.btkPointType.Power

examples : Read an angle. Append a new point as an angle into an acquisition

angle= acq.GetPoint ("LHipAngles") # read an angle
*acq.AppendPoint(newPoint,**btk.btkPointType.Angle**) # append newpoint as an angle*

Iterator

the parameter behind the convenient « Find » method



Context : you want find a parameter by its label

Principle :

1. Find method from your acquisition
2. This will return an iterator
3. Get iterator value (could be a point, analog,event)

Example:

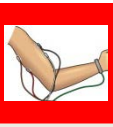
# find a point
*myIt= acq.**FindPoint** ("LHipAngles")*
# know the type of iterator
Print myIt
# get its value
*myIt.**value()** # return a btkPoint here*
# now tou can get the btkPoint by classic method
*myIt.value().**GetValues()***

Similar process with :
FindEvent(), FindAnalog()

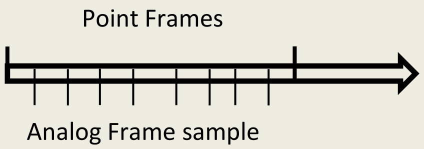
Unknown point

myIt= acq.FindPoint ("unknownPoint")
# doesn't call an Exception
A windows error is displayed if :
myIt.value() #cannot find a btkPoint()

btkAnalog



Keep in mind that point and analog frequencies might be different



Basic info

*acq.**GetAnalogFrequency()***
*acq.**GetAnalogFrameNumber()***
*acq.**GetAnalogNumber**()*
*acq.**GetNumberAnalogSamplePerFrame()***
*acq.**IsEmptyAnalogs()***

Get an analog(eg signal labeled EMG1 or index 0)

*emg1= acq.**GetAnalog** ("EMG1")*
signIndex0 = acq.GetAnalog(0)

Return btkAnalog

- Values
- Description
- Label
- Offset,Scale,Unit

Get Value or Values

Get value at analog frame 10

emg1.**GetValue**(10) #or
acq.GetAnalog ("EMG1").GetValue(10)

Return a numpy array (size : 1,1)

Get all values

emg1.GetValues() # or
acq.GetPoint ("EMG1").GetValues()

Return a numpy array (size : n frames, 1)

Find an analog

see : iterator

Convenient getter/setter

emg1Label= emg1.**GetLabel ()**
Emg1.**Set**Label ("rectus femoris")

emg1.**GetDescription ()**
emg1.**Set**Description("finewire")

Idem with :

GetOffset (), GetScale(), GetUnit()
SetOffset (-300), SetScale(2), GetUnit("mv")

Set Value or Values

For analog EMG1, Set value (100) at a frame 10

emg1.**SetValue**(10,100) #or
acq.GetAnalog ("EMG1"). SetValue(10,100)

Change all values of the LASI marker
Howto : pass a numpy.array

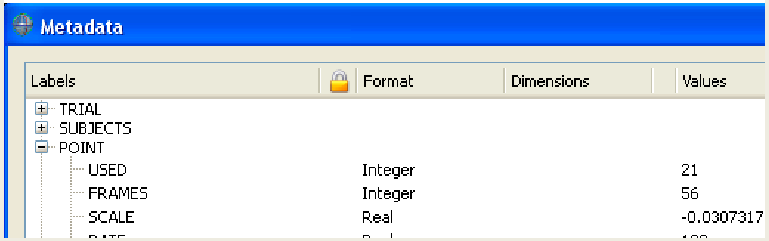
Import numpy as np

Nframe = acq.GetAnalogFrameNumber()
values = np.zeros((Nframe ,1)) # zeros array (size : Nframe rows, 1 col)
emg1 = acq.GetAnalog ("EMG1")
emg1.**SetValues**(values)
---or---
acq.GetAnalog ("EMG1"). SetValues(values)

Create a New Analog and Append it to a btkAcquisition

newAnalog = btk.btkAnalog("newLabel", acq.GetAnalogFrameNumber())
newAnalog.SetValues(values) # values a 1d numpy array
acq.AppendAnalog(newAnalog)

btkMetaData



Labels		Format	Dimensions	Values
⊞ TRIAL				
⊞ SUBJECTS				
⊟ POINT				
USED		Integer		21
FRAMES		Integer		56
SCALE		Real		-0.0307317

Get a metadata

md=acq.GetMetadata () # point out on ROOT

return btkMetadata

- Description
- Label
- info
- Child

Return a **btkMetaDataInfo**

Return another **btkMetaData**

Convenient getter/setter

All metadata have proper get/set method (i.e GetLabel, GetDescription(), GetInfo, GetChild()
... idem with set)

However what we want it's to get a definite metadata !!

Example : we want POINT::SCALE information

# return btkMetaDataInfo attached to SCALE not its value
*scaleInfo = md.FindChild("POINT").**value()**.FindChild("SCALE").**value()**.**GetInfo**()*

Notice : value() an iterator method because FindChild will return a metadata Iterator

# get SCALE value need btkMetaDataInfo method : ToDouble()
*scaleValue = scaleInfo.**ToDouble()***

Other btkMetaDataInfo method : *ToInt(),ToString()*

Create a metadata

⚠ **the c3d format accept only one-level metadata**

Example : add a metadata " Subject " and its sub-metadata "Name « to an acquisition

md_subject = btk.btkMetaData('Subject') # create main metadata
*btk.**btkMetaDataCreateChild**(md_subject, "Name", "House")# add a child*
*acq.GetMetaData().**AppendChild**(md_subject) # append new metadata*

After passing "acq" to a " btkFileWriter ", you will find the metadata "Subject" at the end of the list

btkCollection

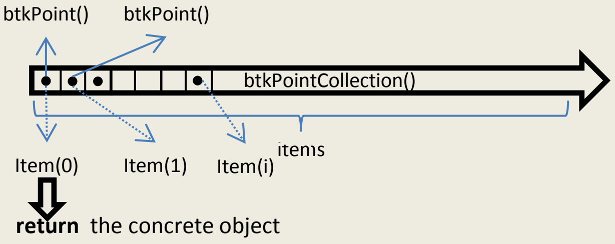
Definition : a Collection is a list of btk-object (e.g: btkEvent(), btkPoint(), btkAnalog())

Do not mix up Collection with the standard python list parameter btkCollection has its own method

Some methods returning a btkCollection

acq.GetPoints()
acq.GetAnalogs
acq.GetEvents()

Illustration



Get a object-item

Pts = acq.GetPoints()
Pts.GetItem(0).GetValues() #get values of the btkPoint located at item 0
Pts.GetItem(0).GetLabel) #get label of the btkPoint located at item 0.

Object iteration

The convenient function Iterate

*for it in **btk.Iterate**(acq.GetEvents()):*
print it.GetLabel() # display each label of it, i.e a btkEvent object

Notice : we write btk.Iterate because the btk package is loaded with the directive : *import btk*

Create an empty acquisition

An empty acquisition is enabled throught calling both btkAcquisition constructor and Init method

newAcq = btk.btkAcquisition()
newAcq.Init(5, 200, 10, 2)

Init signature
(number of Point , point frame number, number of analog , analog sample per point frame) .
In the example above, we infer that we have 5 btkPoints, 10 btkAnalogs, then, that point frame number and analog frame number are 200 and 400 respectively.

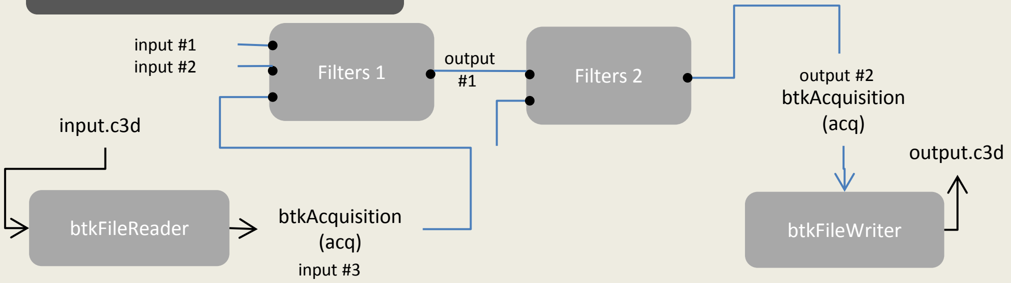
What for ?

- Store time-normalized data (e g : data normalized on a gait cycle)
- Others ?(feel free to propose your application through the Btk website

BTK Filters (Theory)

A set of **Filters** consists of a **Pipeline. A Filter articulates object for a common purpose**. Filters process input data only on demand when using the Update() method. **Updating only the last filter should be enough to process all the other ones**

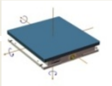
Illustration



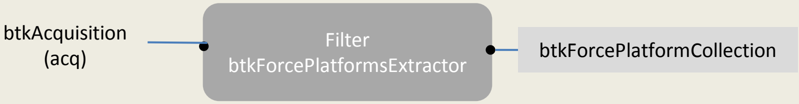
In this illustration, 2 filters are inserted. Let us notice than filter #1 has 3 inputs parameter whereas filter# 2 , 2 inputs only.
Output#1 is not necessary a btkAcquisition. However, output# 2 has to be a btkAcquisition because eventually, we write a c3d

Advice : identify the nature of each filter input from the main Documentation

BTK Filters (Practice)



Get the 6 components of a force platform



btkAcquisition
(acq)

Filter
btkForcePlatformsExtractor

btkForcePlatformCollection

Notice : the output object is a Collection. That’s mean we can iterate on it !
The number of item is the number of force platform

Example :

```
reader = btk.btkAcquisitionFileReader()
reader.SetFilename(“file.c3d")
reader.Update()
acq = reader.GetOutput()
# the filter---------
pfe = btk.btkForcePlatformsExtractor()
pfe.SetInput(acq)
pfe.Update()
pfc = pfe.GetOutput() # a btkPlateFormCollection
#--------------------
pf1 = pfc.GetItem(0) # item 0 = First force platform
```

Return btkForcePlatform

- Label
- Type
- Channel (0 to 5)
- Corner (0 to 3)

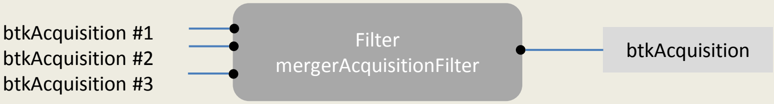
btkAnalog() per channel

Numpy array(1,3) = global position

```
pf1.GetType()
pf1.GetChannelNumber()
ch0 = pf1.GetChannel(0) # ch0: btkAnalog
ch0.GetLabel()
ch0.GetValues() return the n analog values
```

Generally, channels 0 to 2 = force components, 3 to 5 = moment components

Merger



btkAcquisition #1
btkAcquisition #2
btkAcquisition #3

Filter
mergerAcquisitionFilter

btkAcquisition

A merger is proprer method if a motion capture system dissociates biomechanical information on different file

Concrete example : gathering files provided from a « Motion Analysis Corp » system

```
# Readers
readerTRB = btk.btkAcquisitionFileReader()
readerTRB.SetFilename("myGait.trb")
readerANB = btk. btkAcquisitionFileReader()
readerANB.SetFilename("myGait.anb")
readerCAL = btk. btkAcquisitionFileReader()
readerCAL.SetFilename("forcepla.cal")
readerXLS = btk. btkAcquisitionFileReader()
readerXLS.SetFilename("myGait.xls")

# Merger
merger = btk.btkMergerAcquisitionFilter()
merger.SetInput(0, readerTRB.GetOutput())
merger.SetInput(1, readerANB.GetOutput())
merger.SetInput(2, readerCAL.GetOutput())
merger.SetInput(3, readerXLS.GetOutput())
merger.Update()

# Writer
writer =  btkAcquisitionFileWriter()
writer.SetInput(merger.GetOutput())
writer.Update()
```

Remote Power of python scientific package

matplotlib

SciPy.org Sponsored By ENTHOUGHT

Plot

A simple plot of the Hip Angles

```
import matplotlib.pyplot as plt

reader = btk.btkAcquisitionFileReader()
reader.SetFilename(“file.c3d")
reader.Update()
acq = reader.GetOutput()


values =acq.GetPoints("LHipAngles")

fig,(ax1,ax2,ax3) = plt.subplots(3,sharex = True)
plt.suptitle("LHipAngles« )
ax1.plot(values [:,0]
ax1.set_title("X-axis")
ax2.plot(values [:,1]
ax2.set_title(« Y-axis")
ax3.plot(values [:,2]
ax3.set_title(« Z-axis")
```

Filtering data

Example : emg Filtering
1st step : cut 50 hz
2nd : high pass filter

```
from scipy import signal

reader = btk.btkAcquisitionFileReader()
reader.SetFilename(“file.c3d")
reader.Update()
acq = reader.GetOutput()

fa=acq.GetAnalogFrequency()

emg1 = acq.GetAnalog(“EMG1”) #analog to filt

# digital filter configuration
bEmgStop, aEMGStop = signal.butter(2, np.array([49.9,
50.1]) / ((fa*0.5)), 'bandstop')
bEmgHighPass, aEmgHighPass = signal.butter(2,
np.array([20, 500]) / ((fa*0.5)), 'bandpass')

# apply digital filter-----
# stop 50hz
value50= signal.filtfilt(bEmgStop, aEMGStop, emg1
.GetValues(),axis=0 )
# highpass
valueHP= signal.filtfilt(bEmgHighPass, a aEmgHighPass,
value50),axis=0 )
#--------

emg1.SetValues(valueHP) # set new values
```

Linear Algebra

Example : compute optimal Rotation matrix of the marker cluster (LASI,RASI,RPSI,LPSI)

```
import scipy.linalg

reader = btk.btkAcquisitionFileReader()
reader.SetFilename(“file.c3d")
reader.Update()
acq = reader.GetOutput()

# Creation of an numpy array for the referencial frame and the chosen frame
# each row is the X,Y, Z coordinates of the one pelvis marker at the reference frame
data_FrameRef = np.array([acq.GetPoint(’LASI’).GetValues()[0,:],
                          acq.GetPoint(’RASI’).GetValues()[0,:],
                          acq.GetPoint(’LPSI’).GetValues()[0,:],
                          acq.GetPoint(’RPSI’).GetValues()[0,:]])
Frame = 40
data_FrameChosen = np.array([acq.GetPoint(’LASI’).GetValues()[Frame,:],
                             acq.GetPoint(’RASI’).GetValues()[Frame,:],
acq.GetPoint(’LPSI’).GetValues()[Frame,:],
acq.GetPoint(’RPSI’).GetValues()[Frame,:]])
# Difference with mean values
A = data_FrameRef-np.mean(data_FrameRef,axis=0)
B = data_FrameChosen-np.mean(data_FrameChosen,axis=0)
# transposition of the data
A = A.transpose()
B = B.transpose()
# matrix multiplication
# (Note : with a numpy array, it’s the dot method, not the multiplication operator)
C = np.dot(B,A.transpose())
# singular decomposition, called the svd method of the scipy/linalg module
P,T,Q = scipy.linalg.svd(C)
# computation of the nearest rotation matrix R
mat = np.array([[ 1., 0., 0.],
                [ 0., 1., 0.],
                [ 0., 0., scipy.linalg.det(np.dot(P,Q.transpose()))]])
R = np.dot(P,np.dot(mat,Q.transpose()))
```

A Script for starting

Ready to start, but you don’ t know how.

- **Copy/paste this script in your IDE**
- **Save it as myScript.py**
- **Run it**

```
# -*- coding: utf-8 -*-
"""

Created on Mon Jul 20 11:26:33 2015
@author: --
"""

#------ import packages ---------
import btk
import matplotlib.pyplot as plt
import scipy
import scipy.signal
import scipy.linalg

plt.close("all")

#------ READ YOUR FILE ---------
reader=btk.btkAcquisitionFileReader()
reader.SetFilename("input.c3d")
reader.Update()
acq=reader.GetOutput()

#------ EXPLORE YOUR ACQUISITION ---------

#------ CONSTRUCT YOUR PIPELINE ---------

#------ WRITE YOUR FILE ---------
writer=btk.btkAcquisitionFileWriter()
writer.SetInput(acq)
writer.SetFilename('output.c3d')
writer.Update()
```