

コミッショニングツールワーキンググループ

2018/6/18 13:15-14:00

参加者： 苔山, 横澤, 山本尚弘, 佐々井, 神田, 伊藤, 譲原, 田越, 大原, 坂井,

記録：田越

進捗状況

タスク

- PythonからFrameデータを呼び出す(C)：大原, 田越, 神田, => 未着手
 - 複数のCHを一度のファイルオープンで出来るほうが良い.
 - LIGOはFrameライブラリは使っていない, framecppを使っている.
- Frameデータ チャンネル一覧(KAGALI,Python)：FrChannelsを書き換え, 譲原
 - (譲原)FrChannelsを調べた.
 - (田越)kagaliappsにFrChannels相当のプログラムを用意して欲しい
 - その他詳細は付録に記載
- 時間指定をしてファイルを開くツール：田越
 - 田越：主データ装置にインストールした, time_to_framedump.py, time_to_plotasd.pyでの実装方法について解説 (/usr/local/binにある)
 - pythonのargparseはコマンドライン引数の取り扱いに有用.
 - 時刻変換はLIGOTOOLSにあるtconvertを使っている. 非常に有用. シェルスクリプトなのでどこでも動く.
 - Frameデータの読み込みはlalframeのfreadを使っている.
 - 今回の進展は無し.
- 長期の時系列(Python)：佐々井
 - histogram : pythonを勉強しながら進めている. 乱数を振ってヒストグラムを書く.
 - betelgeuse, taurusで動かない. モジュールが入っていない. pygtk?
 - rootでもモジュールがなくて出来ない.
- 時系列のヒストグラム(Python)：佐々井 => 未着手
- detrend(トレンド補正)(C)：大原

- 大原：Python, Matlabを調査した
 - これらを参考に早急に作成する
 - HHTのような新しい方法も作れるかも知れない。(研究テーマになる)
 - 大原：pythonと同じものをCで作った。gitへはまだ。
- window function(C)：大原
 - 大原：トレンド補正の後にやる。
 - まだ。
- high pass, low pass filter, band pass, band stop filter(C), filtfilt(C)：坂井, 大原
 - 坂井：大原さん, 酒井君がつくったものを発展させていく。
 - 坂井：大原さんのコードを読んでいる。これらを改良してKAGALIに入れる。
- スペクトログラム(C,...)：佐々井 => 未着手
- コヒーレンス(2つの信号のクロススペクトラム)(C,Python)：議論無し
- 1つのCHと多数のCHとのコヒーレンス(C,Python)：議論無し
- LIGO でやってるiDQのようなこともいずれする必要があるか。
 - iDQ:多変量解析
 - 伊藤：韓国Gがやっているらしい(John Ohと話をする)

GUI ツール

- 田越：PythonによるGUIツールを調べている。GWpyも調べる。
- 山本：三代君がGWpyで遊んでいる。GWpyを使ったツールがあれば、神岡で使える。
- 山本：GWpyは表示が速い。読み出し速度の問題解決に有効そうである。

その他

- 横澤：チャンネル名とその内容の説明のリストを作っている。
 - 8割方完成。phase-1の高サンプリングレートCHは完了。
 - PEM CHのリストアップをしている。イメージが固まった。
- 山本：譲原 daily summary
 - 付録参照

次回

2018年6月26日 12:15 - 13:00 (コミッショニング会議終了後)

付録

Frameデータ チャンネル一覧(KAGALI,Python) : FrChannelsを書き換え(譲原)

- ソースコードを読んだ (libframe-8.30/src/FrChannels.c)
- 基本的にframe fileのなかにあるchannelでループを回して、channel名とサンプリングレートを出力している
- FrChannelsコマンドと同じようなことをする関数を作りたい
- KAGALI関数にしたときの引数の構想

```
kglstatus status,  
int nChannels,  
char cChannels[200][200],  
int *fs
```

- みたいな感じか？
- この引数だとチャンネル名が入る配列の個数を200個までと最初から決め打ちしているが、今のframe fileには20000くらいチャンネルがある 1度ファイルを開いて、何チャンネルあるか調べないと配列の動的確保ができないのでどうしたらいいか思案中
- そもそもこの関数の目的は、1ヶ月のスパンでchannel名の変動が知りたいということだと認識しているが正しいか？
- => (横澤) 正しい
- (田越) kagaliappsにFrChannels相当のプログラムを用意して欲しい

サマリーページ(譲原)

今回やったこと

- KAGALIに新しい関数を追加した。1つのframe fileを開いて複数のchannel名を与えると多次元配列にその結果を代入して返す。多次元配列の要素数は nChannel x (fs_max * duration)、最大サンプリングレートで制限される。
- 実際にこの関数を動かすときは異なるfsで実行すると確保する配列に無駄ができてしまう。ちなみに90channelの内訳は16384Hzは4channel、2048Hzは82channel、16Hzが4channelなので16384Hzに合わせるとほとんどの配列の中身が無駄。
- このKAGALI関数を使ってbetelgeuseで動作テストをした。やったことは1つのKAGALI関数でframe fileから複数のchannelを読み込むだけ(データの書き出しやプロットはまだなし、将来的にROOTのプロットを使うことにするのでデータ書き出しはしない)

実行時間まとめ

1. 90channelを1時間分読み込んだ(condorなし、betelgeuseでコンパイル) => 7分
2. 90channelを1時間分読み込んだ(condorなし、polluxでコンパイル) => 4分 => 概算すると24時間データを読み込むのに312分かかる(まあ許容範囲か?) => あとbetelgeuse, polluxどちらでコンパイル

ルするかで計算時間が割と変わる、以下はすべてpolluxでコンパイルした

3. condorでfs=16Hzの4channelを1時間分読み込んだ => 7分14秒(メモリ使用量は34GB)
4. condorでfs=16Hzの4channelを24時間分読み込んだ => 1時間48分(メモリ使用量は94GB)
5. condorでfs=2048Hzの82channelを24時間分読み込んだ => 2時間32分(メモリ使用量は94GB)
6. condorでfs=16384Hzの4channelを24時間分読み込んだ => 今走らせてます

一番メモリ容量を食う計算が(5)のパターンだと思うので、メモリ量的には動きそう

使用したコードはbetelgeuseの ~yuzurihara/work/0612_comm/ にある。 使用した関数は
kagali/kglcommon/frame/src/KGLFrameMultiIO.c にあるKGLGetMultiParameterOfFile()と
KGLReadFrameMultiData()

次の課題

- 上記のKAGALI関数の引数でchannel名が入った配列を渡す方法が怪しい => まさかの文字列配列のポインターと文字列配列の配列の違いを理解できていないことが判明。 現在勉強中なので将来的に文字列配列のポインターになると思います。 今は固定長の文字列配列で代用してます。
- 山本くんのROOTプロットコードを触る。
- サマリーツールに取り込む。