Tools for Acquiring IRIS Data

Charles J. Ammon, Penn State

EarthScope Introductory
Data Processing Short Course

August 2016

IRIS Data Request Tools

- Web Services
- Wilber (3)
- FetchMetadata
- FetchData
- FetchSyn
- MatlabFetch
- JWEED
- BREQFAST
- SOD (Standing Order For Data)
- obsPy

IRIS Data Request Tools

- Web Services
- Wilber (3)
- FetchMetadata
- FetchData
- FetchSyn
- MatlabFetch
- JWEED
- BREQFAST
- SOD (Standing Order For Data)
- obsPy

SOD

The Standing-Order-For-Data (SOD) tool is a java-based application that allows you to request and download seismograms from IRIS and other compatible data centers.

The basic idea is to develop a recipe, a set of XML parameters, that describe the data that you want and the processing that you want to applied to the data. The XML scripts can take a little time to get comfortable with, but if you make similar requests, once set up, this method is quite powerful.

The post processing includes instrument deconvolution that is compatible with the SAC functions, marking SAC headers, rotation, etc. The data are ready to go when the script is done.

Sample SOD Recipe

```
<?xml version="1.0"?>
<sod>
    <eventArm>
        <fdsnEvent>
            <originTimeRange>
                <startTime>
                    <year>2003</year>
                    <month>1</month>
                    <day>1</day>
                </startTime>
                <endTime>
                    <year>2003</year>
                    <month>1</month>
                    <day>10</day>
                </endTime>
            </originTimeRange>
            <magnitudeRange>
                <min>6</min>
            </magnitudeRange>
        </fdsnEvent>
        <originOR>
            <originAND>
                <magnitudeRange>
```

Sample SOD Recipe

```
            <unit>DEGREE</unit>
            <min>30</min>
            <max>90</max>
        </distanceRange>
        <phaseRequest>
            <model>prem</model>
            <beginPhase>ttp</beginPhase>
            <beginOffset>
                <unit>SECOND</unit>
                <value>-120</value>
            </beginOffset>
            <endPhase>tts</endPhase>
            <endOffset>
                <unit>SECOND</unit>
                <value>360</value>
            </endOffset>
        </phaseRequest>
        <fdsnDataSelect/>
        <someCoverage/>
        <printlineSeismogramProcess/>
        <sacWriter/>
    </waveformArm>
</sod>
```

http://www.seis.sc.edu/SOD/documentation/tutorials/demo.html

SOD

SOD provides command line tools to look for events, stations, channels, and seismograms.

You execute SOD from the command line using

```
sod -f myReceipe.xml
```

SOD will run until the computer shuts down (and can be reset to restart when power returns). You can even ask for future events - all large events.

SOD is a not a tool to look for older data, some older compression codecs are not supported, so be careful in the 1980’s and early 1990’s.

FetchSyn

In seismology a "synthetic" seismogram means a seismogram computed from a model - a prediction of what some data should look like.

There are many ways to compute seismograms and learning at least some of those is part of becoming a practicing earthquake seismologist.

FetchSyn is a command-line tool to download synthetic seismograms from IRIS (for a few, select, one-dimensional Earth models). You can download the script from:

`https://seiscode.iris.washington.edu/projects/ws-fetch-scripts`

You specify the source location and faulting geometry and then can request predictions at any point. What's nice is that you can use shortcuts (specify a station for a location, a GCMT event ID for an earthquake).

FetchSyn

The underlying technology of FetchSyn is the Syngine web service - a synthetic seismogram engine. Some of the capabilities are still being developed, but the tool is running and useful.

http://service.iris.edu/irisws/syngine/1/

Click on the link above and explore the service.

The FetchSyn python script calls this service, which downloads synthetic seismograms to your disk. The technology at IRIS is based on instaseis

http://instaseis.net

FetchSyn

Let’s examine an earthquake that occurred yesterday in the south-central Indian Ocean.

USGS Event Page

http://earthquake.usgs.gov/earthquakes/eventpage/us1000694i#executive

GCMT Event ID: 201608010742A

http://www.globalcmt.org/cgi-bin/globalcmt-cgi-bin/CMT4/form?itype=ymd&yr=2016&mo=8&day=1&oyr=1976&omo=1&oday=1&jyr=1976&jday=1&ojyr=1976&ojday=1&otype=nd&nday=1&lmw=0&umw=10&lms=0&ums=10&lmb=0&umb=10&llat=-90&ulat=90&llon=-180&ulon=180&lhd=0&uhd=1000&lts=-9999&uts=9999&lpe1=0&upe1=90&lpe2=0&upe2=90&list=0

FetchData

Start by getting some data (this is all on one line)

```
FetchData -N IU,II -L 00 -C LHZ -s 2016-08-01,07:42:50 -e
 2016-08-01,08:42:50 --radius 24:82.5:90 -m meta.txt -o
 data.seed
```

Convert to SAC with Meta Data

```
mseed2sac -m meta.txt -E
2016-08-01,07:42:50/-23.962/82.479/10/SIndianOcean
data.seed
```

FetchSyn

Now let's get the prediction for station BRVK

```
FetchSyn -N II -S BRVK -C Z -s 2016-08-01T07:42:50 -e
2016-08-01T08:42:50 -eventid GCMT:201608010742A -outfile
synth.seed
```

Convert to SAC with Meta Data

```
mseed2sac -m meta.txt -E
2016-08-01,07:42:50/-23.962/82.479/10/SIndianOcean
data.seed
```

ObsPy Demo

End