Pulsar observing with DiFX

A lot of good information is available in VLBA sensitivity upgrade memo 32 on pulsar observing in DiFX. Also, the DiFX users's guide (located in doc/userguide/trunk/ of your svn checkout) has lots of information: compile the pdf file with "make", then open in and skip down to section 5, "DiFX and pulsars".

tempo1 polyco format description

tempo2 documentation

A typical command line to produce a tempo1 polyco using tempo2 and an old-style .par file

```
tempo2 -f b1727-47.par -polyco "58315 58316 60 12 12 coe 1384" -tempo1
```

See the tempo2 documentation above for an explanation of the parameters.

Most of the problems you might run into when running in pulsar mode relate to just getting the "polyco" prediction file right, and understanding what phase 0 in this file refers to. There are a few points of wisdom here:

- Polycos can be produced by tempo or tempo2. However, tempo2 has some known failure modes when writing polycos; certain of the tempo2-only parameters in the ephemeris seem not to get propagated to the polyco. Accordingly, it is recommended to use tempo to produce polycos if at all possible. Note, however, that you can't just take a tempo2 eph file and use it with tempo! Ask for some help from a local pulsar expert.
- Polyco files produced using tempo2 (to generate a tempo(1) format polyco) often have 'coe' instead of the required '0' for the observatory number and this causes older DiFX versions to fail. The polyco file must be edited e.g. with `sed -i "s/coe/ 0 /g" polyco.dat`.
- Know which type of ephemeris file you have. Certain catalogs like PSRCAT provide either tempo2 or tempo ephemerides, depending on the pulsar. Tempo2 ephemeris files usually have prefix .eph and have TCB units. Old tempo files are .par and have TDB units.
- Old tempo .par can be converted into tempo2 .eph (`tempo2 -gr transform tempo1.par new_tempo2.eph` assuming you installed tempo2 plugins), although, for DiFX you might really not want to do that.
- New tempo2 .eph can be converted into tempo .par. A `tempo -gr transform tempo2.eph new_tempo1.par back` might do it, but, ask your pulsar person for help.
- An ephemeris (.par file) is useless for doing gating unless you also have the template profile used to generate that ephemeris, because otherwise you don't know where in the pulse profile phase 0 corresponds to. If you have an sufficiently good ephemeris that remains phase stable for the duration of the observation but you don't know where phase 0 is, you can use difx profile mode to use the baseband data to figure it out. See the users guide for details on how to run profile mode.
- Once you have an ascii profile for the pulsar (either the one that came with the ephemeris, or the one you generated from the output of difx2profile), you can use the tool profile2binconfig.py to generate a .binconfig file appropriate for pulsar gating. The exact parameters you give to profile2binconfig.py depends on the ascii profile file (the number of header lines and the column in the ascii file that contains the pulse profile), an example might be:
    - `profile2binconfig.py —profile=1024-0719_mspsrpi_std.ascii —polyco=bd179e1.J1024-0719.polyco -n 24 -s`

```
–binconfigfile=bd179e1.J1024-0719.binconfig –nonormalise
–lineskip=2 –profilecolumn=3 –dontzeronoise
```
  - Here, -n 24 is the number of bins in the output matched filter gating binconfig, –lineskip=2 says there are two header lines that should be skipped over, and –profilecolumn=3 says that the pulse profile is in the 3rd column.

There are a few "gotchas" for polyco files, too:

- The date/time is represented in two places in the polyco: an mjd+fractional mjd, and a date + UTC time (hhmmss.fff). Up until DiFX-2.5, the date was taken from the MJD, and the time from the UTC hhmmss. This is in fact incorrect, as David Nice says that the MJD + fractional MJD is the definitive value, and the UTC time is only provided "for convenience" and can be rounded off!! In practise, this rarely happened, but if the UTC time differed from the fractional MJD, then the reference time for the pulsar phase was being set incorrectly, which would ruin the gating. This has been fixed in DiFX-2.5, but it is always worth checking that the two times agree.
- In a related problem, sometimes tempo would give an MJD like 57342.99999999990, so an incorrectly rounded off value. DiFX would take the day from the MJD and the time from hhmmss, and be off by a whole day. In this case, mpifxcorr would fail to run.