

# QC Practical



Before lunch:

- Download new init scripts/ip addresses from Google Drive and log into AWS
- Push your feature branch to GitHub (if you haven't already)
- Do a Pull Request
  - \*Ask for reviews from your group  $\pm 1$
  - \*Note: there is no group 13
  - \*Jedi-ac<xx> for groups 1-17
  - \*jedi-acad<xx> for groups 18-20
- Move your ZenHub issue to the Review/QA column
- Review code as requested

# QC Practical



(Optional) Merge Anna's solution for yesterday's exercise

```
cd ~/jedi/fv3-bundle/ufo  
git pull upstream develop
```

# QC Practical



1. Review yesterday's code from another group (before lunch)
2. Like the radiosonde UFO, the background check filter can only treat one observed variable
  - Background check in `ufo/src/ufo/ufo_bgcheck_mod.F90`
  - Implement a background check that can be controlled by the configuration file (the `config_get_string_vector` function in `fv3-bundle/oops/src/oops/util/config_mod.F90` might be useful)
3. There are a number of other issues in the background check example
  - What are the main issues?

# Background check



```
subroutine ufo_bgcheck_post(self, hofx)
type(ufo_bgcheck), intent(in) :: self
real(c_double), intent(in) :: hofx(:)
! Some declarations skipped

missing = obSPACE_missing_value()
iobs = obSPACE_get_nlocs(self%obsdb)
allocate(yobs(iobs))
allocate(yerr(iobs))
allocate(flags(iobs))
flags(:) = 0

call obSPACE_get_db(self%obsdb, "ObsValue", trim(self%variable), yobs)
call obSPACE_get_db(self%obsdb, "ObsError", trim(self%variable), yerr)

do jobs = 1, iobs
  if (hofx(jobs)/=missing .and. yobs(jobs)/=missing .and. yerr(jobs)/=missing) then
    if (abs(hofx(jobs)-yobs(jobs)) > yerr(jobs)*self%threshold) then
      flags(jobs) = 2
    endif
  else
    flags(jobs) = 1
  endif
enddo

call obSPACE_put_db(self%obsdb, "QC", trim(self%variable), flags)

deallocate(yobs, yerr, flags)

end subroutine ufo_bgcheck_post
```

# Rerun 3D-Var



1. With the new observations it's important to check that things are converging correctly and to look at the increment.

```
cd <build>/fv3-jedi/test
ctest -V -R test_fv3jedi_3dvar_gfs > 3dvar.log
grep 'Jo/n' 3dvar.log
```

2. Now you can plot the increment:

```
cd <build>/fv3-jedi/test/Data/RESTART/
In PlotCubeSphereField.py:
    readvar = 'T' => readvar = 'ua'
python PlotCubeSphereField.py
display CubedSpherePlot_Field-ua_Level-50.png
```