

A Python Package for Climate Field Reconstruction

Feng Zhu¹, Julien Emile-Geay², Gregory J. Hakim³, Dominique Guillot⁴, Deborah Khider⁵, Robert Tardif³, W. Andre Perkins⁶

1. NSF NCAR 2. University of Southern California 3. University of Washington 4. University of Delaware 5. USC Information Sciences Institute 6. Allen Institute for AI

 pip install cfr

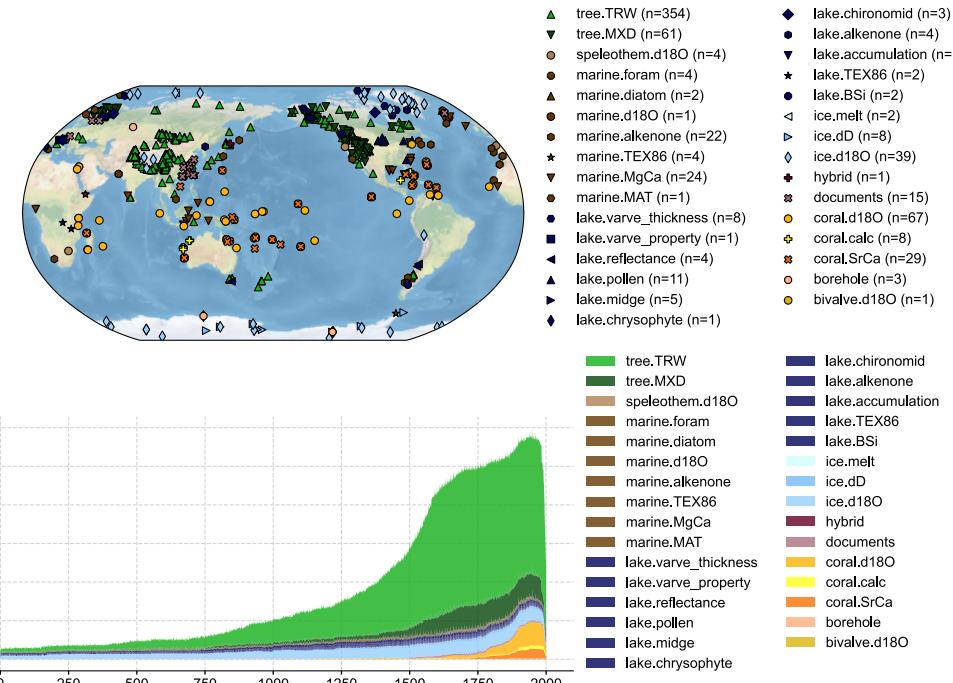


Proxy and Climate Data Cloud-Fetching

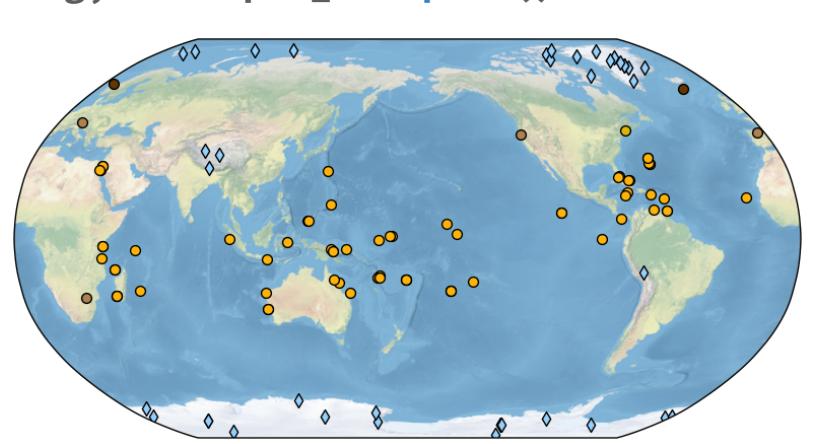
```
pdb = cfr.ProxyDatabase().fetch('PAGES2kv2')
obs = cfr.ClimateField().fetch('HadCRUTv5',
    vn='tas_mean', lat_name='latitude',
    lon_name='longitude').rename('tas')
```

Proxy Database Filtering and Visualization

```
fig, ax = pdb.plot(plot_count=True)
```

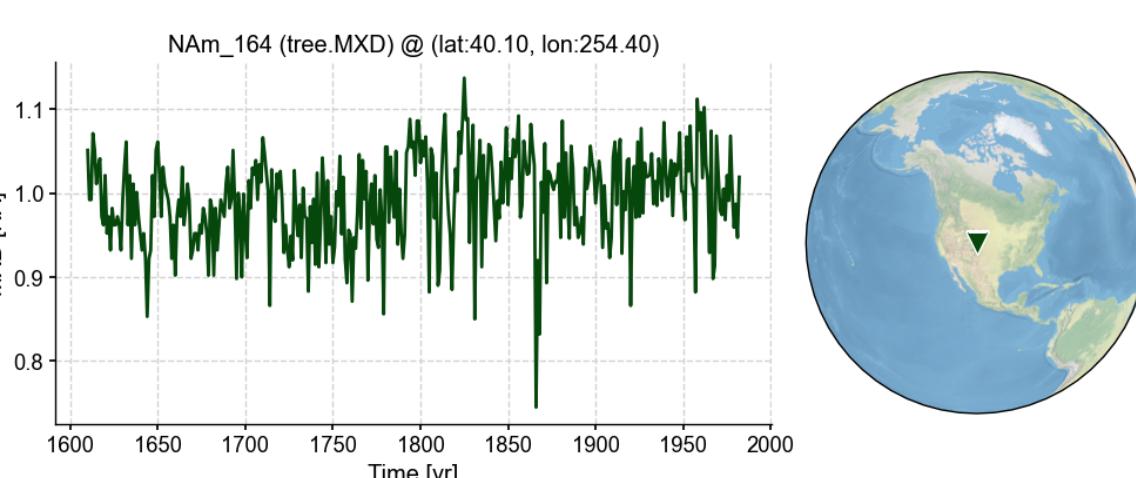


```
pdb_sub = pdb.filter(by='ptype', keys=['d18O'])
fig, ax = pdb_sub.plot()
```



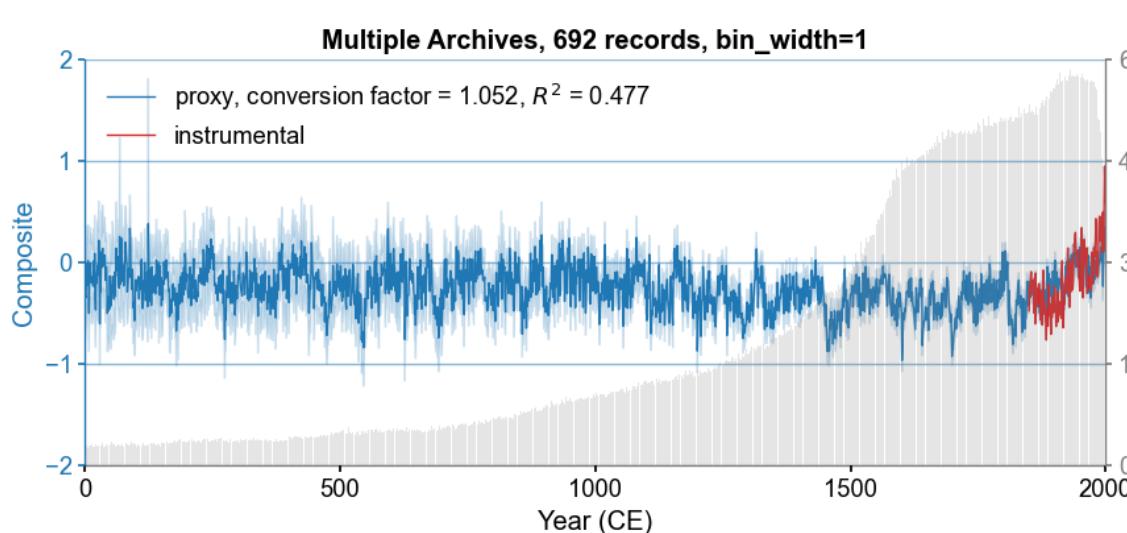
Proxy Records Accessing and Visualization

```
pobj = pdb['NAm_164']
fig, ax = pobj.plot()
```



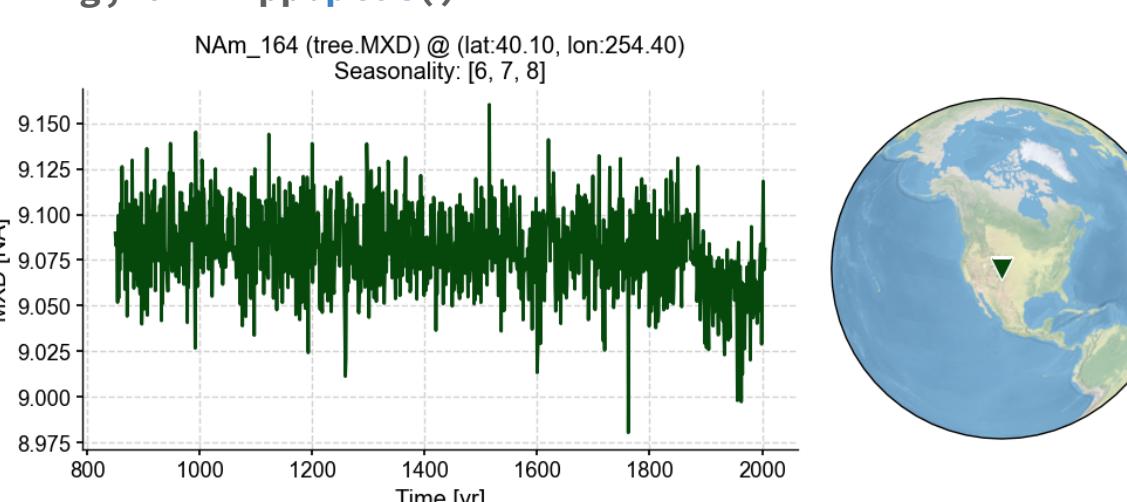
Proxy Composites (PAGES2k Consortium, 2017)

```
pdb = pdb.make_composite(obs=obs, bin_width=1)
fig, ax = pdb.plot_composite()
```



Proxy System Modeling (PSM)

```
model = cfr.ClimateField().fetch(
    'iCESM_past1000historical/tas')
pobj.get_clim(model, tag='model')
pobj.get_clim(obs, tag='obs')
psm = cfr.psm.Linear(pobj)
psm.calibrate(season_list=[[12,1,2], [6,7,8]])
pp = psm.forward()
fig, ax = pp.plot()
```



- Linear Regression (Hakim et al. 2016)
- Bilinear Regression (Tardif et al., 2019)
- Tree-ring width: VS-Lite (Tolwinski-Ward et al., 2011, 2013)
- Ice core d18O: PRYSM (Dee et al., 2015)
- Coral d18O: PRYSM (Dee et al., 2015)
- Coral Sr/Ca: PRYSM (Dee et al., 2015)
- Lake Varve Thickness: pseudoPAGES2k (Zhu et al., 2023)
- TEX₈₆: BAYSPAR (Tierney et al., 2014)
- UK₃₇: BAYSPLINE (Tierney et al., 2018)
- Mg/Ca: BAYMAG (Tierney et al., 2019)
- d18O: BAYFOX (Malevich et al., 2019)

Paleoclimate Data Assimilation (PDA)

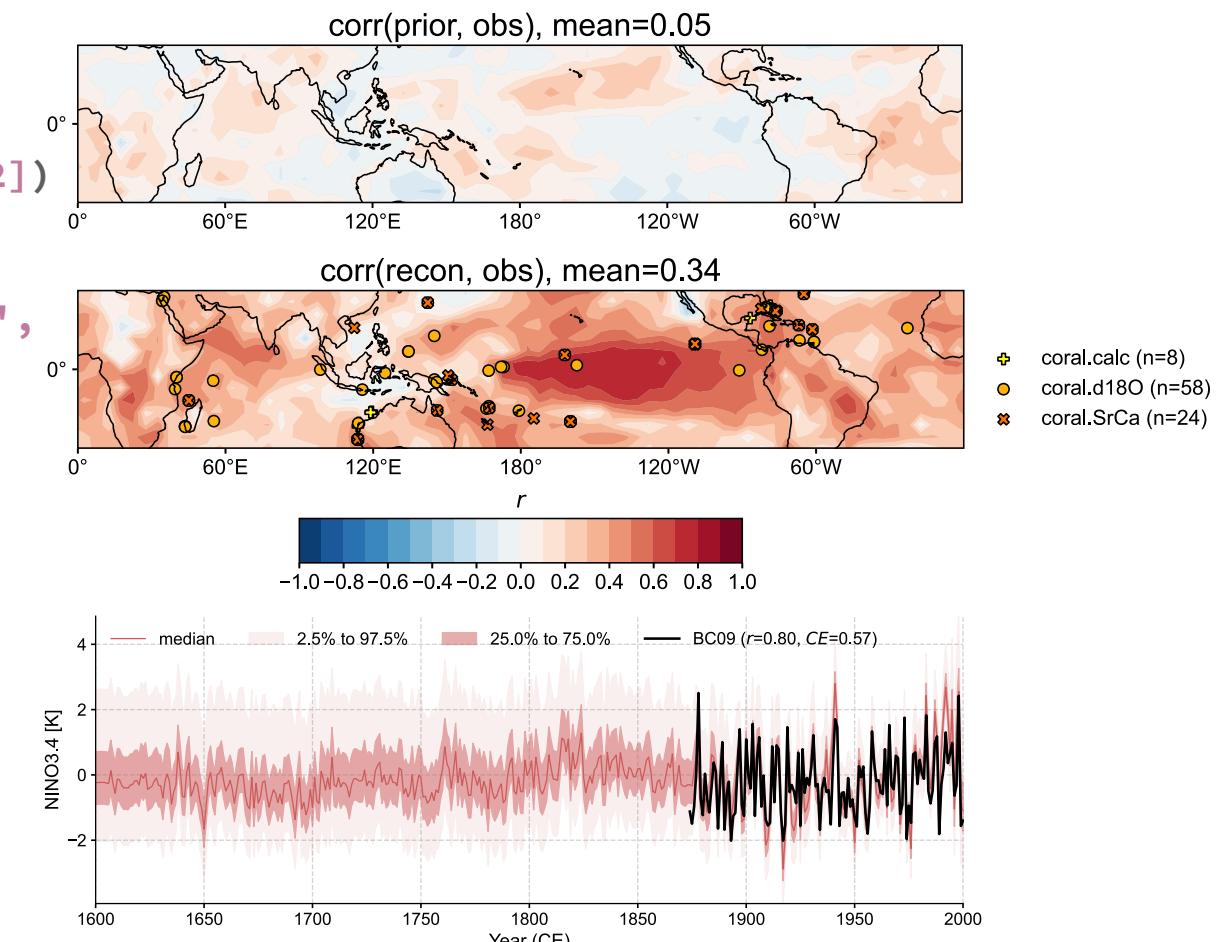
```
job = cfr.ReconJob()
job.load_proxydb('PAGES2kv2')
job.filter_proxydb(by='ptype', keys=['coral'])
job.annualize_proxydb(ptypes='ptype', months=[12,1,2])
```

```
job.load_clim(tag='prior',
    path_dict={'tas': 'iCESM_past1000historical/tas',
    anom_period=(1951, 1980)})
job.load_clim(tag='obs',
    path_dict={'tas': 'gistemp1200_GHCNv4_ERSSTv5',
    anom_period=(1951, 1980),
    rename_dict={'tas': 'tempanomaly'})
```

```
job.calib_psms(ptype_season_dict={
    'coral.d18O': [12, 1, 2],
    'coral.calc': [12, 1, 2],
    'coral.SrCa': [12, 1, 2]},
    anom_period=(1951, 1980))
job.forward_psms()

job.annualize_clim(tag='prior', months=[12, 1, 2])
job.regrid_clim(tag='prior', nlat=42, nlon=63)
job.crop_clim(tag='prior', lat_min=-35, lat_max=35)

job.run_da_mc(save_dirpath='./recons/lmr', recon_seeds=list(range(1, 11)))
```



Graphical Expectation-Maximization (GraphEM)

```
job = cfr.ReconJob()
job.load_proxydb('PAGES2kv2')
job.filter_proxydb(by='ptype', keys=['coral'])
job.annualize_proxydb(ptypes='ptype', months=[12,1,2])
```

```
job.load_clim(tag='obs',
    path_dict={'tas': 'gistemp1200_GHCNv4_ERSSTv5',
    anom_period=(1951, 1980),
    rename_dict={'tas': 'tempanomaly'})
```

```
job.annualize_clim(tag='obs', months=[12, 1, 2])
job.regrid_clim(tag='obs', nlat=42, nlon=63)
job.crop_clim(tag='obs',
    lat_min=-20, lat_max=20, lon_min=150, lon_max=260)
```

```
job.run_graphem(
    save_dirpath='./recons/graphem',
    grap_method='hybrid',
    cutoff_radius=5000, sp_FF=2, sp_FP=2)
```

