```
Filename: /Users/wujung/code/ooi_sonar/concat_raw_output.py

Line #    Mem usage    Increment   Line Contents
================================================
    24     14.8 MiB      0.0 MiB   def concat_raw(data_path,fname_all,date_wanted):
    25
    26                                 # Index the files in the wanted date range
    27     15.2 MiB      0.4 MiB       raw_file_times = [FILE_NAME_MATCHER.match(os.path.join(data_path,x)) for x in fname_all];
    28                                 # idx = [x for x in range(len(X)) if X[x].group('Date')=='20150912']  # solution 1
    29     15.2 MiB      0.0 MiB       date_list = map(lambda x: x.group('Date'),raw_file_times)  # if in python 3 need to do list(map()) to conver
    30     15.2 MiB      0.0 MiB       time_list = map(lambda x: x.group('Time'),raw_file_times)
    31
    32     15.2 MiB      0.0 MiB       if len(date_wanted)==1:
    33                                     idx_date = [i for i,dd in enumerate(date_list) if dd==date_wanted]  # solution 2
    34     15.2 MiB      0.0 MiB       elif len(date_wanted)==2:
    35     15.2 MiB      0.0 MiB           idx_date = [i for i,dd in enumerate(date_list) if dd>=date_wanted[0] and dd<=date_wanted[-1]]  # solutic
    36                                 else:
    37                                     print 'Invalid range: date_wanted!'
    38                                     idx_date = []
    39
    40     15.2 MiB      0.0 MiB       if len(idx_date)!=0:
    41     15.2 MiB      0.0 MiB           if time_list[idx_date[0]]>'000000':  # if midnight was recorded in the previous file
    42                                         idx_date.insert(0,idx_date[0]-1)
    43
    44                                     # Read first file for initialization
    45                                     particle_data, data_times, power_data_dict, freq, bin_size, config_header, config_transducer = \
    46    338.2 MiB    323.0 MiB               parse_echogram_file(os.path.join(data_path,fname_all[idx_date[0]]))
    47   1477.7 MiB   1139.5 MiB           data_mtx = get_data_mtx(power_data_dict,freq)
    48
    49                                     # print 'original size is ' + str({k: v.shape for (k,v) in power_data_dict.items()})
    50                                     # freq = frequencies.value()  # get freuqency values
    51                                     # freq = {k: str(int(v/1E3))+'k' for (k,v) in frequencies.items()}  # get frequencies
    52
    53   1477.7 MiB      0.0 MiB           for fnum in idx_date[1:]:
    54   1478.4 MiB      0.7 MiB               print '====== Processing file: ' + fname_all[fnum]
    55                                         # Read next file
    56    290.3 MiB  -1188.1 MiB               _, data_times_tmp, power_data_dict_tmp, freq_tmp, _, _, _ = parse_echogram_file(os.path.join(data_pa
    57    318.1 MiB     27.8 MiB               data_mtx_tmp = get_data_mtx(power_data_dict_tmp,freq_tmp)
    58    164.9 MiB   -153.2 MiB               data_mtx = np.concatenate((data_mtx,data_mtx_tmp),axis=1)
    59
    60                                 else:
    61                                     print 'date wanted does not exist!'
```