Readme:

Quickstart for users familiar with LIARv1:

The fastest way to get your old function call to work is to delete the 4th and all subsequent inputs (unless you needed to use the MolalityTF flag before).

The function call remains the same for the first three inputs. The fourth input and all additional inputs (Equation numbers, flags, etc.) are now optional. If you include them, then you will need to input them as input value pairs:

LIAR(Coordinates, Measurements, MeasIDVec, ‘Equations’, 1:16, ‘MolalityTF’, False,…)

If you skip “Equations” (as we now recommend) then the LIR routines will use your input measurement uncertainties (or default values if not provided) to determine which equation will provide the lowest uncertainty estimate for each estimate. Only that estimate will be returned.

Locally Interpolated Regressions (LIRs) for MATLAB: Estimates alkalinity (LIAR), nitrate (LINR), and pH (Total Scale) (LIPHR) and uncertainties from 16 possible combinations of other parameter measurements (each). The Locally Interpolated Regression method is described in detail in the paper:

*LIARv1: Carter et al., 2016, doi: 10.1002/lom3.10087*

*LIARv2, LIPHR, LINR citation:* [*https://doi.org/10.1002/lom3.10232*](https://doi.org/10.1002/lom3.10232)

For questions not addressed by the papers or this readme file, please contact Brendan Carter, [Brendan.carter@gmail.com](mailto:Brendan.carter@gmail.com), and he will get back to you as he is able. Suggestions, bug reports, and queries regarding code updates are welcome. [LIR push requests on Git](https://github.com/effataigus/LIRs) are welcome, but Brendan is still learning Git so a parallel E-mail might be prudent.

Running the LIR estimation routines in MATLAB is a 3 step procedure:

1. Download the required files: LIAR.m and LIARv2\_files.mat, LINR.m and LINR\_files.mat, and LIPHR.m and LIPHR\_files.mat (note, the [seawater package from CSIRO](http://www.cmar.csiro.au/datacentre/ext_docs/seawater.htm), is also required for some code uses).
2. Put all needed files into the current working MATLAB directory or a directory on the MATLAB path, \*
3. Call the LIAR function according to the detailed instructions below.

\* *Not sure what the MATLAB path is?* [*Check this out*](https://www.mathworks.com/help/matlab/matlab_env/what-is-the-matlab-search-path.html)*. An easy way get the LIR files on the path (without putting them in your working directory) is to navigate to the folder in the command window, then enter the command “addpath pwd”. You may need to do this again after each MATLAB restart.*

Updates: We intend to update regression coefficients as new quality controlled datasets become available. Please indicate versions if you use routines for any publications (for reproducibility). Contact the corresponding author for updates or check the [Git page](https://github.com/effataigus/LIRs).

Subfiles in the LIAR\_files.mat file include:

Coords: Latitudes, longitudes, depths, and approximate potential densities of seawater at each of the LIAR grid coordinate locations.

L\*Poly/L\*PolyExtra: Polygons defining subregions of the Atlantic and Arctic Oceans (NA: North Atlantic, SA: South Atlantic, NO: Arctic). “Extra” polygons for the North and South Atlantic cover the subsets of these ocean basins from 0 to 180° E (as opposed to 0 to 180° W).

Cs: *α* coefficients at the coordinates in Coords.

AAindsCs: a boolean array indicating whether various coordinates in Coords are in either the Arctic or Atlantic Oceans.

EMLRrec: 16 histograms ofagainst salinity for the 16 different possible regressions (see paper for a full explanation).

OAEstMat (LIPHR only): 16 sets of regressions of pH residual against the difference between the date of our test data set measurements and the mean dates of the training data measurements. Each set has regressions for 10 global potential density anomaly intervals. These are used to estimate the impact of ocean acidification on the pH estimated (see paper for a full explanation). This adjustment can be turned off when the function is called (see below).

Dates: Mean dates for the training data used to train each of the regressions.

XRec: The iteration number that produced the regression +1. This can be thought of as a measure of how data dense the region used to train the regressions was (with 2 being the densest and larger numbers being less dense). Generally, we expect higher numbers to produce worse estimates, but this hasn’t been tested.

NumFitters: A record of the number of measurements used to train each regression.

**How to call the function:**

The best answer to this question is found in the code comments. Open the LI\*R.m file and read the green text at the top of the function. The following is a basic overview only.

There are 3 required inputs for LIAR and LINR, and 5 for LIPHR. The order of the first three inputs is fixed for all LIRs and should be:

1. Coordinates: A 3-column array of Longitude (first column, degrees East), Latitude (2nd column, degrees N), and Depth (3rd column, meters) where you’ll want estimates.
2. Measurements: A 1 to 5 column array of input predictors (i.e. measurements) with one row for every row in your coordinates input. This is how you provide the measurement data that you want the LIR to predict the property of interest with. These measurements can be provided in any order.
3. MeasIDVec: This tells the code what the parameters provided in Measurements are. This is a one-row vector with as many elements (i.e. columns) as you had columns in Measurements.

1. Salinity

2. Potential temperature

3. Nitrate

4. AOU

5. Silicate

6. O2 (recommended over AOU)

7. Temperature (recommended over potential temperature)

The rest of the inputs are input pairs that can be provided in any order (as long as the first three inputs are in the correct order). LIPHR has the added complication that it needs to be provided the date information that the user wants the estimates for, or alternately, it needs to be told to ignore the effects of ocean acidification when providing the estimate. This is done by including one of the following two pairs of inputs:

…,’EstDates’,2020,…

…,’OAAdjustTF’,false,…

Read the comments for the other LIR bells and whistles, including:

1. better estimates and estimates of uncertainty when the user provides input uncertainty estimates,
2. the ability to provide inputs in molar units (ml/L O2 not supported),
3. LIPHR can recalculate pH estimates to be suited for calculating AT and DIC (recommended constants: Lueker et al. 2000, Lee et al. 2010 borate, Perez and Fraga 1987 KF). Confused? Modern spectrophotometric pH measurements tend to disagree with pH calculated from these properties.
4. the ability to tell the function to stop printing stuff to the command line.

**Examples:**

For two (made up and potentially implausible) locations with the following property measurements:

1. 0°E 20°N at 1000 m depth (S=35, Si=50 μmol kg-1, O2=100 μmol kg-1, T=3°C)
2. 150°W 45°S at 200 m depth (S=33, Si=50 μmol kg-1, O2=200 μmol kg-1, T=16°C)

Estimates can be obtained for July 1st 2020 using

LINR([0 20 1000;-150 -45 200],[35 50 100 3;33 50 200 16],[1 5 6 7])

1. N=33.8848 μmol kg-1
2. N=4.1740 μmol kg-1

LIAR([0 20 1000;-150 -45 200],[35 50 100 3;33 50 200 16],[1 5 6 7])

1. AT=2328.4156 μmol kg-1
2. AT=2256.0218 μmol kg-1

LIPHR([0 20 1000;-150 -45 200],[35 50 100 3;33 50 200 16],[1 5 6 7],'EstDates',2020.5)

1. pHT=7.7835
2. pHT=7.8015

**Bug fixes from 2018.10.10:**

A bug afflicting all LIRs was identified and corrected. This bug allowed information from the density structure of the Atlantic Ocean into some calculations for the density structure of the Pacific/Southern/Indian Oceans. This information (only) impacted which data were included in coefficient regressions during LIR training. Fortunately, the similarity of the density structures of the deeper ocean basins (deeper because depth matters more than density in the shallow ocean), the broad density windows used, and the regression methodology underpinning LIR routines reduced the impacts of this bug. The following table shows the improvement in the global average RMSE (positive is a smaller RMSE) observed when the bug was fixed. For estimates made with the older code that included the bug, these errors should be added to the published error estimates.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Equation | LIARv2 | LIPHR | LINR | LISIR\* | LIPR\* |
| 1 | 0.1 | 0.0009 | 0.00 | 0.1 | 0.000 |
| 2 | 0.1 | 0.0007 | 0.00 | -0.1 | 0.001 |
| 3 | 0.1 | 0.0011 | 0.00 | 0.1 | 0.000 |
| 4 | 0.1 | 0.0030 | 0.02 | -0.2 | 0.002 |
| 5 | 0.2 | 0.0004 | 0.00 | 0.1 | 0.000 |
| 6 | 0.2 | 0.0025 | 0.00 | -0.1 | 0.001 |
| 7 | 0.2 | 0.0005 | 0.00 | 0.1 | 0.000 |
| 8 | 0.2 | 0.0037 | 0.01 | -0.1 | 0.002 |
| 9 | 0.1 | 0.0006 | 0.01 | 0.1 | 0.000 |
| 10 | 0.1 | 0.0004 | 0.00 | 0.3 | 0.001 |
| 11 | 0.1 | 0.0011 | 0.01 | 0.1 | 0.000 |
| 12 | 0.1 | 0.0031 | 0.02 | 0.3 | 0.002 |
| 13 | 0.3 | 0.0006 | 0.01 | 0.2 | 0.000 |
| 14 | 0.5 | 0.0067 | 0.01 | 0.2 | 0.001 |
| 15 | 0.5 | 0.0010 | 0.02 | 0.1 | 0.000 |
| 16 | 1.7 | 0.0054 | 0.01 | 0.0 | 0.001 |
| \* Still in development… No valid comparison yet exists for LIOR due to concurrent method tweaks, but a comparison between similar methods (with and without the bug fixed) suggests this fix had a small impact for LIOR as well. | | | | | |