BlobMetrics: an analysis framework for StitchBlobs outputs

Contents

1 Minimum software and data requirements

- R software (`https://www.r-project.org/`) and the following libraries:
  - `abind` (`readnetcdf.R`)
  - `akima` (`readnetcdf.R`, `intercomparison.R`)
  - `argparse` (`stitch_metric_framework.R`)
  - `ggplot2` (`generateReport.R`)
  - `gtable` (`generateReport.R`)
  - `grid` (`generateReport.R`)
  - `knitr` (`generateReport.R`)
  - `markdown` (`generateReport.R`)
  - `ncdf4` (`readnetcdf.R`)
  - `ncdf4.helpers` (`readnetcdf.R`)
  - `PCICt` (`readnetcdf.R`)
  - `reshape2` (`intercomparison.R`)
  - `rmarkdown` (`generateReport.R`)
  - `RNetCDF` (`readnetcdf.R`)
- StitchBlobs output (in the form of NetCDF files)
- BlobStats output (in the form of text files)

2 Usage

2.1 Command line usage

The main control framework is run from the command line, with the following syntax:

```
Rscript --vanilla stitch_metric_framework.R [flags]
```

In order to view all possible options, run the above command with the `-h` or `--help` flag:

```
Rscript --vanilla stitch_metric_framework.R -h
```

which will print out the list of available flags and exit. These flags will be explained further in subsequent sections.

2.1.1 Namelists

Each utility must be used in conjunction with a namelist file, which will provide all of the necessary variables that are not specified in the command line. Example namelists are included in the directory with the R function source files.

The `-nl` or `--namelist` flag specifies the master namelist, which contains the necessary variables for all of the framework utilities. However, each utility can be provided with a separate namelist file if desired (flags specified in subsequent sections), which will override inputs from the master namelist.

For example,

```
Rscript --vanilla stitch_metric_framework.R  -nl namelist_master.R -rf
-mt -st -nlst namelist_summarize.R
```

will use `namelist_master.R` for the `--readfiles` and `--mergetable` utilities but `namelist_summarize.R` for `--summarize`

2.2 Example workflow: running BlobMetrics for the first time

After downloading the repository from Github, navigate to the folder containing all of the R scripts.

1. Download the necessary libraries for running the code:

   ```
   Rscript --vanilla download_libraries.R
   ```

   Note that some of the libraries require compiling and installation might fail if the requisite compilers are not available. If a package fails, then the files with the package dependencies (noted next to the package name) will not run.

2. Prepare all of the necessary input files:

   - a list of files containing all of the BlobStats output using StitchBlobs input (with full path names)
   - (optional) a list of files containing all of the BlobStats output using DetectBlobs input (with full path names)
   - a list of files containing all of the StitchBlobs output (with full path names)

3. Generate the template for creating the master namelist file

   ```
   Rscript --vanilla gen_blank_setupfile.R
   ```

   This will return the file `blank_setupfile.R` with all of the required variables. Fill in the desired values and save the modified file with the desired filename.

4. Generate the master namelist file:

   ```
   Rscript --vanilla stitch_metric_framework.R -gn -sl [setup list file]
   ```

   It will create a new directory (if it doesn't yet exist) with the name specified in the setup file; the master namelist file can be found in this directory.

5. Run one or more of the desired utilities:

   ```
   Rscript --vanilla stitch_metric_framework.R [flags] -nl [namelist]
   ```

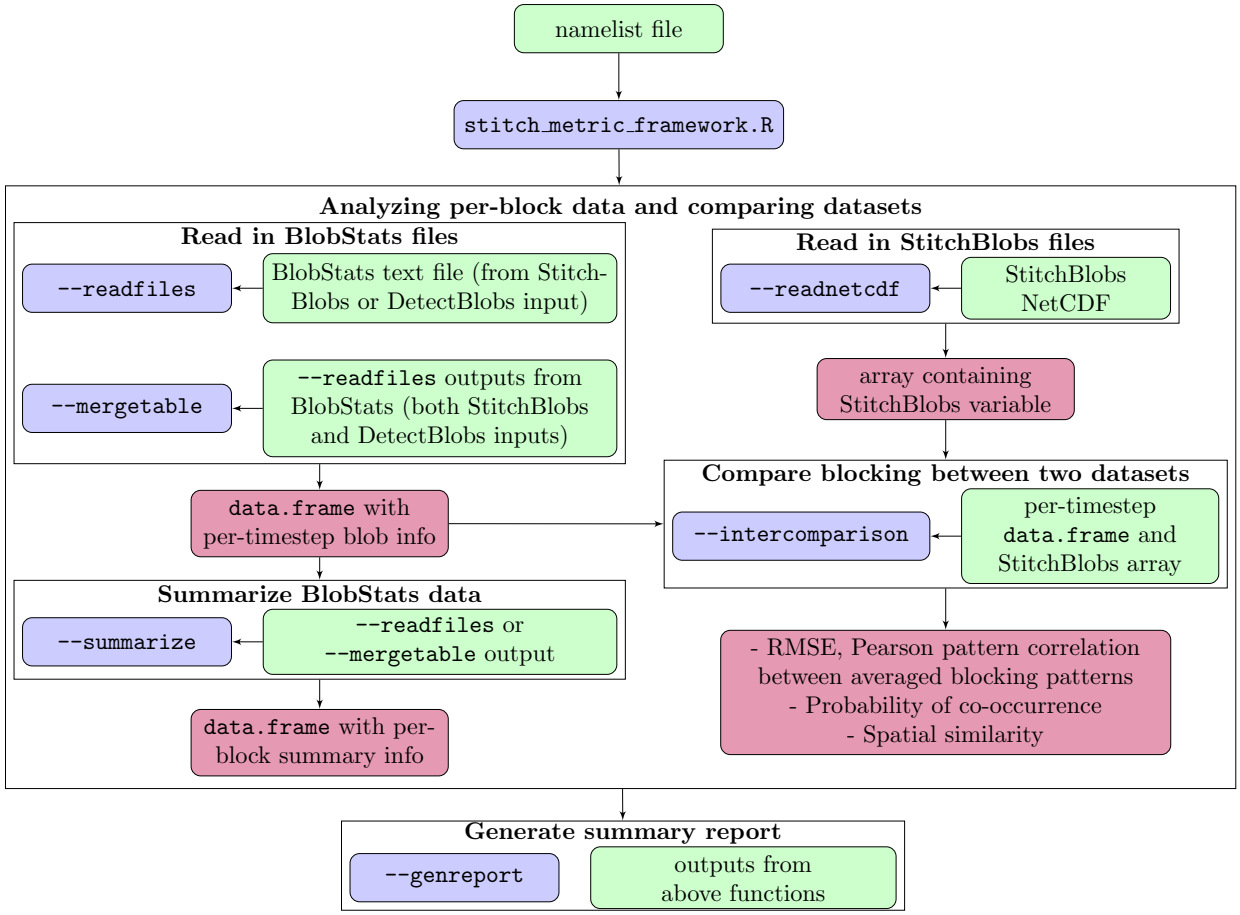

Figure 1: BlobMetrics schematic. Inputs are in green, analysis tools are in purple, and outputs are in pink.

3 BlobMetrics Utilities

BlobMetrics takes StitchBlobs information in the form of both text files containing per-timestep blob information (min/max lat/lon extent, lat/lon center coordinates, and area) as well as the NetCDF files with the original blobs, turns them into R-compatible datasets, and provides summary information about the blocks in an output report. The framework is outlined in Figure 1.

3.1 Read BlobStats files into a single table (`--readfiles`)

This utility takes each BlobStats file and reads the information into a single combined data frame. The columns of the data frame depend upon the included variables in the BlobStats output. Possible variables include `minlat, minlon, maxlat, maxlon, centlat, centlon,` and `area` and are specified when running BlobStats. By default, the outputs are saved in RData format, but there is optional functionality to save this output to text or CSV as well.

3.1.1 Requirements

Text files containing output from BlobStats. The format of these files is explained in more detail in Appendix A.

3.1.2 Command line syntax

```
Rscript --vanilla stitch_metric_framework.R [-rf] [-nl or -nlrf FILE]
```

The following flags are required:

`-rf` (`--readfiles`)
Tell program to read in BlobStats data

`-nl` (`--namelist`) or `-nlrf` (`--namelistrf`) `FILE`
Name of namelist file

3.1.3 Output

An example output `data.frame` looks like this in the R console:

```
             datehour minlat maxlat minlon maxlon centlat centlon    area  area_km bnum var
1 1980-12-01 00:00:00     50     72    187    218    61.0   202.5 0.08299 42330251    1 TM90
2 1980-12-01 06:00:00     50     74    187    222    62.0   204.5 0.08980 45803790    1 TM90

                          file
1   ERA_1980_DJF_NP_Z_stats.txt
2   ERA_1980_DJF_NP_Z_stats.txt
```

`datehour`
The date string in the format `YYYY-MM-DD HH:MM:SS`

`minlat, maxlat, minlon, maxlon, centlat, centlon`
Latitude and longitude coordinates for the block's extent and centroid

`area`
Fractional area of the block

`area_km`
area of the block in km^2

`var`
Algorithm name specified either by the `--algname` flag in the console or `var` in the function (default `VAR`)

`bnum`
The blob ID number as specified in the BlobStats file

`file`
name of the BlobStats file which contains the specified blob information

3.2 Handling instances of merging/splitting blobs (`--mergetable`)

There are some instances where multiple blobs will merge into a single blob at a later date, or a single blob will split off into multiple blobs (this was noted in Sinclair 1995). This can cause BlobStats to produce latitude/longitude blob extents which are much larger than those of each individual blob, and the centroid coordinate will subsequently "jump" a noticeable distance from one time step to the next.

The DetectBlobs binary in TempestExtremes will produce output that is very similar to StitchBlobs output, but provides latitude/longitude extents for each unique feature; blobs which split off from larger features have their own separate identifier.

For example, here is output from StitchBlobs for one blob over 24 hours:

```
           datehour minlat maxlat minlon maxlon centlat centlon    area  area_km var bnum
1980-12-01 00:00:00     50     72    187    218    61.0   202.5 0.08299 42330251   Z    1
1980-12-01 06:00:00     50     74    187    222    62.0   204.5 0.08980 45803790   Z    1
1980-12-01 12:00:00     45     75    139    226    60.0   182.5 0.12303 62753232   Z    1
1980-12-01 18:00:00     43     75    138    231    59.0   184.5 0.13999 71403925   Z    1
```

Here is corresponding output from DetectBlobs:

```
           datehour minlat maxlat minlon maxlon centlat centlon    area  area_km var bnum
1980-12-01 00:00:00     50     72    187    218    61.0   202.5 0.08299 42330251   Z    1
1980-12-01 06:00:00     50     74    187    222    62.0   204.5 0.08980 45803790   Z    2
1980-12-01 12:00:00     50     75    187    226    62.5   206.5 0.09634 49139611   Z    3
1980-12-01 12:00:00     45     55    139    155    50.0   147.0 0.02670 13618721   Z    4
1980-12-01 18:00:00     49     75    186    231    62.0   208.5 0.10133 51684833   Z    5
1980-12-01 18:00:00     43     57    138    157    50.0   147.5 0.03866 19719092   Z    6
```

Note that at times 12Z and 18Z, the detected blob in the StitchBlobs dataset (`bnum 1`) is actually comprised of two blobs, because the smaller blob (`bnum 4` and `6` in the DetectBlobs dataset) is separate from the larger

blob (3 and 5 in the DetectBlobs output) at these time steps, but the smaller blob merges into the larger blob at a later time.

While these merged blobs only made up a small subset of instances in our own dataset, we recognize that this data might skew results with respect to distribution of block size or centroid coordinate. The `--mergetable` utility provides the option to distinguish between the individual blobs within the larger detected region. The summarization utility (Section 3.3), which provides information on each unique block's size, speed, etc will note any instances in which there is blob merging. The user can then choose to keep or omit blobs in which there was merging. As with `--readfiles`

3.2.1 Requirements

Separate files or file lists for StitchBlobs and DetectBlobs data.

3.2.2 Command line syntax

```
Rscript --vanilla stitch_metric_framework.R [-mt] [-nl or -nlmt FILE]
```

The following flags are required:

`-mt` (`--mergetable`)
Tell program to merge the data from StitchBlobs and DetectBlobs

`-nl` (`--namelist`) or `-nlmt` (`--namelistmt`) `FILE`
Name of namelist file

3.2.3 Output

The output `data.frame` looks similar to one returned by the first two methods, with the exception of an additional `bnum2` variable. When `bnum=bnum2`, the latitude/longitude extent is encompassing a single blob.

```
           datehour minlat maxlat minlon maxlon centlat centlon    area   area_km var bnum bnum2
1980-03-06 00:00:00     34     42    191    206    38.0   198.5 0.02822  14394019   Z    1     1
```

When the two do not match, there are multiple blobs contained within the latitude/longitude extent of the original StitchBlobs output.

```
           datehour minlat maxlat minlon maxlon centlat centlon    area   area_km var bnum bnum2
1980-03-08 18:00:00     39     45    136    156    42.0   146.0 0.02500  12751612   Z    1    13
1980-03-08 18:00:00     32     51    194    222    41.5   208.0 0.07581  38667988   Z    1    12
```

3.3 Create a per-blob summary table (`--summarize`)

This utility reads in a data frame with per-timestep information and creates a table that provides per-blob information on quantities such a the blob's starting and ending centroid coordinates, the blob's duration in

days, and others described in more detail below. There is optional functionality to save the output to one of three file types (RData, text, or CSV). If desired, blobs which are comprised of multiple blobs that merge into a single blob are omitted.

3.3.1 Requirements

RData containing output from `--readfiles` or `--mergetable`.

3.3.2 Command line syntax

```
Rscript --vanilla stitch_metric_framework.R [-st] [-nl or -nlst FILE]
```

The following flags are required:

`-st` (`--summarize`)
Tell program to summarize each unique blob's data

`-nl` (`--namelist`) or `-nlst` (`--namelistst`) `FILE`
Name of namelist file

3.3.3 Output

An example summary table looks like this in the R console:

```
             startdate             enddate duration_days merged start_centlat
1  1980-03-06 00:00:00 1980-03-14 12:00:00          8.50    YES          38.0
2  1980-03-17 06:00:00 1980-03-26 00:00:00          8.75     NO          39.5

   start_centlon end_centlat end_centlon   dist_km zonal_dist_km
1          198.5        43.5       194.0  719.2533      379.0267
2          204.0        39.0       208.5  391.4136      387.4485

   zonal_speed_kph min_area  max_area avg_area var bnum
1        1.8579740 12751612  75295717 32349835   Z    1
2        1.8449929 15500859  35505588 32349835   Z    2
```

`startdate, enddate`
The date string in the format `YYYY-MM-DD HH:MM:SS`

`duration_days`
Number of days that block persists

`merged`
Checks whether or not the block extent is the result of multiple blobs merging (see Section 3.2). If this value is `YES`, then it is recommended to check the per-timestep information in order to see when and where the blobs merge, as well as how it affects the calculation of the block size and centroid.

`start_centlat, start_centlon, end_centlat, end_centlon`
start and end coordinates of block centroid.

`dist_km`
Great circle distance from start to end coordinates.

`zonal_dist_km`
Only the zonal component of the distance between the start and end coordinates (calculated using the start and end longitude coordinates of the centroid and the midpoint of the start and end latitude coordinates).

`zonal_speed_kph`
Average speed, in km/hr, of the block's movement. Calculated as distance over duration.

`min_area, max_area, avg_area`
Various information for the block size, in km^2

`var`
Algorithm name (as provided previously when reading in the BlobStats data)

`bnum`
The block ID number from BlobStats (might differ from the ID number of DetectBlobs output)

3.4 Reading in NetCDF data (`--readnetcdf`)

This utility reads data from NetCDF files into arrays in R, combining data from multiple files if desired. There is an optional utility to read in only a geographical subset of the data, as well as the ability to save the specified variables to either a single NetCDF output file or an Rdata file.

3.4.1 Requirements

NetCDF files with the desired variables. Note that if reading multiple files into a single session, they should all contain the variables that are specified in the command line or function call, otherwise the utility will throw an exception.

If the dataset is very large, it might cause R to crash due to memory constraints (although this is dependent upon the available system memory— this scenario is more likely to happen on personal computers). It is recommended to split the data up by time or regional subsets if this is likely to be a problem.

3.4.2 Command line syntax

```
Rscript --vanilla stitch_metric_framework.R [-rn] [-nl or -nlrn FILE]
```

The following flags are required:

`-rn (--readnetcdf)`
Tell program to read NetCDF data into R.

`-nl` (`--namelist`) or `-nlrn` (`--namelistrn`) `FILE`
Name of namelist file

3.4.3 Output

The RData file contains the axis variables (`time_axis, lev_axis, lat_axis, lon_axis`), a vector of string date times (`time_format`), and the variables specified within the variable list.

3.5 Comparing two datasets (`--intercomparison`)

This utility provides information about the amount of agreement between the two datasets with respect to detected blocks. This agreement is quantified by the following metrics:

- **Pearson pattern correlation:** Strength of linear relationship between blocking frequency values at corresponding coordinates for the two datasets. Note: there will be high correlation when the *pattern* is similar, regardless of the relative magnitudes.
- **Root mean square error (RMSE):** Measure of differences in blocking frequency values between the two datasets.
- **Probability of co-occurrence:** The likelihood that a block will appear at a location in dataset 1 given that it also appears at a similar location in dataset 2.
- **Spatial similarity:** When a block is present, amount of field that is commonly designated as blocked by both datasets.

3.5.1 Requirements

Per-timestep blob information (from either `--readfiles` or `--mergetable`) and the corresponding arrays from `--readnetcdf`. Both files should be in RData format.

3.5.2 Command line syntax

```
Rscript --vanilla stitch_metric_framework.R [-ic] [-nl or -nlrn FILE]
```

The following flags are required:

`-ic (--intercomparison)`
Tell program to compare the two datasets specified in the namelist.

`-nl` (`--namelist`) or `-nlic` (`--namelistic`) `FILE`
Name of namelist file

3.5.3 Output

The RData file contains the following variables:

- `df_analyze`: The input data from both datasets (similar to output seen in Section 3.1 or 3.2), combined into a single `data.frame`

- `df_overlaps`: A `data.frame` variable that lists all instances where a blob from dataset 1 overlaps with a blob from dataset 2. An example output, where `V1` denotes the ERA-Interim reanalysis and `V2` denotes the JRA reanalysis, would look like this:

```
                datehour similarity V1bnum V1bnum2 V1minlat V1maxlat V1minlon V1maxlon
6829 2005-02-25 18:00:00  0.8806536      5        5       31       42      162      190
6830 2005-02-26 00:00:00  0.8718225      5        5       30       41      163      191
6831 2005-02-26 06:00:00  0.8957676      5        5       30       41      166      191
6832 2005-02-26 12:00:00  0.8561735      5        5       30       41      168      191
6833 2005-02-26 18:00:00  0.8239728      5        5       31       41      170      191
6834 2005-02-27 00:00:00  0.7644073      5        5       31       40      173      191

     V1centlat V1centlon V2bnum V2bnum2 V2minlat V2maxlat V2minlon V2maxlon V2centlat
6829      36.5     176.0      5       5    31.25    41.25   162.50   190.00    36.250
6830      35.5     177.0      5       5    31.25    41.25   163.75   190.00    36.250
6831      35.5     178.5      5       5    31.25    41.25   166.25   191.25    36.250
6832      35.5     179.5      5       5    31.25    41.25   168.75   191.25    36.250
6833      36.0     180.5      5       5    31.25    40.00   171.25   190.00    35.625
6834      35.5     182.0      5       5    32.50    40.00   173.75   190.00    36.250

     V2centlon
6829   176.250
6830   176.875
6831   178.750
6832   180.000
6833   180.625
6834   181.875
```

- `p1given2` and `p2given1`: probability of co-occurrence for dataset 1 given dataset 2 and the reverse.
- `pearson_num` and `rmse_num`: the Pearson pattern correlation and RMSE coefficients.
- `sim_25`, `sim_50` and `sim_75`: the 25th, 50th, and 75th percentile values of spatial similarity between the two datasets.
- `V1` and `V2`: The names of your variables for dataset 1 and dataset 2.

A BlobStats File Format

Each BlobStats file is formatted as follows:

Line 1: Tab-separated column names

Blob information line: `Blob IDNUM (NUM_TIMESTEPS)` where `IDNUM` is the blob's unique identifier number and `NUM_TIMESTEPS` is the number of timesteps in the blob's lifespan.

Per-timestep blob information: Always contains the timestep number in column 1. The other columns depend on the included variables.

For example, a BlobStats file with two Blobs, each with a lifetime of 2 time steps, would look like this:

```
time,minlat,maxlat,minlon,maxlon,centlat,centlon,area
Blob 1 (2)
1992-03-13-43200        33.00000        43.00000        130.00000       142.00000       38.00000
1992-03-13-64800        33.00000        43.00000        130.00000       144.00000       38.00000
Blob 2 (2)
1992-03-24-64800        28.00000        36.00000        132.00000       149.00000       32.00000
1992-03-25-00000        27.00000        37.00000        131.00000       151.00000       32.00000
```