BlobMetrics: an analysis framework for StitchBlobs outputs

Contents

1 Minimum Requirements

- R software (`https://www.r-project.org/`) and the following libraries:
  - abind
  - argparse
  - RNetCDF (`https://journal.r-project.org/archive/2013/RJ-2013-023/RJ-2013-023.pdf`)
- StitchBlobs output (in the form of NetCDF files)
- BlobStats output (in the form of text files)

If the required libraries have not yet been installed in R, then an internet connection will be necessary, as R will attempt to download the missing libraries from online repositories before running the code.

2 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 optional flags and exit. These flags will be explained further in subsequent sections.

These various tools can also be utilized in an interactive R session by opening R and loading the various functions from the source R scripts. For example, to read a list of BlobStats files into a single data table, type the following commands into R:

```
source("read_stitch.R")
file_list<-readLines("list_of_blobfiles.txt")
table_stats<-read_stats_to_table(file_list,6,var="TM90")
```

The `source` command reads in all of the commands from the R script `read_stitch.R`, which loads the function `read_stats_to_table`. This function takes an existing text file (`list_of_blobfiles.txt`) containing a list of BlobStats file names, loads the file names into a vector of strings, and reads all of those files into a data frame that is named `table_stats`. `6` refers to the number of hours per time step in the original data and `TM90` refers to the block detection algorithm (Tibaldi and Molteni 1990) that was used to produce input files for StitchBlobs.

3 Read BlobStats files into a single table

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. There is optional functionality to save the output to one of three file types (RData, text, or CSV).

3.1 Caveat

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.

While these merged blobs only made up a tiny 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; therefore, there is a separate `DetectBlobs` binary which will produce output that is very similar to `StitchBlobs` output, but does not stitch the blobs together across the time axis.

3.2 Command line syntax

```
Rscript --vanilla stitch_metric_framework.R [-rf] [-fn or -fl FILE] [-th TIMERES]

(optional)
[-an ALGORITHM] [--rtable OUTPUT] [--texttable OUTPUT] [--csvtable OUTPUT]
```

The following flags are required:

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

`-fn` (`--filename`) or `-fl` (`--filelist`) `FILE`
Name of single input BlobStats output, or a text file containing a list of BlobStats file names.

`-th` (`--thourly`) `TIMERES`
Time resolution in terms of hours (i.e. 6 for 6 hourly)

The following flags are optional:

`-an` (`--algname`) `ALGORITHM`
Name of the objective blocking detection algorithm used to produce input files for StitchBlobs. Recommended to use this flag if you will be looking at outputs from multiple algorithms, as it will help to differentiate between datasets if you combine multiple data frames into a single file (see Section 4).

`--rtable OUTPUT`
File name for output RData file. The result will be an RData table containing the data frame `df_tot`.

`--texttable OUTPUT`
File name for output text file with tab-separated columns.

`--csvtable OUTPUT`
File name for output CSV file.

3.3 Function syntax

To load and use this function in an interactive R session, do

```
source("read_stitch.R")
desired_name<-read_stats_to_table(flist,nhrs,..)
```

which will produce a `data.frame` object with the variable name `desired_name`.

The following arguments are required:

`flist`
a vector object containing the BlobStats file names.

`nhrs`
Time resolution in terms of hours (i.e. 6 for 6 hourly)

The following arguments are optional:

`var`
Name of the objective blocking detection algorithm used to produce input files for StitchBlobs. If left blank, a column `var` will be filled with the string `VAR`

`rfn, textfn, csvfn`
Strings specifying output file names for RData, text, and CSV file formats. If left blank, the function will merely return the `data.frame` object to the console.

3.4 Output

An example output data table 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`

`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 `-an` 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

4 Read previously created data table(s) into an R session

This utility reads previously created data tables into the R session, combining tables into one single table. This function is particularly useful if attempting to examine data from multiple detection algorithms. There is optional functionality to save the output to one of three file types (RData, text, or CSV).

4.1 Command line syntax

```
Rscript --vanilla stitch_metric_framework.R [-rt] [-fn or -fl FILE]
[--isr or --istext or --iscsv]


(optional)
[--rtable OUTPUT] [--texttable OUTPUT] [--csvtable OUTPUT]
```

The following flags are required:

`-rt` (`--readtable`)
Tell program to read in BlobStats data

`-fn` (`--filename`) or `-fl` (`--filelist`) `FILE`
Name of single file containing output from `--readfiles`, or a text file containing a list of files with output from `--readfiles`.

`--isr`
Boolean telling the program that the input is in RData format

`--istext`
Boolean telling the program that the input is in text file format

`--iscsv`
Boolean telling the program that the input is in CSV format

Only specify one of the Boolean flags! (This program currently does not have the capability to handle multiple formats.)

The following flags are optional:

`--rtable OUTPUT`
File name for output RData file. The result will be an RData table containing the data frame `df_data`.

`--texttable OUTPUT`
File name for output text file with tab-separated columns.

`--csvtable OUTPUT`
File name for output CSV file.

4.2 Function syntax

To load this function in an interactive R session, do

```
source("combine_tables.R")
desired_name<-combine_dfs(flist,ftype,...)
```

which will combine the loaded `data.frame` objects from each file into a single `data.frame` object with the variable name `desired_name`.

The following arguments are required:

`flist`
a vector object containing the BlobStats file names.

`ftype`
File format of input files (specify one of three strings: `"R"`, `"text"`, or `"CSV"`)

The following arguments are optional:

`rfn, textfn, csvfn`
Strings specifying output file names for RData, text, and CSV file formats. If left blank, the function will merely return the `data.frame` object to the console.

4.3 Output

The output is identical to that seen in Section 3.4, but column values for the `var` column might vary.

5 Create a per-blob summary table

This utility reads in a data table (using the method outlined in Section 4) 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).

5.1 Command line syntax

The `-st` flag is combined with either the `--readfiles` (Section 3) or `--readtable` (Section 4) commands.

```
Rscript --vanilla stitch_metric_framework.R [-st] [options from --readfiles or --readtable]
```

The following flags are required:

`-st` (`--summarize`)
Tell program to summarize the data table output(s) from reading in BlobStats data.

`-rt` (`--readfiles`) or `-rt` (`--readtable`)
Tell program to read in BlobStats data.

`-fn` (`--filename`) or `-fl` (`--filelist`) `FILE`
Name of single file, or a text file containing a list of files.

If using the `--readtable` option:

`--isr`
Boolean telling the program that the input is in RData format

`--istext`
Boolean telling the program that the input is in text file format

`--iscsv`
Boolean telling the program that the input is in CSV format

Only specify one of the Boolean flags! (This program currently does not have the capability to handle multiple formats.)

The following flags are optional:

`--rsumm OUTPUT`
File name for output RData file. The result will be an RData table containing the data frame `df_summ`.

`--textsumm OUTPUT`
File name for output text file with tab-separated columns.

`--csvsumm OUTPUT`
File name for output CSV file.

5.2 Function syntax

To load this function in an interactive R session, do

```
source("summ_table.R")
desired_name<-gen_summary_table(df_in,...)
```

which will summarize each unique blob in the input `data.frame` object and produce an output `data.frame` object with the variable name `desired_name`.

The following arguments are required:

`df_in`
Name of the input data frame (create using methods from Section 3 or 4).

The following arguments are optional:

`rfn, textfn, csvfn`
Strings specifying output file names for RData, text, and CSV file formats. If left blank, the function will merely return the `data.frame` object to the console.

5.3 Output

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

```
            startdate             enddate duration_days start_centlat start_centlon end_centlat
1 1980-12-01 00:00:00 1980-12-13 18:00:00         12.75          61.0         202.5        57.0
2 1980-12-14 06:00:00 1980-12-29 18:00:00         15.50          43.5         239.5        57.0

  end_centlon  dist_km zonal_dist_km zonal_speed_kph min_area  max_area avg_area bnum
1       152.5 2825.715     2795.8618        9.136803 24503497 105945491 48812899    1
2       144.0 6377.548     6282.5810       16.888658 13394293  93958976 48812899    2

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

A BlobStats File Format

Each BlobStats file is formatted as follows:

Line 1: Date of first time step in format `YYYY-MM-DD`

Line 2: 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:

```
1980-12-01
Time    minlat  maxlat  minlon  maxlon  centlat centlon area
Blob 1 (2)
1       50.00000        74.00000        187.00000       222.00000       62.00000        204.50000
2       45.00000        75.00000        139.00000       226.00000       60.00000        182.50000
Blob 2 (2)
53      39.00000        48.00000        226.00000       253.00000       43.50000        239.50000
54      36.00000        49.00000        221.00000       254.00000       42.50000        237.50000
```