

---

# BlobMetrics: an analysis framework for StitchBlobs outputs

## Contents

<b>1</b>	<b>Minimum software and data requirements</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>3</b>
2.1	Command line usage . . . . .	3
2.1.1	Namelist . . . . .	4
<b>3</b>	<b>Read BlobStats files into a single table</b>	<b>4</b>
3.1	Requirements . . . . .	4
3.2	Command line syntax . . . . .	5
3.3	Output . . . . .	5
<b>4</b>	<b>Handling instances of merging/splitting blobs</b>	<b>5</b>
4.1	Requirements . . . . .	6
4.2	Command line syntax . . . . .	6
4.2.1	Required namelist variables . . . . .	7
4.3	Function syntax . . . . .	7
4.4	Output . . . . .	8
<b>5</b>	<b>Create a per-blob summary table</b>	<b>8</b>
5.1	Requirements . . . . .	8
5.2	Command line syntax . . . . .	8
5.2.1	Required namelist variables . . . . .	9
5.3	Function syntax . . . . .	9
5.4	Output . . . . .	10

---

<b>6</b>	<b>Reading in NetCDF data</b>	<b>10</b>
6.1	Requirements . . . . .	11
6.2	Command line syntax . . . . .	11
6.2.1	Required namelist variables . . . . .	11
6.3	Function syntax . . . . .	12
6.4	Output . . . . .	12
6.4.1	NetCDF . . . . .	13
6.4.2	RData . . . . .	13
<b>A</b>	<b>BlobStats File Format</b>	<b>14</b>

---

# 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)

Before using this code, navigate to the directory where all of the code is stored and run the following command:

```
Rscript --vanilla setup.R
```

which should download all of the necessary libraries. 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 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`

There is an option to generate a master namelist file for all of the various analysis operations, using an initial list of variables. A template for this initial list is generated by running the command

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

which returns the file `blank_setupfile.R` with all of the required variables. Fill in the desired values and then save the modified file with your desired name (example, `setup_reanalysis.R`)

Then, run the command

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

which will return a master namelist file with all of the variables filled in. Examples of `blank_setupfile.R`, a modified version with all of the variables filled in (`setup_full.R`), and the resultant master namelist (`DJF_NP_namelist_master_test.R`) can be found in the `STITCH_METRICS` directory.

## 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 this output to one of three file types (RData, text, or CSV).

### 3.1 Requirements

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

---

## 3.2 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
```

## 3.3 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<sup>2</sup>

**var**

Algorithm name specified either by the `--alname` 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 Handling instances of merging/splitting blobs

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	bnun
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	bnun
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 (**bnun** 1) is actually comprised of two blobs, because the smaller blob (**bnun** 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; therefore, we provide this extra functionality to distinguish between the individual blobs within the larger detected region. The summarization utility (Section 5), 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.

## 4.1 Requirements

Separate files or file lists for StitchBlobs and DetectBlobs data. If using the interactive R session, two separate data tables must first be produced using the method outlined in Section 3 or Section ??).

## 4.2 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

#### 4.2.1 Required namelist variables

`nrun_mt`  
Number of times to run `--mergetable`

`ftype_mt`  
Input file type ("R", "text", or "CSV")

`stitch_file` or `stitch_list`  
Single file or list of files containing data from StitchBlobs (generated using either `--readfiles` or `--readtable`)

`detect_file` or `detect_list`  
Single file or list of files containing data from DetectBlobs (generated using either `--readfiles` or `--readtable`)

`rfn_merged`, `txt_merged`, `csv_merged`  
Optional output file names in RData, text, or CSV format (store the respective variable as a blank string to suppress output for that particular file format)

`df_merged`  
Optional variable name for the output data frame in the RData file. Defaults to `df_merged` if string is left blank.

#### 4.3 Function syntax

To load this function in an interactive R session, do

```
source("mergetable.R")
desired_name<-merge_dfs(df_stitch,df_nostitch,...)
```

which will produce a `data.frame` object with the variable name `desired_name`. Note that `df_stitch` and `df_nostitch` will need to be created using either `read_table` or `combine_tables`.

The following arguments are required:

`df_stitch`  
a data frame (created using `read_table` or `combine_tables`) containing BlobStats output with StitchBlobs data

`df_nostitch`  
a data frame (created using `read_table` or `combine_tables`) containing BlobStats output with DetectBlobs data

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.

## 4.4 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

## 5 Create a per-blob summary table

This utility reads in a data frame with per-timestep information and creates a table that provides per-blob information on quantities such as 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.

### 5.1 Requirements

### 5.2 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



---

### 5.2.1 Required namelist variables

**nrun\_st**

Number of times to run `--summarize`

**ftype\_st**

Input file type ("R", "text", or "CSV")

**filename\_summ** or **filelist\_summ**

Single file or list of files containing data from StitchBlobs (generated using either `--readfiles` or `--readtable`)

**keep\_merge**

Keep or omit blobs which are comprised of multiple blobs? If `TRUE`, all blobs are kept; if `FALSE`, blobs where the `merged` column has a value of `YES` are omitted from the output.

**rfn\_summ**, **txt\_summ**, **csv\_summ**

Optional output file names in RData, text, or CSV format (store the respective variable as a blank string to suppress output for that particular file format)

**df\_summ**

Optional variable name for the output data frame in the RData file. Defaults to `df_summ` if string is left blank.

## 5.3 Function syntax

To load this function in an interactive R session, do

```
source("summarize.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`. Note that `df_in` will need to be created using either `read_table` or `combine_tables`.

The following arguments are required:

**df\_in**

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

The following arguments are optional:

**keep\_merge**

Default is `TRUE`; if set to `FALSE`, the final data table will not contain merged blobs.

**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.

---

## 5.4 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 4). 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<sup>2</sup>

**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)

## 6 Reading in NetCDF data

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.

---

## 6.1 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.

## 6.2 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
```

### 6.2.1 Required namelist variables

```
nrun_rn
Number of times to run --readnetcdf

filename_netcdf or filelist_netcdf
Single NetCDF file name or list of NetCDF file names

varvec
Vector of variable names to be read to R in the format c("VAR1", "VAR2", "VAR3"...)

outvec
Vector of output variable names (must be same length as varvec)

outrdata, outnetcdf
Optional output file names in RData or NetCDF format

timename
Name of time axis

levname
Name of vertical axis

latname
Name of latitude axis

lonname
Name of longitude axis

minlat, maxlat, minlon, maxlon
subsetting boundaries for horizontal (lat/lon) direction; if not subsetting, set variable to blank string
("")
```

---

`minlev,maxlev`

If applicable, subsetting boundaries for the vertical axis. If subsetting to a single level (i.e. 500 mb) set both `minlev` and `maxlev` to that value. If vertical axis doesn't exist, set variable to blank string (`""`)

### 6.3 Function syntax

To load this function in an interactive R session, do

```
source("read_netCDF_to_R.R")
vlist<-c("VAR1","VAR2","VAR3")
vars_output<-read_netcdf(flist,vlist,...)
```

The following arguments are required:

`flist`

Vector containing input NetCDF file names

`vlist`

Vector containing variable names that will be read into R

which will read all of the NetCDF variables into R matrices and return a `list` object with the name `vars_output` containing all of the resulting output variables, as well as the axes for time, latitude, longitude, etc.

The following arguments are optional:

`olist`

Vector containing variable names of output variables (default is the input variable list)

`timename, levname, latname, lonname`

Names of the axis variables (defaults are `"time"`, `"lev"`, `"lat"`, `"lon"`)

`minlat, maxlat, minlon, maxlon, minlev, maxlev`

Coordinates of lat/lon extent and vertical levels, if subsetting.

`ncout, rdataout`

Strings specifying output file names for NetCDF and RData file formats. If left blank, the function will return a `list` object that contains all of the variables.

### 6.4 Output

If neither `ncout` nor `rdataout` are specified, the function returns a `list` object (here named `vars_output` as per the example).

Each variable stored within `vars_output` can be accessed via `vars_output[["VAR"]]`. In order to turn the `vars_output` object into distinct R variables that are accessible from the global environment, do the following:

---

```
list_variables<-names(vars_output)
for (v in list_variables){
  assign(v,vars_output[[v]])
}
remove(vars_output)
```

`list_variables` is a vector of names of all of the variables that are contained within `vars_output`. This sequence of commands will store each variable in `vars_output` in the global environment, then delete the `list` object (for space saving purposes).

#### 6.4.1 NetCDF

The returned NetCDF will contain the variables specified by the user, with axes `time_axis`, `lat_axis`, `lon_axis` and, optionally, `lev_axis` (not all variables are on multiple vertical levels). Currently, the output NetCDF does not contain the metadata from the original file, although this might change in the future.

`time_axis` is in units of `hours` since 1800-01-01 00:00 regardless of the original time units.

#### 6.4.2 RData

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.

---

## 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
```