

---

# TempestExtremes

## 1 DetectCyclonesUnstructured

Usage: DetectCyclonesUnstructured <parameter list>

Parameters:

```
--in_data <string> [""]
--in_connect <string> [""]
--out <string> [""]
--searchbymin <string> [""] (default PSL)
--searchbymax <string> [""]
--maxlat <double> [0.000000] (degrees)
--minlat <double> [0.000000] (degrees)
--topofile <string> [""]
--maxtopoht <double> [0.000000] (m)
--mergedist <double> [0.000000] (degrees)
--closedcontourcmd <string> [""] [var,dist,delta,minmaxdist;...]
--noclosedcontourcmd <string> [""] [var,dist,delta,minmaxdist;...]
--thresholdcmd <string> [""] [var,op,value,dist;...]
--outputcmd <string> [""] [var,op,dist;...]
--timestride <integer> [1]
--regional <bool> [false]
--out_header <bool> [false]
--verbosity <integer> [0]
```

**--in\_data <string>**

The input datafile in NetCDF format.

**--in\_connect <string>**

A connectivity file, which uses a vertex list to describe the graph structure of the input grid. This parameter is not required if the data is on a latitude-longitude grid.

**--out <string>**

The output file containing the filtered list of candidates in plain text format.

**--searchbymin <string>**

The input variable to use for initially selecting candidate points (defined as local minima). By default this is “PSL”, representing detection of surface pressure minima. Only one of **searchbymin** and **searchbymax** may be set.

**--searchbymax <string>**

The input variable to use for initially selecting candidate points (defined as local maxima). Only one of **searchbymin** and **searchbymax** may be set.

**--maxlat <double>**

The maximum absolute latitude for candidate points. Candidates at higher latitudes are discarded.

**--minlat <double>**

The minimum absolute latitude for candidate points. Candidates at lower latitudes are discarded.

**--topofile <string>**

An auxiliary file containing topographic information.

**--maxtopoht <double>**

If **--topofile** is specified, discard candidates over topography higher than specified by this parameter.

---

**--mergedist <double>**

Merge candidate points with distance (in degrees) shorter than the specified value. Among two candidates within the merge distance, only the candidate with lowest **searchbymin** or highest **searchbymax** value will be retained.

**--closedcontourcmd <cmd1>;<cmd2>;...** Eliminate candidates if they do not have a closed contour. Closed contour commands are separated by a semi-colon. Each closed contour command takes the form **var,dist,delta,pivotdist**. These arguments are as follows.

**var <variable>** The variable used for the contour search.

**dist <double>** The great-circle distance (in degrees) from the pivot within which the closed contour criteria must be satisfied.

**delta <double>** The amount by which the field must change from the pivot value. If positive (negative) the field must increase (decrease) by this value along the contour.

**pivotdist <double>** The distance away from the candidate to search for the pivot. If **delta** is positive (negative), the pivot is a local minimum (maximum).

**--noclosedcontourcmd <cmd1>;<cmd2>;...**

As **closedcontourcmd**, except eliminates candidates if a closed contour is present.

**--thresholdcmd <cmd1>;<cmd2>;...** Eliminate candidates that do not satisfy a threshold criteria (there must exist a point within a given distance of the candidate that satisfies a given equality or inequality). Threshold commands are separated by a semi-colon. Each threshold command takes the form **var,op,value,dist**. These arguments are as follows.

**var <variable>** The variable used for the contour search.

**op <string>** Operator that must be satisfied for threshold (options include **>**, **>=**, **<**, **<=**, **=**, **!=**).

**value <double>** The value on the RHS of the comparison.

**dist <double>** The great circle distance away from the candidate to search for a point that satisfies the threshold (in degrees).

**--outputcmd <cmd1>;<cmd2>;...** Include additional columns in the output file. Output commands take the form **var,op,dist**. These arguments are as follows.

**var <variable>** The variable used for the contour search.

**op <string>** Operator that is applied over all points within the specified distance of the candidate (options include **max**, **min**, **avg**, **maxdist**, **mindist**).

**dist <double>** The great circle distance away from the candidate wherein the operator is applied (in degrees).

**--timestride <integer>**

Only examine discrete times at the given stride (by default 1).

**--regional**

When a latitude-longitude grid is employed, do not consider longitudinal boundaries to be periodic.

**--out\_header**

Output a header describing the columns of the data file.

**--verbosity <integer>**

Set the verbosity level (default 0).

## 1.1 Variable Specification

Quantities of type **<variable>** include both NetCDF variables in the input file (for example, "Z850") and simple operations performed on those variables. By default it is assumed that NetCDF variables are specified in the **.nc** file as

---

```
float Z850(time, lat, lon) or float Z850(time, ncol)
```

for structured latitude-longitude grids and unstructured grids, respectively. If variables have no time variable, they have the related specification

```
float Z850(lat, lon) or float Z850(ncol)
```

If variables include an additional dimension, for instance,

```
float Z(time, lev, lat, lon) or float Z(time, lev, ncol)
```

they may be specified on the command-line as `Z(<lev>)`, where the integer index `<lev>` corresponds to the first dimension (or the dimension after `time`, if present).

Simple operations on variables are also supported, including

```
_VECMAG(<variable>, <variable>) 2-component vector magnitude,  
_PLUS(<variable>, <variable>) Pointwise sum of variables,  
_DIFF(<variable>, <variable>) Pointwise difference of variables.
```

The following are valid examples of `<variable>` type,

```
_VECMAG(U850, V850) and _DIFF(U(3),U(5)).
```

## 2 StitchNodes

Usage: `StitchNodes <parameter list>`

Parameters:

```
--in <string> [""]  
--out <string> [""]  
--format <string> ["no,i,j,lon,lat"]  
--range <double> [5.000000] (degrees)  
--minlength <integer> [3]  
--min_endpoint_dist <double> [0.000000] (degrees)  
--min_path_dist <double> [0.000000] (degrees)  
--maxgap <integer> [0]  
--threshold <string> [""] [col,op,value,count;...]  
--timestep <integer> [1]  
--out_format <string> ["std"] (std|visit)
```

```
--in <string>
```

The input file (a list of candidates from `DetectCyclonesUnstructured`).

```
--out <string>
```

The output file containing the filtered list of candidates in plain text format.

---

`--format <string>`

The structure of the columns of the input file.

`--range <double>`

The maximum distance between candidates along a path.

`--minlength <integer>`

The minimum length of a path (in terms of number of discrete times).

`--min_endpoint_dist <double>`

The minimum great-circle distance between the first candidate on a path and the last candidate (in degrees).

`--min_path_dist <double>`

The minimum path length, defined as the sum of all great-circle distances between candidate nodes (in degrees).

`--maxgap <integer>`

The largest gap (missing candidate nodes) along the path (in discrete time points).

`--threshold <cmd1>;<cmd2>;...`

Eliminate paths that do not satisfy a threshold criteria (a specified number of candidates along path must satisfy an equality or inequality). Threshold commands are separated by a semi-colon. Each threshold command takes the form `col,op,value,count`. These arguments are as follows.

`col <integer>` The column in the input file to use in the threshold criteria.

`op <string>` Operator used for comparison of column value (options include `>`, `>=`, `<`, `<=`, `=`, `!=`).

`value <double>` The value on the right-hand-side of the operator.

`count <integer>` The minimum number of candidates along the path that must satisfy this criteria.

`--timestep <integer>`

Only examine discrete times at the given stride (by default 1).