Model Evaluation Tools – Tropical Cyclone Version 5.0 (MET-TC)

User's Guide

Developmental Testbed Center
Research Applications Laboratory
National Center for Atmospheric Research

Boulder, CO

September 2014

Table of Contents

Foreword

This user's guide is provided as an aid to users of the Model Evaluation Tools – Tropical Cyclone (MET-TC). MET-TC is a set of tropical cyclone specific verification tools developed by the Developmental Testbed Center (DTC) for use by the hurricane modeling community. This module is included in the MET v5.0 release.

The first release of MET-TC (v4.1) was intended to replicate the functionality of the verification software developed at the National Hurricane Center (NHC). Additional capabilities have been added to incorporate user needs and advancements in TC verification. This software is primarily intended for operational purposes, but is also utilized for research purposes in the hurricane modeling community. The purpose of the MET-TC software is to replicate the functionally of the NHC verification system, with a modular design allowing additional capabilities and features to be added in future releases. The goal of this toolkit is to provide a standard set of verification metrics and comprehensive output statistics, which are available to all users to enable consistent forecast evaluation studies.

MET-TC is an evolving module within the MET software package. The first version of MET-TC was released in May 2013 with MET v4.1. The second version of MET-TC was released in August 2014. The updates for MET-TC v5.0 primarily involve adding additional flags and added flexibility to each of the MET-TC modules to enhance user capabilities. Future releases are expected to include new capabilities and enhancements to the MET-TC tool.

Contributors to this user's guide include: Kathryn Newman, John Halley Gotway, Tressa Fowler, Paul Kucera and Barbara Brown.

TERMS OF USE

IMPORTANT!
USE OF THIS SOFTWARE IS SUBJECT TO THE FOLLOWING TERMS AND CONDITIONS:

1. **License**. Subject to these terms and conditions, University Corporation for Atmospheric Research (UCAR) grants you a non-exclusive, royalty-free license to use, create derivative works, publish, distribute, disseminate, transfer, modify, revise and copy the Model Evaluation Tools – Tropical Cyclone (MET-TC) software, in both object and source code (the "Software").

You shall not sell, license or transfer for a fee the Software, or any work that in any manner contains the Software.

2. **Disclaimer of Warranty on Software**. Use of the Software is at your sole risk. The Software is provided "AS IS" and without warranty of any kind and UCAR EXPRESSLY DISCLAIMS ALL WARRANTIES AND/OR CONDITIONS OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT OF A THIRD PARTY'S INTELLECTUAL PROPERTY, MERCHANTABILITY OR SATISFACTORY QUALITY AND FITNESS FOR A PARTICULAR PURPOSE. THE PARTIES EXPRESSLY DISCLAIM THAT THE UNIFORM COMPUTER INFORMATION TRANSACTIONS ACT (UCITA) APPLIES TO OR GOVERNS THIS AGREEMENT. No oral or written information or advice given by UCAR or a UCAR authorized representative shall create a warranty or in any way increase the scope of this warranty. Should the Software prove defective, you (and neither UCAR nor any UCAR representative) assume the cost of all necessary correction.

3. **Limitation of Liability**. UNDER NO CIRCUMSTANCES, INCLUDING NEGLIGENCE, SHALL UCAR BE LIABLE FOR ANY DIRECT, INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES INCLUDING LOST REVENUE, PROFIT OR DATA, WHETHER IN AN ACTION IN CONTRACT OR TORT ARISING OUT OF OR RELATING TO THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF UCAR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

4. **Compliance with Law**. All Software and any technical data delivered under this Agreement are subject to U.S. export control laws and may be subject to export or import regulations in other countries. You agree to comply strictly with all applicable laws and regulations in connection with use and distribution of the Software, including export control laws, and you acknowledge that you have responsibility to obtain any required license to export, re-export, or import as may be required.

5. **No Endorsement/No Support**. The names UCAR/NCAR, National Center for Atmospheric Research and the University Corporation for Atmospheric Research may not be used in any advertising or publicity to endorse or promote any products or commercial entity unless specific written permission is obtained from UCAR. The Software is provided without any support or maintenance, and without any obligation to provide you with modifications, improvements, enhancements, or updates of the Software.

6. **Controlling Law and Severability**. This Agreement shall be governed by the laws of the United States and the State of Colorado. If for any reason a court of competent

jurisdiction finds any provision, or portion thereof, to be unenforceable, the remainder of this Agreement shall continue in full force and effect. This Agreement shall not be governed by the United Nations Convention on Contracts for the International Sale of Goods, the application of which is hereby expressly excluded.

7. **Termination**. Your rights under this Agreement will terminate automatically without notice from UCAR if you fail to comply with any term(s) of this Agreement. You may terminate this Agreement at any time by destroying the Software and any related documentation and any complete or partial copies thereof. Upon termination, all rights granted under this Agreement shall terminate. The following provisions shall survive termination: Sections 2, 3, 6 and 9.

8. **Complete Agreement**. This Agreement constitutes the entire agreement between the parties with respect to the use of the Software and supersedes all prior or contemporaneous understandings regarding such subject matter. No amendment to or modification of this Agreement will be binding unless in writing and signed by UCAR.

9. **Notices and Additional Terms**. Copyright in Software is held by UCAR. You must include, with each copy of the Software and associated documentation, a copy of this Agreement and the following notice:

"The source of this material is the Research Applications Laboratory at the National Center for Atmospheric Research, a program of the University Corporation for Atmospheric Research (UCAR) pursuant to a Cooperative Agreement with the National Science Foundation; ©2007
University Corporation for Atmospheric Research. All Rights Reserved."

The following notice shall be displayed on any scholarly works associated with, related to or derived from the Software:

"Model Evaluation Tools – Tropical Cyclone (MET-TC) was developed at the National Center for Atmospheric Research (NCAR) through a grant from the National Oceanic and Atmospheric Administration (NOAA) through the Hurricane Forecast Improvement Project (HFIP). Model Evaluation Tools (MET) was developed through a grant from the United States Air Force Weather Agency (AFWA). NCAR is sponsored by the United States National Science Foundation."

By using or downloading the Software, you agree to be bound by the terms and conditions of this Agreement.

Acknowledgments

Funding for the development of MET-TC is from the National Oceanic and Atmospheric Administration (NOAA)'s Hurricane Forecast Improvement Project (HFIP) through the Developmental Testbed Center (DTC). The DTC is funded by the NOAA, the Air Force Weather Agency (AFWA), and the National Science Foundation (NSF). The National Center for Atmospheric Research (NCAR) is sponsored by NSF. We would like to thank James Franklin at the National Hurricane Center (NHC) for his insight into the original development of the existing NHC verification software.

Chapter 1 – Overview of MET-TC

1.1 Purpose of organization of the User's Guide

The purpose of this User's Guide is to provide basic information to the users of the Model Evaluation Tools – Tropical Cyclone (MET-TC) to enable users to apply MET-TC to their tropical cyclone datasets and evaluation studies. MET-TC is intended for use with model forecasts run through a vortex tracking software or with operational model forecasts in Automated Tropical Cyclone Forecast (ATCF) file format.

The User's Guide is organized as follows. Chapter 1 provides an overview of MET-TC and its components, as well as basic information on the software build. Chapter 2 describes the required input, including file format and the MET-TC output. Chapters 3 through 5 describe the TC-dland tool, TC-Pairs, and TC-Stat tools, respectively. These chapters cover the input and output, and practical usage including a description of the configuration files. Chapter 6 gives a short overview of graphical utilities available within the MET-TC release.

1.2 MET-TC components

The MET-TC software is included as a module within the full Model Evaluation Tools (MET) v5.0. Figure 1.1 shows the basic stages of the MET-TC module. The required input and output, as well as each separate tool are described in more detail in later chapters.

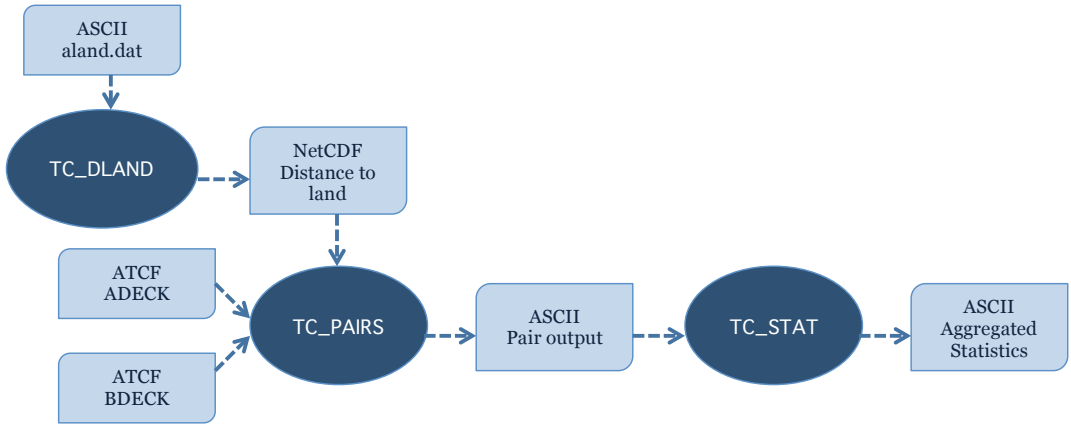

Figure 1.1: Basic representation of the structure of the MET-TC module. Dark blue areas represent each executable and light blue areas represent required input and output to MET-TC.

The TC-dland tool is used to generate a gridded file that determines the location of coastlines and islands, and is used as input to the TC-Pairs tool to determine the distance from land of a particular track point. The TC-Pairs tool matches pairs of input model data and best track (or any reference forecast) and calculates position errors. The TC-Stat tool uses the TC-Pairs output to perform filter and summary jobs over the matched pair dataset.

1.3 Software installation

MET-TC was developed to utilize the software framework of MET, therefore MET-TC is available as a module included in the overall MET package. For more information on the required and recommended libraries, how to install MET, the MET directory structure, and supported architectures, refer to Chapter 2 of the MET Users' Guide.

Within the full directory structure of MET, locations of MET-TC specific code and tools include:

The installed **bin/** directory will contain executables for each module of MET once it has been successfully built. MET-TC modules include: `tc_dland, tc_pairs, tc_stat`.

The installed **share/met/config/** directory contains several configuration and static data files used by MET. For MET-TC, the `TCPairsConfig_default` and `TCStatConfig_default` are default configuration files for the TC_Pairs and TC-Stat tools. Users are encouraged to copy the default configuration files to another location to modify. The installed **share/met/tc_data/** directory contains static files used in MET_TC. The files `aland.dat, shland.data, wland.dat` are included to be used as input files for TC-dland, `dland_nw_hem_tenth_degree.nc` and `dland_global_tenth_degree.nc` are pre-computed distance to land files, and `wwpts_us.txt` is a sample watch/warning file to be used in TC-Pairs.

The **doc/** directory contains the documentation for both MET and MET-TC. The MET User's Guide should be referenced for further information on building the MET software package.

The **src/tools/tc_utils/** directory contains the source code for the three tools in MET-TC.

The **scripts/Rscripts**/ directory contains the R script `plot_tcmpr.R`, which provides graphic utilities for MET-TC. For more information on the graphics capabilities, see Chapter 6 of this User's Guide.

The **MET_BASE** variable is defined in the code at compilation time as the path to the **share/met** installation directory (**<prefix>/share/met**). **MET_BASE** may be used in the MET-TC configuration files when specifying paths and the appropriate path will be

substituted in. If **MET_BASE** is defined as an environment variable, its value will be used instead of the one defined at compilation time.

1.4 Code support

MET-TC support is supplied through the MET helpdesk. The MET help email address is: met_help@ucar.edu. Comments and suggestions for improvements to MET-TC are welcome and encouraged. We cannot guarantee all suggested improvements will be incorporated into the code, but all suggestions will be considered.

Chapter 2 – Data input/output

This chapter discusses the input and output file formats expected and produced by MET-TC. When discussing the input data, it is expected that users have run model output through vortex tracking software in order to obtain position and intensity information in Automated Tropical Cyclone Forecasting System (ATCF) file format.

2.1 Input data format

Best track and aids files in Automated Tropical Cyclone Forecasting System (ATCF) format (hereafter referred to as ATCF format) are necessary for model data input into the TC-Pairs tool. The ATCF format was first developed at the Naval Oceanographic and Atmospheric Research Laboratory (NRL), and is currently used for the National Hurricane Center (NHC) operations. ATCF format must be adhered to in order for the MET-TC tools to properly parse the input data.

The ATCF file format includes a section with common fields:

BASIN, CY, YYYYMMDDHH, TECHNUM/MIN, TECH, TAU, LatN/S, LonE/W, VMAX, MSLP, TY, RAD, WINDCODE, RAD1, RAD2, RAD3, RAD4, POUTER, ROUTER, RMW, GUSTS, EYE, SUBREGION, MAXSEAS, INITIALS, DIR, SPEED, STORMNAME, DEPTH, SEAS, SEASCODE, SEAS1, SEAS2, SEAS3, SEAS4

BASIN: basin
CY: annual cyclone number: 1 - 99
YYYYMMDDHH: Warning Date-Time-Group.
TECHNUM/MIN: objective technique sorting number, minutes for best track: 00 - 99
TECH: acronym for each objective technique or CARQ or WRNG, BEST for best track
TAU: forecast period: -24 through 240 hours, 0 for best-track
LatN/S: Latitude for the date time group (DTG)
LonE/W: Longitude for the DTG
VMAX: Maximum sustained wind speed in knots
MSLP: Minimum sea level pressure, 850 - 1050 mb.

TY: Highest level of tropical cyclone development
RAD: Wind intensity for the radii defined in this record: 34, 50 or 64 kt.
WINDCODE: Radius code
RAD1: If full circle, radius of specified wind intensity, or radius of first quadrant wind intensity
RAD2: If full circle this field not used, or radius of 2nd quadrant wind intensity
RAD3: If full circle this field not used, or radius of 3rd quadrant wind intensity
RAD4: If full circle this field not used, or radius of 4th quadrant wind intensity
POUTER: pressure in millibars of the last closed isobar
ROUTER: radius of the last closed isobar
RMW: radius of max winds
GUSTS: gusts
EYE: eye diameter
SUBREGION: subregion
MAXSEAS: max seas
INITIALS: Forecaster's initials
DIR: storm direction
SPEED: storm speed
STORMNAME: literal storm name, number, NONAME or INVEST, or TCcyx
DEPTH: system depth
SEAS: Wave height for radii defined in SEAS1 - SEAS4
SEASCODE - Radius code
SEAS1: first quadrant seas radius as defined by SEASCODE
SEAS2: second quadrant seas radius as defined by SEASCODE
SEAS3: third quadrant seas radius as defined by SEASCODE
SEAS4: fourth quadrant seas radius as defined by SEASCODE

Of the above common fields in the ATCF file format, MET-TC requires the input file has the first 17 comma-separated columns present. Although all 17 columns must exist, valid data in each field is not required. In order to ensure proper matching, unique data in the BASIN, CY, YYYYMMDDHH, and TAU fields should be present.

The TC-Pairs tool expects two input data sources in order to generate matched pairs and subsequent error statistics. The expected input for MET-TC is an ATCF format file from model output, or the operational aids files with the operational model output for the 'adeck' and the NHC best track analysis (BEST) for the 'bdeck'. The BEST is a subjectively smoothed representation of the storm's location and intensity over its lifetime. The track and intensity values are based on a retrospective assessment of all available observations of the storm.

The BEST is in ATCF file format and contains all the above listed common fields. Given the reference dataset is expected in ATCF file format, any second ATCF format file from model output or operational model output from the NHC aids files can be supplied as well. The expected use of the TC-Pairs tool is to generate matched pairs between model output and the BEST. Note that some of the columns in the TC-Pairs output are populated based on the

BEST information (e.g. storm category), therefore use of a different baseline many reduce the available filtering options.

All operational model aids and the BEST can be obtained from the NHC ftp server: ftp://ftp.nhc.noaa.gov/atcf/archive/

For more detailed information on the ATCF format description and specifications see: http://www.nrlmry.navy.mil/atcf_web/docs/database/new/abdeck.txt

In order to adhere to ATCF file format, model data must be run through a vortex tracking algorithm prior to becoming input for MET-TC. Many vortex tracking algorithms have been developed in order to obtain basic position, maximum wind, and minimum sea level pressure information from a model forecasts. One vortex tracking algorithm that is supported and freely available is the GFDL vortex tracker. Refer to http://www.dtcenter.org/HurrWRF/users/downloads/index.php for more information on the GFDL vortex tracker package.

2.2 Output data format

The MET package produces output in four basic file formats: STAT files, ASCII files, NetCDF files, and Postscript plots. The MET-TC tool produces output in TCSTAT, which stands for Tropical Cyclone – STAT. This output format consists of tabular ASCII data that can be easily read by many analysis tools and software packages, making the output from MET-TC very versatile. Like STAT, TCSTAT is a specialized ASCII format containing one record on each line. Currently, the only line type available in MET-TC is TCMPR (Tropical Cyclone Matched Pairs). As more line types are included in future releases, all line types will be included in a single TCSTAT file. MET-TC also outputs a NetCDF format file in the TC-dland tool, as input to the TC-Pairs tool.

Chapter 3 – TC-dland tool

3.1 Introduction

Many filtering criteria within the MET-TC tools depend on the distinction between when a storm is over land or water. The TC-dland tool was developed to aid in quickly parsing data for filter jobs that only verify over water, threshold verification based on distance to land, and exclusion of forecasts outside a specified time window of landfall.

The TC-dland tool underwent a major overhaul for MET v5.0, switching from running legacy FORTRAN code over one quarter of the earth to computing more sophisticated great circle arc distances over the entire globe. The resulting distances are more accurate but take considerably longer to compute. Existing users of MET-TC may notice TC-dland running more slowly and slight differences in the computed distances compared with the previous version.

While the TC-dland tool is available to be run, most users will find the pre-computed distance to land files distributed with the release sufficient. Therefore, the typical user will not actually need to run this tool.

3.2 Input/output format

The input for the TC-dland tool is a file containing the longitude (degrees W negative) and latitude (degrees N positive) of all the coastlines and islands considered to be a significant landmass. The default input is to use all three land data files (`aland.dat`, `shland.dat`, `wland.dat`) found in the installed **share/met/tc_data/** directory. The use of all three files produces a global land data file. The aland.dat file contains the longitude and latitude distinctions used by NHC for the Atlantic and eastern North Pacific basins, the shland.dat contains longitude and latitude distinctions for the Southern Hemisphere (south Pacific and South Indian Ocean), and the wland.dat contains the remainder of the Northern Hemisphere (western North Pacific and North Indian Ocean). Users may supply their own input file in order to refine the definition of coastlines and a significant landmass.

The output file from TC-dland is a NetCDF format file containing a gridded field representing the distance to the nearest coastline or island, as specified in the input file. This file is used in the TC-Pairs tool to compute the distance from land for each track point in the adeck and bdeck. As noted in chapter 1.3, pre-computed distance to land (NetCDF output from TC-dland) files are available in the release. In the installed **share/met/tc_data** directory:
`dland_nw_hem_tenth_degree.nc`: TC-dland output from aland.dat using a $1/10^{th}$ degree grid
`dland_global_tenth_degree.nc`: TC-dland output from all three land data files (global coverage) using a $1/10^{th}$ degree grid.

3.3 Practical information

This section briefly describes how to run `tc_dland`. The default grid is set to 1/10th degree NW hemisphere grid.

3.3.1 `tc_dland` usage

Usage: `tc_dland`

```
out_file
[-grid spec]
[-noll]
[-land file]
[-log file]
[-v level]
```

Required arguments for `tc_dland`:

Out_file: The –`out_file` argument indicates the NetCDF output file containing the computed distances to land.

Optional arguments for `tc_dland`:

Grid spec: The –`grid` spec argument overrides the default grid (1/10th NH grid). Spec = `lat_ll lon_ll delta_lat delta_lon n_lat n_lon`

Noll: The –`noll` argument skips writing the lon/lat variables in the output NetCDF file to reduce the file size.

Land file: The –`land` file argument overwrites the default land data files (aland.dat, shland.dat, and wland.dat).

Log file: The –`log` file argument outputs log messages to the specified file.

V level: The –`v` level argument overrides the default level of verbosity (2).

Chapter 4 – TC-Pairs tool

4.1 Introduction

The TC-Pairs tool provides position and intensity verification for tropical cyclone forecasts in ATCF file format. The TC-Pairs tool matches an ATCF format tropical cyclone (TC) forecast with a second ATCF format reference TC dataset (most commonly the Best Track analysis). The matched pairs produce position errors, as well as wind, sea level pressure, and distance to land values for each TC dataset. The pair generation can be subset based on user-defined filtering criteria. Practical aspects of the TC-Pairs tool are described in section 4.2.

4.2 Practical information

This section describes how to configure and run the TC-Pairs tool. The TC-Pairs tool is used to match a tropical cyclone model forecast to a corresponding reference dataset. Both tropical cyclone forecast/reference data must be in ATCF format. Output from the TC-dland tool (NetCDF gridded distance file) is also a required input for the TC-Pairs tool. It is recommended to run `tc_pairs` on a storm-by-storm basis, rather than over multiple storms or seasons to avoid memory issues.

4.2.1 `tc_pairs` usage

The usage statement for `tc_pairs` is shown below:

```
Usage:    tc_pairs
          -adeck source
          -bdeck source
          -config file
          [-out base]
          [-log file]
          [-v level]
```

`tc_pairs` has three required arguments and can take up to three optional arguments.

Required arguments for `tc_pairs`:

Adeck source: The `-adeck` argument indicates the ATCF format data source containing tropical cyclone model forecast (output from tracker) data to be verified. It specifies the name of an ATCF format file or top-level directory containing ATCF format files ending in ".dat" to be processed.

Bdeck source: The `-bdeck` argument indicates the ATCF format data source containing the tropical cyclone reference dataset to be used for verifying the adeck source. It specifies the name of an ATCF format file or top-level directory containing ATCF format files ending in ".dat" to be processed. This source is expected to be the NHC Best Track Analysis, but could also be any ATCF format reference.

Config file: The `-config` argument indicates the name of the configuration file to be used. The contents of the configuration file are discussed below.

Optional arguments for `tc_pairs`:

Out base: The `–out` argument indicates the path of the output file base. This argument overrides the default output file base (./out_tcmpr)

Log file: The `–log` argument indicates the name of the log file associated with `tc_pairs` output.

V level: The `-v` option indicates the desired level of verbosity. The value of "level" will override the default setting of 2. Setting the verbosity to 0 will make the tool run with no log messages, while increasing the verbosity above 2 will increase the amount of log output.

An example of the `tc_pairs` calling sequence is shown below:

```
tc_pairs -adeck aal092010.dat -bdeck bal092010.dat -config TCPairsConfig
```

In this example, the TC-Pairs tool matches the model track (aal092010.dat) and the best track analysis (bal092010.dat) for the 9th Atlantic Basin storm in 2010. The track matching and subsequent error information is generated with configuration options specified in the `TCPairsConfig` file.

The TC-Pairs tool implements the following logic:

- Parse the adeck and bdeck data files and store them as track objects.
- Apply configuration file settings to filter the adeck and bdeck track data down to a subset of interest.
- Apply configuration file settings to derive additional adeck track data, such as interpolated tracks, consensus tracks, time-lagged tracks, and statistical track and intensity models.
- For each adeck track that was parsed or derived, search for a matching bdeck track with the same basin and cyclone number and overlapping valid times. If not matching against the BEST track, also ensure that the model initialization times match.

- For each adeck/bdeck track pair, match up their track points in time, lookup distances to land, compute track location errors, and write an output TCMPR line for each track point.

4.2.2 TC_Pairs configuration file

The default configuration file for the TC-Pairs tool named '`TCPairsConfig_default`' can be found in the installed **share/met/config/** directory. It is encouraged for users to copy these default files before modifying their contents. The contents of the configuration file are described in the subsections below.

---

model = [ ];

e.g.: `model = ["DSHP", "LGEM", "HWRF"];`

The **`model`** variable contains a list of comma-separated models to be used. The models are identified with an ATCF ID (normally four unique characters). This model identifier should match the model column in the ATCF format input file. An empty list indicates that all models in the input file(s) will be processed.

---

storm_id = [ ];

e.g.: `storm_id=["AL092010"];`

The **`storm_id`** variable contains the specific storm identifier (or comma separated list of storm IDs) to be processed, represented by BBNNYYYY, where B is the two-letter basin, N is the two digit cyclone number, and Y is the four digit year. An empty list indicates all storms in the input file(s) will be processed.

---

basin = [ ];

e.g.: `basin = ["AL", "EP"];`

The **`basin`** variable contains a comma-separated list of basins to be processed. The basin name is expected to be the two-letter identifier for the storm basin (eg: AL = Atlantic basin, EP = eastern North Pacific basin). Valid basins include: WP, IO, CP, EP, AL, SH. An empty list indicates all basins in the input file(s) will be processed.

---

cyclone = [ ];

e.g.: `cyclone = [“01”,02”];`

The **`cyclone`** variable contains the cyclone number (or a comma separated list of cyclone numbers) to be processed. Valid cyclone numbers are 01-99. An empty list indicates all storm names in the input file(s) will be processed.

---

storm_name = [ ];

e.g.: `storm_name = [“IRENE”,”EARL”];`

The **`storm_name`** variable contains the specific named storm (or a comma separated list of storm names) to be processed. If the input file(s) span multiple years, which contain repeated storm names, use of the storm id field is recommend. An empty list indicates all storm names in the input file(s) will be processed.

---

init_beg = ” “;
init_end = ” “;

e.g.: `init_beg=”20100701“;`
`init_end=”20101031“;`

The **`init_beg`** and **`init_end`** fields specify a model initialization time window (YYYYMMDD_[HH[MMSS]]). Tracks whose initial time meets the specified criteria will be used. An empty string indicates all initialization times in the input file(s) will be processed.

---

init_inc = [ ];
init_exc = [ ];

e.g.: `init_inc = [“20101031_06”];`
`init_exc = [“20101031_12”];`

The **`init_inc`** and **`init_exc`** are comma separated lists of specific initialization times (YYYYMMDD_[HH[MMSS]]) to include (inc) or exclude (exc). Tracks whose initial time meets the specified criteria will be used. A blank field will result in all initialization times in the input file(s) being processed.

---

valid_beg = ” “;
valid_end = ” “;

e.g.: `valid_beg = ”20100701_00“`

```
valid_end = "20101031_18"
```

The **valid_beg** and **valid_end** fields specify a valid time window (YYYYMMDD_[HH[MMSS]]). Tracks for which all valid times fall within the time window will be used. An empty string indicates all valid times in the input file(s) will be processed.

---

init_hour = [ ];

e.g.: `init_hour = ["00", "06", "12", "18"];`

The **init_hour** field specifies a comma-separated list of model initialization hours to be used (HH[MMSS]). An empty list indicates all initialization hours in the input file(s) will be processed.

---

init_mask = " ";

e.g.: `init_mask = "MET_BASE/poly/EP.poly";`

The **init_mask** field specifies a lat/lon polyline defining a masking region to be applied. Tracks whose initial location falls within the initial mask region will be used. An empty string indicates no masking region will be applied.

---

valid_mask = " ";

e.g.: `valid_mask = "MET_BASE/poly/EP.poly";`

The **valid_mask** field specifies a lat/lon polyline defining a masking region to be applied. Tracks for which all locations fall within the valid mask will be used. An empty string indicates no masking region will be applied.

---

check_dup = TRUE or FALSE;

e.g.: `check_dup = FALSE;`

The **check_dup** flag expects either TRUE and FALSE, indicating whether the code should check for duplicate ATCF lines when building tracks. Setting **check_dup** to TRUE will check for duplicated lines, and produce output information regarding the duplicate. The duplicated ATCF line will not be processed in the `tc_pairs` output. Setting **check_dup** to FALSE, will still exclude tracks that decrease with time, and will overwrite repeated

lines, but specific duplicate log information will not be output. Setting **check_dup** to FALSE will make parsing the track quicker.

---

interp12 = NONE, FILL, or REPLACE;

e.g.: `interp12 = NONE;`

The **interp12** flag expects the entry NONE, FILL, or REPLACE, indicating whether special processing should be performed for interpolated forecasts. The NONE option indicates no changes are made to the interpolated forecasts. The FILL and REPLACE (default) options determine when the 12-hour interpolated forecast (normally indicated with a "2" at the end of the ATCF ID) will be renamed with the 6-hour interpolated ATCF ID (normally indicated with the letter "I" at the end of the ATCF ID). The FILL option renames the 12-hour interpolated forecasts with the 6-hour interpolated forecast ATCF ID only when the 6-hour interpolated forecasts is missing (in the case of a 6-hour interpolated forecast which only occurs every 12-hours (e.g. EMXI, EGRI), the 6-hour interpolated forecasts will be "filled in" with the 12-hour interpolated forecasts in order to provide a record every 6-hours). The REPLACE option renames all 12-hour interpolated forecasts with the 6-hour interpolated forecasts ATCF ID regardless of whether the 6-hour interpolated forecast exists. The original 12-hour ATCF ID will also be retained in the output file (all modified ATCF entries will appear at the end of the TC-Pairs output file). This functionality expects both the 12-hour and 6-hour early (interpolated) ATCF IDs are listed in the model field.

---

consensus = [ ];

e.g.:
```
consensus = [
     {
       name     = "CON1";
       members  = ["MOD1","MOD2","MOD3"];
       required = [ true, false, false ];
       min_req  = 2;
     }
];
```

The **consensus** field allows the user to generate a user-defined consensus forecasts from any number of models. All models used in the consensus forecast need to be included in the **model** field (1st entry in `TCPairsConfig_default`). The **name** field is the desired consensus model name. The **members** field is a comma-separated list of model IDs that make up the members of the consensus. The **required** field is a comma-separated list of true/false values associated with each consensus member. If a member is designated as true, the member is required to be present in order for the consensus to be generated. If a member is false, the consensus will be generated regardless of whether the member is present. The length of the required array must be the same length as the members array.

The **min_req** field is the number of members required in order for the consensus to be computed. The required and min_req field options are applied at each forecast lead time. If any member of the consensus has a non-valid position or intensity value, the consensus for that valid time will not be generated.

---

lag_time = [ ];

e.g.: `lag_time = ["06","12"];`

The **lag_time** field is a comma-separated list of forecast lag times to be used (HH[MMSS]). For each adeck track identified, a lagged track will be derived for each entry. In the `tc_pairs` output, the original adeck record will be retained, with the lagged entry listed as the adeck name with '_LAG_HH' appended.

---

best_baseline = [ ];

e.g.: `best_baseline = ["BCLP", "BCD5", "BCLA"];`

The **best_baseline** field specifies a comma-separated list of CLIPER/SHIFOR baseline forecasts to be derived from the best tracks.

The following are valid baselines for the **best_baseline** field:

BTCLIP: Neumann original 3-day CLIPER in best track mode. Used for the Atlantic basin only. Specify model as `BCLP`.

BTCLIP5: 5-day CLIPER (*Aberson, 1998*)/SHIFOR (*DeMaria and Knaff, 2001*) in best track mode for either Atlantic or eastern North Pacific basins. Specify model as `BCS5`.

BTCLIPA: Sim Aberson's recreation of Neumann original 3-day CLIPER in best-track mode. Used for Atlantic basin only. Specify model as `BCLA`.

oper_baseline = [ ];

e.g.: `oper_baseline = ["OCLP","OCS5","OCD5"];`

The **oper_baseline** field specifies a comma-separated list of CLIPER/SHIFOR baseline forecasts to be derived from the operational (CARQ) tracks.

The following are valid baselines for the **oper_baseline** field:

OCLIP: Merrill modified (operational) 3-day CLIPER run in operational mode. Used for Atlantic basin only. Specify model as OCLP.

OCLIP5: 5-day CLIPER (*Aberson, 1998*)/ SHIFOR (*DeMaria and Knaff, 2001*) in operational mode, rerun using CARQ data. Specify model as OCS5.

OCLIPD5: 5-day CLIPER (*Aberson, 1998*)/ DECAY-SHIFOR (*DeMaria and Knaff, 2001*). Specify model as OCD5.

---

match_points = TRUE or FALSE;

e.g.: `match_points = TRUE;`

The **match_points** field specifies whether only those track points common to both the adeck and bdeck tracks should be written out. If **match_points** is selected as FALSE, the union of the adeck and bdeck tracks will be written out, with "NA" listed for unmatched data.

---

dland_file = " ";

e.g.: `dland_file="MET_BASE/tc_data/dland_global_tenth_degree.nc";`

The **dland_file** string specifies the path of the NetCDF format file (default file: dland_global_tenth_degree.nc) to be used for the distance to land check in the tc_pairs code. This file is generated using tc_dland (default file provided in installed **share/met/tc_data** directory).

---

watch_warn = [ ];

```
e.g.: watch_warn = {
      file_name   = "MET_BASE/tc_data/wwpts_us.txt";
      time_offset = -14400;
     };
```

The **watch_warn** field specifies the file name and time applied offset to the **watch_warn** flag. The **file_name** string specifies the path of the watch/warning file to be used to determine when a watch or warning is in affect during the forecast initialization and verification times. The default file is named wwpts_us.txt, which is found in the installed **share/met/tc_data/** directory within the MET build. The **time_offset** string is the time window (in seconds) assigned to the watch/warning. Due to the non-uniform time watches and warnings are issued, a time window is assigned for which

watch/warnings are included in the verification for each valid time. The default watch/warn file is static, and therefore may not include warned storms beyond the current MET code release date; therefore users may wish to contact met_help@ucar.edu to obtain the most recent watch/warning file if the static file does not contain storms of interest.

---

version = " ";

e.g.: `version = "V5.0";`

The version string indicates the version of TC-Pairs configuration file used. This version corresponds to the MET release. Future versions of MET may include changes to TC-Pairs and the TC-Pairs configuration file. This value should generally not be modified.

4.2.3 TC_Pairs output

TC-Pairs produces output in TCSTAT format. The default output file name can be overwritten using the `–out` file argument in the usage statement. The TCSTAT file output from TC-Pairs may be used as input into the TC-Stat tool. The header column in the TC-Pairs output is described in table 3.1.

Table 3.1: Header information for TC-Pairs TCST output

HEADER		
Column Number	**Header column Name**	**Description**
1	VERSION	Version number
2	AMODEL	User provided text string designating model name
3	BMODEL	User provided text string designating model name
4	STORM_ID	BBCCYYY designation of storm
5	BASIN	Basin
6	CYCLONE	Cyclone number
7	STORM_NAME	Name of Storm
8	INIT	Initialization time of forecast
9	LEAD	Forecast lead time in HH format
10	VALID	Forecast valid time in YYYYMMDD_HH
11	INIT_MASK	Initialization time masking grid applied
12	VALID_MASK	Valid time masking grid applied
13	LINE_TYPE	Output line type (TCMPR currently only line type)
14	TOTAL	Total number of matched pairs
15	INDEX	Index associated with init time
16	LEVEL	Level of storm classification
17	WATCH_WARN	HU or TS watch or warning in effect
18	INITIALS	Forecaster initials

19	ALAT	Latitude position of adeck model
20	ALON	Longitude position of adeck model
21	BLAT	Latitude position of bdeck model
22	BLON	Longitude position of bdeck model
23	TK_ERR	Track error of adeck relative to bdeck
24	X_ERR	X component position error
25	Y_ERR	Y component position error
26	ALTK_ERR	Along track error
27	CRTK_ERR	Cross track error
28	ADLAND	adeck distance to land
29	BDLAND	bdeck distance to land
30	AMSLP	adeck mean sea level pressure
31	BMSLP	bdeck mean sea level pressure
32	AMAX_WIND	adeck maximum wind speed
33	BMAX_WIND	bdeck maximum wind speed
34, 35	A/BAL_WIND_34	a/bdeck 34-knot radius winds in full circle
36, 37	A/BNE_WIND_34	a/bdeck 34-knot radius winds in NE quadrant
38, 39	A/BSE_WIND_34	a/bdeck 34-knot radius winds in SE quadrant
40, 41	A/BSW_WIND_34	a/bdeck 34-knot radius winds in SW quadrant
42, 43	A/BNW_WIND_34	a/bdeck 34-knot radius winds in NW quadrant
44, 45	A/BAL_WIND_50	a/bdeck 50-knot radius winds in full circle
46, 47	A/BNE_WIND_50	a/bdeck 50-knot radius winds in NE quadrant
48, 49	A/BSE_WIND_50	a/bdeck 50-knot radius winds in SE quadrant
50, 51	A/BSW_WIND_50	a/bdeck 50-knot radius winds in SW quadrant
52, 53	A/BNW_WIND_50	a/bdeck 50-knot radius winds in NW quadrant
54, 55	A/BAL_WIND_64	a/bdeck 64-knot radius winds in full circle
56, 57	A/BNE_WIND_64	a/bdeck 64-knot radius winds in NE quadrant
58, 59	A/BSE_WIND_64	a/bdeck 64-knot radius winds in SE quadrant
60, 61	A/BSW_WIND_64	a/bdeck 64-knot radius winds in SW quadrant
62, 63	A/BNW_WIND_64	a/bdeck 64-knot radius winds in NW quadrant

Chapter 5 - TC-Stat

5.1 Introduction

The TC-Stat tool ties together results from the TC-Pairs tool by providing summary statistics and filtering jobs on TCST output files. The TC-Stat tool requires TCST output from the TC-Pairs tool. See section 4.2.3 of this users guide for information on the TCST output format of the TC-Pairs tool. A filtering job using TC-Stat provides the ability to filter the input data by various conditions and thresholds described in section 5.3.2. A summary job produces summary statistics including frequency of superior performance, time-series independence calculations, and confidence intervals on the mean. These statistical aspects are described in section 5.2, and practical use information for the TC-Stat tool is described in section 5.3.

5.2 Statistical aspects

5.2.1 Filter TCST lines

The TC-Stat tool can be used to simply filter specific lines of the TCST file based on user-defined filtering criteria. All of the TCST lines that are retained from one or more files are written out to a single output file. The output file is also in TCST format.

Filtering options are outlined below in section 5.3.2 (configuration file). Additional filtering options that can be executed on the command line will give the minimum, maximum, or exact value for a particular column:

`-column_min` arguments: column_name value
`-column_max` arguments: column_name value
`-column_eq` arguments: column_name value
`-column_str` arguments: column_name string

If multiple options are listed, the job will be performed on their intersection.

5.2.2 Summary statistics for columns

The TC-Stat tool can be used to produce summary information for a single column of data. After the user specifies the specific column of interest, and any other relevant search criteria, summary information is produced from values in that column of data. The summary statistics produced are: mean, standard deviation, minimum, 10^{th}, 25^{th}, 50^{th}, 75^{th}, and 90^{th} percentiles, and maximum.

Confidence intervals are computed for the mean of the column of data. Confidence intervals are computed using the assumption of normality for the mean. For further

information on computing confidence intervals, refer to Appendix D of the MET user's guide.

When operating on columns, a specific column name can be listed (e.g. `TK_ERR`), as well as the differences of two columns (e.g. `AMAX_WIND-BMAX_WIND`), and the absolute difference of the column(s) (e.g. `abs(AMAX_WIND-BMAX_WIND)`). Additionally, several shortcuts can be applied to choose multiple columns with a single entry. Shortcut options for the –column entry are as follows:

`TRACK`: track error (TK_ERR), along-track error (ALTK_ERR), and cross-track error (CRTK_ERR)
`WIND`: all wind radii errors (34-, 50-, and 64-kt) for each quadrant
`TI`: track error (TK_ERR) and absolute intensity error (abs(AMAX_WIND-BMAX_WIND))
`AC`: along- and cross-track errors (ALTK_ERR, CRTK_ERR)
`XY`: X- and Y-component track errors (X_ERR, Y_ERR)

The TC-Stat tool can also be used to generate frequency of superior performance and the time to independence calculations when using the TC-Stat summary job.

Frequency of Superior Performance

The frequency of superior performance (FSP) looks at multiple model forecasts (adecks), and ranks each model relative to other model performance for the column specified in the summary job. The summary job output lists the total number of cases included in the FSP, the number of cases where the model of interest is the best (e.g.: lowest track error), the number of ties between the models, and the FSP (percent). Ties are not included in the FSP percentage; therefore the percentage may not equal 100%.

Time-Series Independence

The time-series independence evaluates effective forecast separation time using the Siegel method, by comparing the number of runs above and below the mean error to an expected value. This calculation expects the columns in the summary job to be a time series. The output includes the forecast hour interval and the number of hours to independence.

5.3 Practical information

The following sections describe the usage statement, required arguments, and optional arguments for `tc_stat`.

5.3.1 `tc_stat` usage

The usage statement for `tc_stat` is shown below:

Usage: `-lookin source`
`[-out file]`
`[-log file]`
`[-v level]`
`[-config file] | [JOB COMMAND LINE]`

TC-Stat has one required argument and accepts optional ones. The usage statement for the TC-Stat tool includes the "job" term, which refers to the set of tasks to be performed after applying user-specified filtering options. The filtering options are used to pare down the TC-Pairs output to only those lines that are desired for the analysis. The job and its filters together comprise a "job command line". The "job command line" may be specified either on the command line to run a single analysis job or within the configuration file to run multiple analysis jobs at the same time. If jobs are specified in both the configuration file and the command line, only the jobs indicated in the configuration file will be run. The various jobs are described in Table 5-1 and the filtering options are described in section 5.3.2.

Required arguments for `tc_stat`:

Lookin source: The `-lookin` argument indicates the location of the input TCST files generated from `tc_pairs`. This argument can be used one or more times to specify the name of a TCST file or top-level directory containing TCST files to be processed. Multiple tcst files may be specified by using a wild card (*).

Optional arguments for `tc_stat`:

Out file: The `-`out argument indicates the desired name of the TCST format output file.

Log file: The `-log` argument indicates the name of the log file associated with `tc_stat` output.

V level: The `-v` option indicates the desired level of verbosity. The value of "level" will override the default setting of 2. Setting the verbosity to 0 will make the tool run with no log messages, while increasing the verbosity above 2 will increase the amount of log output.

Config file: The `-config` argument indicates the name of the configuration file to be used. The contents of the configuration file are discussed below.

An example of the `tc_stat` calling sequence is shown below:

```
tc_stat -lookin /home/tc_pairs/*al092010.tcst -config
TCStatConfig
```

In this example, the TC-Stat tool uses any TCST file (output from `tc_pairs`) in the listed directory for the 9th Atlantic Basin storm in 2010. Filtering options and aggregated statistics are generated following configuration options specified in the `TCStatConfig` file.

5.3.2 `tc_stat` configuration file

The default configuration file for the TC-Stat tool named `TCStatConfig_default` can be found in the installed **share/met/config** directory. Like the other configuration files described in this document, it is recommended that users make a copy of these files prior to modifying their contents.

Most of the user-specified options listed in the TC-Stat configuration file are used to filter the TCST output from `tc_pairs` down to the desired subset over which the statistics will be computed. Only output from the TC-Pairs tool that meet the filtering criteria listed in the TC-Stat configuration file will be retained. A detailed description of the TC-Stat configuration file is listed below.

Note that the options specified in the first section of the configuration file, prior to the job list, will be applied to every job specified in the joblist. However, if an individual job specifies an option listed above, it will be applied to that job. For example, if `model = ["GFSI","LGEM","DSHP"]` is set at the top, but the job in the joblist sets the `–model` option to `"LGEM"`, that job will only run using the LGEM model data.

```
amodel = [ ];
bmodel = [ ];
```

The **amodel** and **bmodel** fields stratify by the amodel and bmodel columns based on a comma-separated list of model names used for all analysis performed. The names must be in double quotation marks (e.g.: "HWFI"). The **amodel** list specifies the model to be verified against the listed bmodel. The **bmodel** specifies the reference dataset, generally the best track analysis. Using the "`-amodel`" and "`-bmodel`" options within the job command lines may further refine these selections.

```
storm_id = [ ];
```

The **storm_id** field specifies a comma-separated list of storm id's to be used: the expected format is 2-letter basin, 2-digit cyclone number, 4-digit year. An empty list indicates all storms in TCST input will be used. This may also be set up using basin, cyclone, and timing information below.

---

basin = [ ];

The **basin** field stratifies by the basin column, based on a comma-separated list of basins to be used. The expected format is the 2-letter basin identifier. An empty list indicates all basins in the TCST input will be used. Using the "-basin" option within the job command lines may further refine these selections.

---

cyclone = [ ];

The **cyclone** field stratifies by the cyclone column, based on a comma-separated list of 2-digit (01-99) cyclone numbers. An empty list indicates all cyclones in the TCST input will be used. Using the "-column" option within the job command lines may further refine these selections.

---

storm_name = [ ];

The **storm_name** field stratifies by the storm_name column, based on a comma-separated list of storm names. An empty list indicates all storms in the TCST input will be used. Using the "-storm_name" option within the job command lines may further refine these selections.

---

init_beg = " ";
init_eng = " ";

The **init_beg** and **init_end** fields stratify by initialization time. Beginning and ending time windows are defined by YYYYMMDD[_HH[MMSS]]. Using the "-init_beg", "-init_end" options within the job command lines may further refine these selections.

---

init_inc = [ ];
init_exc = [ ];

The **init_inc** and **init_exc** fields stratify by initialization times, based on a comma-separated list of specific initialization times to include (inc) or exclude (exc). Time strings are defined by YYYYMMDD[_HH[MMSS]]. Using the "-init_inc", "-init_exc" options within the job command lines may further refine these selections.

---

valid_beg = " ";
valid_end = " ";

The **valid_beg** and **valid_end** fields stratify by valid time. Beginning and ending time windows are defined by YYYYMMDD[_HH[MMSS]]. Using the "-valid_beg", "-valid_end" options within the job command lines may further refine these selections.

```
valid_inc = [ ];
valid_exc = [ ];
```

The **valid_inc** and **valid_exc** fields stratify by valid times, based on a comma-separated list of specific valid times to include (inc) or exclude (exc). Time strings are defined by YYYYMMDD[_HH[MMSS]]. Using the "-valid_inc", "-valid_exc" options within the job command lines may further refine these selections.

```
init_hour   = [ ];
valid_hour  = [ ];
lead        = [ ];
```

The **init_hour**, **valid_hour**, and **lead** fields stratify by the initialization time, valid time, and lead time, respectively. This field specifies a comma-separated list of initialization times, valid times, and lead times in HH[MMSS] format. Using the "-init_hour", "-valid_hour", and "-lead" options within the job command lines may further refine these selections.

```
init_mask  = [ ];
valid_mask = [ ];
```

The **init_mask** and **valid_mask** fields stratify by the init_mask and valid_mask columns in the TCST output. Using the "-init_mask" and "-valid_mask" options within the job command lines may further refine these selections.

```
line_type = [ ];
```

The **line_type** field stratifies by the line_type column. Currently TCMPR is the only line_type option used in MET-TC.

```
track_watch_warn = [ ];
```

The **track_watch_warn** flag stratifies over the watch_warn column in the TCST files. If any of the watch/warning statuses are present in a forecast track, the entire track is verified. The value "ALL" matches HUWARN, HUWATCH, TSWARN, TSWATCH. Using the

"`-track_watch_warn`" option within the job command lines may further refine these selections.

Other uses of the WATCH_WARN column include filtering when:

1. A forecast is issued when a watch/warn is in effect
2. A forecast is verifying when a watch/warn is in effect
3. A forecast is issued when a watch/warn is NOT in effect
4. A forecast is verified when a watch/warn is NOT in effect

The following filtering options can be achieved by the following:

```
1. init_str_name   = ["WATCH_WARN"];
   init_str_val    = ["ALL"];

2. column_str_name = ["WATCH_WARN"];
   column_str_val  = ["ALL"];

3. init_str_name   = ["WATCH_WARN"];
   init_str_val    = ["NA"];

4. column_str_name = ["WATCH_WARN"];
   column_str_val  = ["NA"];
```

Further information on the **`column_str`** and **`init_str`** fields is described below. Listing a comma-separated list of watch/warning types in the **`column_str_val`** field will stratify by a single or multiple types of warnings.

```
column_thresh_name = [ ];
column_thresh_val  = [ ];
```

The **`column_thresh_name`** and **`column_thresh_val`** fields stratify by applying thresholds to numeric data columns. Specify a comma-separated list of columns names and thresholds to be applied. The length of **`column_thresh_val`** should match that of **`column_thresh_name`**. Using the "`-column_thresh name thresh`" option within the job command lines may further refine these selections.

```
column_str_name = [ ];
column_str_val  = [ ];
```

The **`column_str_name`** and **`column_str_val`** fields stratify by performing string matching on non-numeric data columns. Specify a comma-separated list of columns names and values to be checked. The length of the **`column_str_val`** should match that of the **`column_str_name`**. Using the "`-column_str name val`" option within the job command lines may further refine these selections.

```
init_thresh_name = [];
init_thresh_val  = [];
```

The **init_thresh_name** and **init_thresh_val** fields stratify by applying thresholds to numeric data columns only when lead = 0. If lead =0, but the value does not meet the threshold, discard the entire track. The length of the **init_thresh_val** should match that of the **init_thresh_name**. Using the "`-init_thresh name val`" option within the job command lines may further refine these selections.

```
init_str_name = [];
init_str_val  = [];
```

The **init_str_name** and **init_str_val** fields stratify by performing string matching on non-numeric data columns only when lead = 0. If lead =0, but the string does not match, discard the entire track. The length of the **init_str_val** should match that of the **init_str_name**. Using the "`-init_str name val`" option within the job command lines may further refine these selections.

```
water_only = ;
```

The **water_only** flag stratifies by only using points where both the `amodel` and `bmodel` tracks are over water. When `water_only = TRUE`; once land is encountered the remainder of the forecast track is not used for the verification, even if the track moves back over water.

```
rapid_inten =  {
     track  =
     time   =
     exact  =
     thresh =   }
```

e.g.:
```
rapid_inten = {
       track   = NONE;
       time    = "24";
       exact   = TRUE;
       thresh  = >=30.0;
     }
```

The **rapid_inten** field specifies those track points for which rapid intensification (RI) or rapid weakening (RW) occurred, based on user defined RI/RW thresholds. The **track** entry specifies that RI/RW is not turned on (`NONE`), is computed based on the bmodel only

(BDECK), is computed based on the amodel only (ADECK), or computed when both the amodel and bmodel (the union of the two) indicate RI/RW (BOTH). If **track** is set to ADECK, BDECK, or BOTH, only tracks exhibiting rapid intensification will be retained. Rapid intensification is officially defined as when the change in the maximum wind speed over a 24-hour period is greater than or equal to 30 kts. This is the default setting, however flexibility in this definition is provided through the use of the **time, exact and thresh** options. The **time** field specifies the time window (HH[MMSS] format) for which the RI/RW occurred. The **exact** field specifies whether to only count RI/RW when the intensity change is over the exact time window (TRUE), which follows the official RI definition, or if the intensity threshold is met anytime during the time window (FALSE). Finally, the **thresh** field specifies the user defined intensity threshold (where ">=" indicates RI, and "<=" indicates RW). Using the "-rapid_inten" option within the job command lines may further refine these selections.

```
landfall     = FALSE;
landfall_beg = -86400;
landfall_end = 0;
```

The **landfall, landfall_beg,** and **landfall_end** fields specify whether only those track points occurring near landfall should be retained. The landfall retention window is defined as the number of seconds offset from the time of landfall. Landfall is defined as the last bmodel track point before the distance to land switches from water to land. When landfall_end is set to 0, the track is retained from the landfall_beg to the time of landfall. Using the "-landfall", "-landfall_beg", and "-landfall_end" options within the job command lines may further refine these selections.

```
match_points = TRUE;
```

The **match_points** flag specifies whether only those track points common to both the amodel and bmodel track should be retained. This option may be modified using the "-match_points" option within the job command lines.

```
event_equal = FALSE;
```

The **event_equal** flag specifies whether only those track points common to all models in the dataset should be retained. The event equalization is performed only using cases common to all listed amodel entries. A case is defined by comparing the following columns in the TCST files: BMODEL, BASIN, CYCLONE, INIT, LEAD, VALID. This option may be modified using the "-event_equal" option within the job command lines.

```
event_equal_lead = [];
```

The **event_equal_lead** flag specifies lead times that must be present for a track to be included in the event equalization logic. The event equalization is performed only using cases common to all lead times listed, enabling the verification at each lead time to be performed on a consistent dataset. This option may be modified using the "**-**event_equal_lead" option within the job command lines.

---

out_init_mask = "";

The **out_init_mask** field applies polyline masking logic to the location of the amodel track at the initialization time. If the track point falls outside the mask, discard the entire track. This option may be modified using the "**-**out_init_mask" option within the job command lines.

---

out_valid_mask = "";

The **out_valid_mask** field applies polyline masking logic to the location of the amodel track at the valid time. If the track point falls outside the mask, discard the entire track. This option may be modified using the "**-**out_valid_mask" option within the job command lines.

---

jobs[] = [ ];

The user may specify one or more analysis jobs to be performed on the TCST lines that remain after applying the filtering parameters listed above. Each entry in the joblist contains the task and additional filtering options for a single analysis to be performed. The format for an analysis job is as follows:

-job job_name REQUIRED and OPTIONAL ARGUMENTS

e.g.: -job filter -model HWFI -dump_row ./tc_filter_job.tcst
-job summary –line_type TCMPR –column TK_ERR –dump_row ./tc_summary_job.tcst

---

version = "V5.0";

The **version** string indicates the version of TC-Stat configuration file used. This version corresponds to the MET release. Future versions of MET may include changes to TC-Stat and the TC-Stat configuration file. This value should generally not be modified.

5.3.3 `tc_stat` output

The output generated from the TC-Stat tool contains statistics produced by the analysis. Additionally, it includes information about the analysis job that produced the output for each line. The output can be redirected to an output file using the "`-out`" option. The format of output from each `tc_stat` job command is listed below.

Job: Filter

This job command finds and filters TCST lines down to those meeting the criteria selected by the filter's options. The filtered TCST lines are written to a file specified by the "`-dump_row`" option. The TCST output from this job follows the TCST output description in Chapters 3 and 4.

Job: Summary

This job produces summary statistics for the column name specified by the "`-column`" option. The output of the summary job consists of three rows: "JOB_LIST", which shows the job definition parameters used for this job. "COL_NAME", followed by the summary statistics that are applied. "SUMMARY", which is followed by the total, mean (with confidence intervals), standard deviation, minimum value, percentiles (10^{th}, 25^{th}, 50^{th}, 75^{th}, 90^{th}), maximum value, sum, time to independence, and frequency of superior performance. The output columns are shown below in table 5.1. The "`-by`" option can also be used one or more times to make this job more powerful. Rather than running the specified job once, it will be run once for each unique combination of the entries found in the column(s) specified with the "`-by`" option.

Table 5.1: Columnar output of "summary" job output from the TC-Stat tool.

Column number	Description
1	SUMMARY: (job type)
2	Column (dependent parameter)
3	Case (storm + valid time)
4	Total
5	Valid
6-8	Mean including normal upper and lower confidence limits
9	Standard deviation
10	Minimum value
11-15	Percentiles (10^{th}, 25^{th}, 50^{th}, 75^{th}, 90^{th})
16	Maximum Value
17	Sum
18-19	Independence time

20-23	Frequency of superior performance

Chapter 6 - plotting and graphics support

6.2 Introduction

Graphical capabilities are included in the MET-TC release in order to assist users with a set of standard plotting capabilities. The basic R script is located in the MET build in **scripts/Rscripts/plot_tcmpr.R**.

6.2 `plot_tcmpr.R` usage

The usage statement with a short description of the options for `plot_tcmpr.R` can be obtained by typing: `Rscript plot_tcmpr.R` with no additional arguments. The only required argument is the `–lookin` source, which is the path to the TC-Pairs TCST output files. The R script reads directly from the TC-Pairs output, and calls TC-Stat directly for filter jobs specified in the `'–filter options'` argument.

In order to run this script, the **MET_BUILD_DIR** environment variable must be set to the top-level MET build directory. In addition, the `tc_stat` tool must be in your path.

6.3 Examples

The supplied R script can generate a number of different plot types including boxplots, mean, median, rank, and relative performance. Pairwise differences can be plotted for the boxplots, mean, and median. Normal confidence intervals are applied to all figures unless the `no_ci` option is set to TRUE. Below are two example plots generated from the tools.

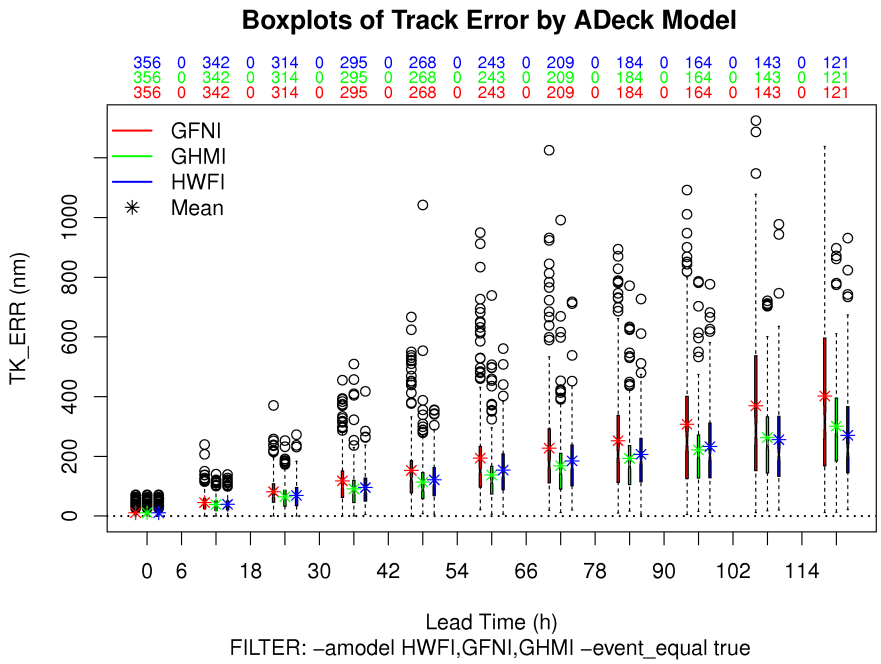

Figure 6.1: Example boxplot from `plot_tcmpr.R`. Track error distributions by lead time for three operational models GFNI, GHMI, HFWI.

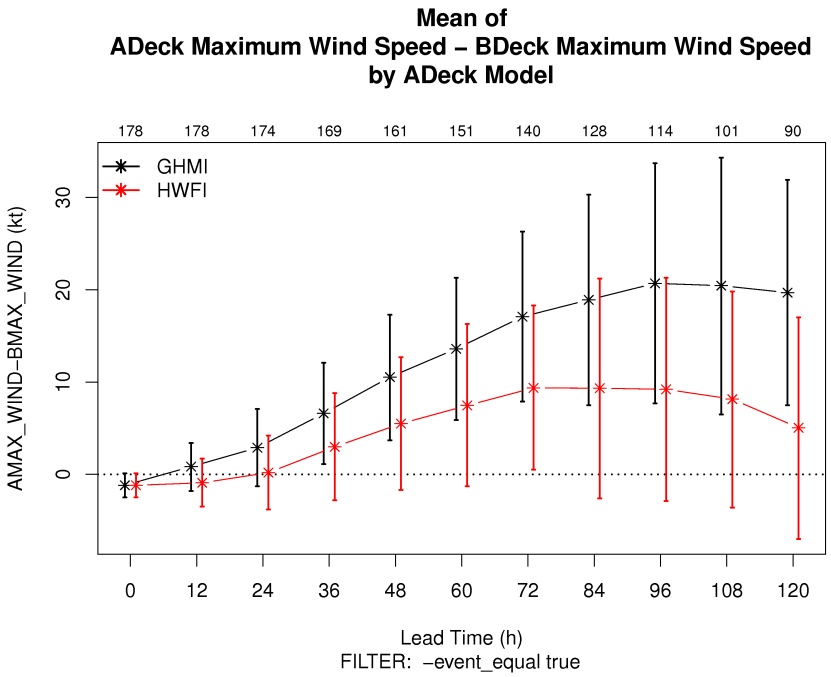

Figure 6.2: Example mean intensity error with confidence intervals at 95% from `plot_tcmpr.R`. Raw intensity error by lead time for a homogeneous comparison of two operational models GHMI, HWFI.

References

Alberson, S.D., 1998: Five-day Tropical cyclone track forecasts in the North Atlantic Basin. *Weather & Forecasting*, Vol. 13, p.1005-1015.

Knaff, J.A**.,** M. DeMaria, C.R. Sampson, and J.M. Gross, 2003: Statistical, Five-Day Tropical Cyclone Intensity Forecasts Derived from Climatology and Persistence. *Weather & Forecasting**, Vol.*** 18 **Issue** 2, p. 80-92.