An Automated Method to Identify Mesoscale Convective Complexes (MCCs) Implementing Graph Theory

Kim D. Whitehall[1, 3], Chris A. Mattmann[1, 5], Gregory S. Jenkins[3], Duane E. Waliser[1, 2], Robert Rwebangira[4], Belay Demoz[3], Jinwon Kim[2], Cameron E. Goodale[1], Andrew F. Hart[1], Paul Ramirez[1], Michael J. Joyce[1], Paul Loikith[1], Huikyo Lee[1], Shakeh Khudikyan[1], Maziyar Boustani[1], Alex Goodman[1], Paul A. Zimdars[1], Jesslyn Whittell[1]

[1]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, United States, [2]UCLA Joint Institute for Regional and Earth System Science Engineering (JIFRESSE), Los Angeles, CA, United States, [3]Program in Atmospheric Sciences, Howard University, Washington, DC, United States, [4]Department of Systems and Computer Sciences, Howard University, Washington, DC, United States, [5]University of Southern California, Los Angeles, CA. United States

Contact: **kim.whitehall@bison.howard.edu**

I. Introduction

Current methods of climate data manipulation involve averaging (e.g. monthly) but this is limiting as the mesoscale conditions that define the climate are lost.

Hence, methods that facilitate manipulation of daily and hourly climate datasets are required.

II. Mesoscale Convective Complexes (MCCs)

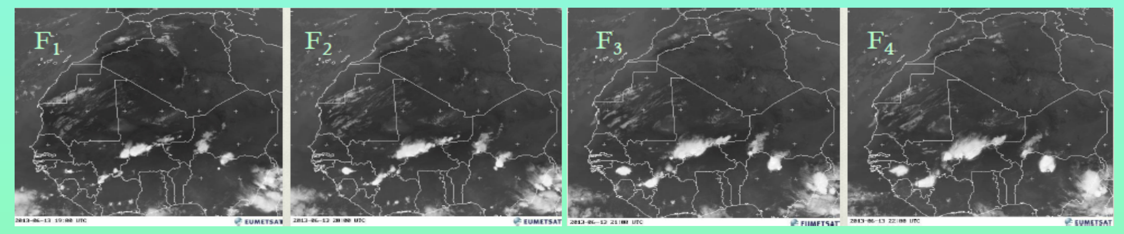
Isolated seasonal studies indicate that in West Africa, ~170 MCCs (convectively-driven weather systems with durations of ~10-12 hrs) occur within the 180 days of summer (June - November), contributing ~75% of the annual wet season rainfall.

But the existing methods of detecting MCCs are not transferable to voluminous data (e.g. hourly data), nor multiple datasets (e.g. rainfall datasets) for comprehensive studies.

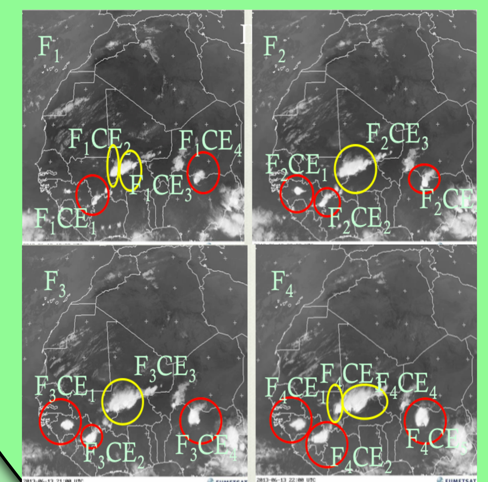
The main objective of this method is to provide a fully-automatic identification method of identifying MCCs over West Africa from a 10 year hourly dataset, that will be transferable to other datasets, locations, and weather features.

III. Graph Theory Algorithm

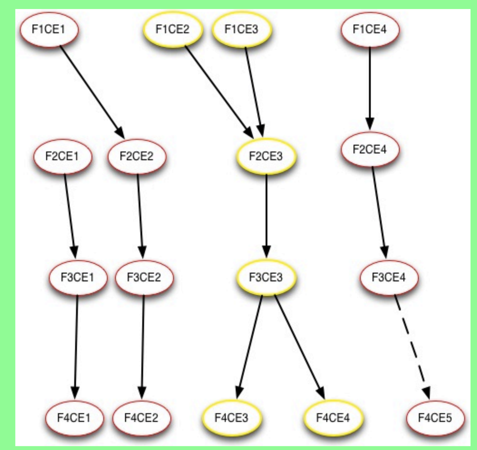
*Assume consecutive data **frames** in time.*



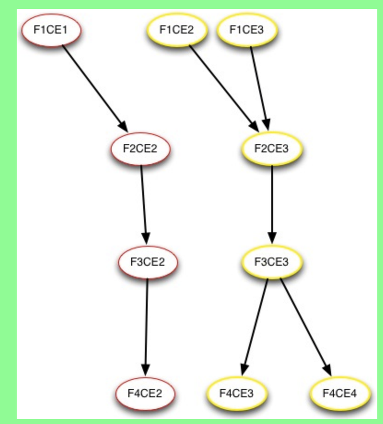
Generate graph nodes - isolate cloud areas that could develop into a MCC from each frame



Generate graph edges - correlate nodes from consecutive frames according to their % overlap



Implement graph methods to solve for MCCs



Calculate metrics to derive the MCC characteristics

IV. Implementation

Implemented in Python 2.7, including the SciPy, NumPy, PyNio, NetCDF4, MatPlotLib, and Networkx packages.

Tested with the NCEP/CPC 4km Global (60N - 60S) MERG IR satellite dataset (http://mirador.gsfc.nasa.gov/collections/MERG__001.shtml).

Current outputs include a text file of all CEs with some characteristic data, ASCII format of the location of these CEs (for easy ingestion into a GIS program), all MCCs for the analyzed time in a text file and simple metrics including average duration and initiation time.

V. Advantages

1. ***Inherently handles merging, splitting, decaying and new systems during MCC development.***
2. *Over W. Africa, with MERG IR dataset = 245, 952, 024 data points per summer, but in this approach, fewer are required.*

VI. Apache OCW

*The Apache Foundation Open Climate Workbench project (**Apache OCW**, http://climate.incubator.apache.org/) is a Python-based project that maintains the high-level functionality of extracting heterogenous data to data in homogenous Numpy arrays, regridding, calculating metrics and visualizing data, from its donor NASA JPL/UCLA project The Regional Climate Model Evaluation System (**RCMES,** http://rcmes.jpl.nasa.gov/), but allows users to build other applications that require the underlying functionality of data extraction, manipulation, calculations, and visualization.*

VII. Future work

1. *Expand metrics to determine MCC characteristics.*
2. ***Integrate the algorithm into Apache OCW***
3. *Transfer method to other datasets*
4. ***Generalize the method to searching geo-spatial data***