UROP 2023 Summer Report

Astoria Wu, Elias Han, Hanson Shen, Runchang Li, Yuxuan Shen, Ryutaro Miya

Supervisor:Prasun Ray

January 23, 2025

1 Abstract

The report will mainly aim to provide a very complete instructions about how to reproduce the article results from downloading data, installing packages needed, using of computer clusters to explaining the code for plotting the plots.

2 Introduction

The project is based on the article published on Nature, Complex Networks Reveal Global Pattern of Extreme-rainfall Teleconnections by Niklas Boers, Bedartha Goswami, Aljoscha Rheinwalt, Bodo Bookhagen, Brian Hoskins and Jurgen Kurths.

The focus is to reproduce the main results in the article, including figure 1 to figure 4, while some of the figures in the extended figure section might be plotted for auxiliary purpose. For example, extended figure 4 b is plotted to check whether we have the similar time series to that of the authors.

Apart from the article, we also use the code provided by the authors in the method section on github to supplement our understanding to the article

We will start by detailing the steps for data downloading and then introducing the steps for reproducing the article.

3 Data Download

In total we will use 5 different data set, TRMM Precipitation daily Version 7 data set, NCEP Reanalysis 1 and Reanalysis 2 precipitation data, NCEP Reanalysis 1 and Reanalysis 2 v-wind speed data.

For TRMM data set downloading, there is a set of steps that will need to be followed. First click on the download icon in the web page (the link is provided in the previous paragraph). Then set the desired timescale (usually it is 1998/01/01 to 2016/12/31), and the download method to Get Original Files. After clicking Get data, the same window as the one shown in the following figure will show up. Then follow the instructions given in Download instructions. After that the txt file downloaded it can be used to download the data. We have provided our code for downloading TRMM in our github page, download_TRMM.py.

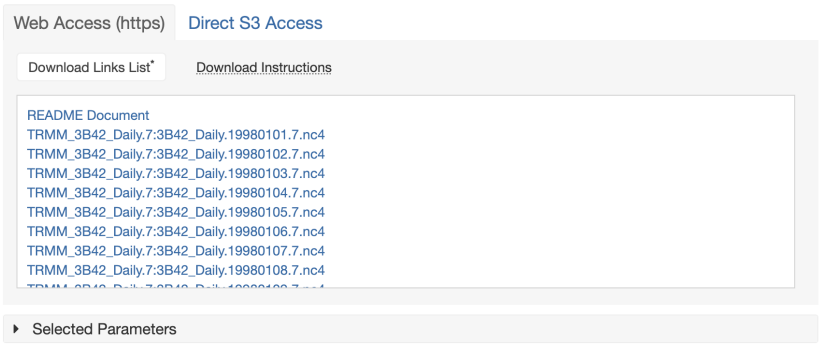

Figure 1: Download Notice TRMM

Note that for NCEP data in this project we only need mean for statistic and daily for timescale.

One error we made during the downloading stage for NCEP data set is that instead of clicking the number of files and download directly as shown in the figure above, we clicked the download icon on the first line. Then we assumed that clicking the icon on the first line will directly lead us the corresponding data, but it is not true. When doing so, it will bring us to a page contains many data sets and we will need to scroll down and find the data set needed manually. So it is recommended to click the number and download respectively from there.

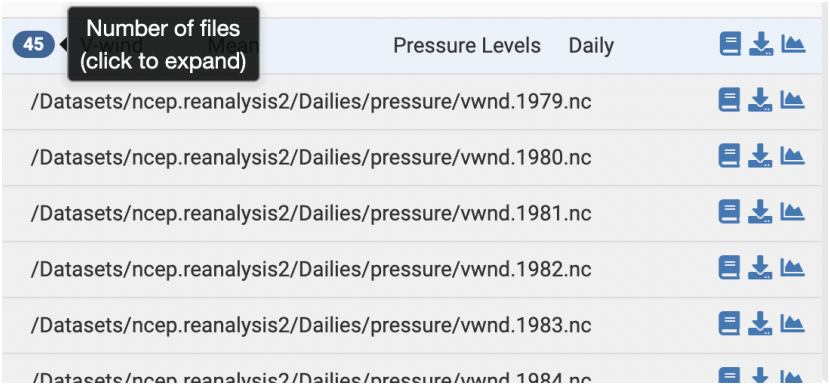

Figure 2: Download Notice NCEP

4 Usage of math computer cluster

Computation in this project can be sometimes very heavy, taking over 10 days for normal computers. While it is necessary to ensure algorithm efficiency or use technique like multiprocessing, computer with higher performance seems necessary for this project. And luckily Imperial Math department have such high performance computer cluster open to all undergrad and masters students.

Here are several links to the relevant resources. main page, scp usage, new cluster needed for this project due to storage limitation of the old cluster, ssh and other gateways for requesting outside campus.

5 Figure 1

5.1 cartopy package

The first thing to notice is that basemap package used in the original code is not available now due to python update. Therefore, cartopy package is going to be used throughout this project in order to obtain similar plots based on the world map.

One problem we have encountered during installing cartopy is that pipinstall fails to install cartopy due to dependency errors. While it is possible to install all the needed packages, which we have tried but still fail to install cartopy, one more straightforward method is to use: conda install cartopy.

5.2 Plots

Here are the two figures, figure 1a and figure 1b, produced by us.

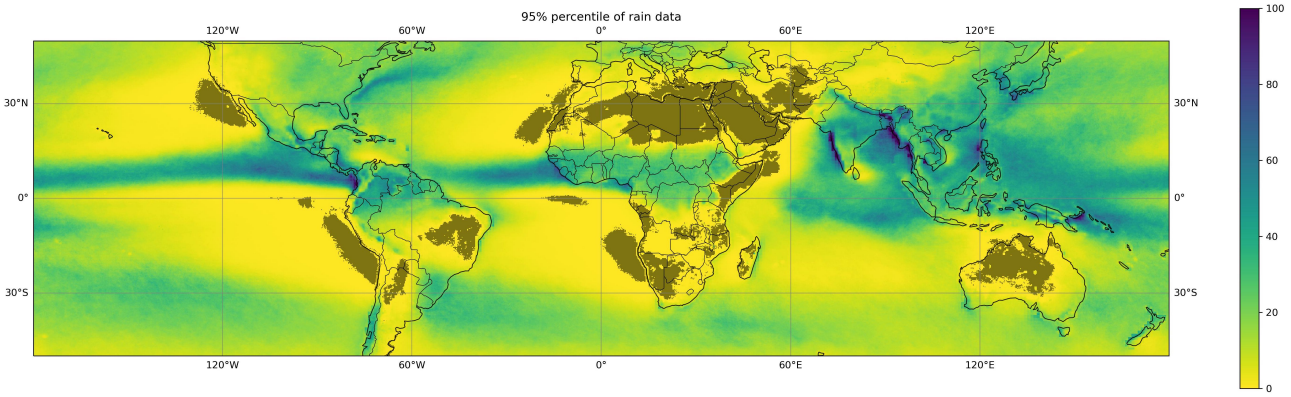

Figure 3: Figure1a, the mask part are obtained by plotting the places with percentile == 0

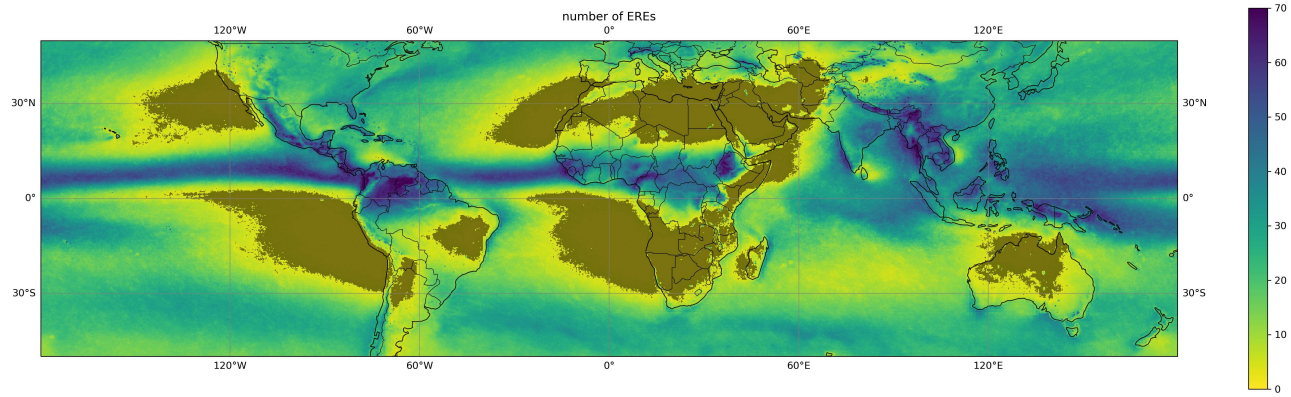

Figure 4: Figure1b, the mask part are obtained by plotting the places with ERE == 0

We can see that figure 1a has a lower similarity to that of figure 1b, especially in terms of the masked part.

However, what is really important is that the figure 1b has a very high similarity to the original figure. This means that in terms of ERE distribution, we have successfully reproduced the authors' result. As we are going to built our future results on EREs instead of percentiles, the slight variation in Figure 1a is not of great significance.

For details of how to obtain these two plots, please look at code on github.

6 Figure 2

6.1 Event Synchronization

The way to calculate the event synchronization between two nodes is to find the number of pair of EREs between two nodes that satisfy a particular threshold.

We denote a^i as the i^{th} ERE in the first node, b^j as the j^{th} ERE in the second node, and t_a^i as the date of the i^{th} ERE at the first node. The threshold is calculated as

$$\tau_{a,b}^{i,j} = \frac{1}{2}\min\{(t_a^i - t_a^{i-1}), (t_a^{i+1} - t_a^i), (t_b^j - t_b^{j-1}), (t_b^{j+1} - t_b^j)\}$$

We also require maximum temporal delay between events at different locations to be 10 days. Therefore, the final threshold we use is

$$\tau = \min\{\tau_{max}, \tau_{a,b}^{i,j}\}, \tau_{max} = 10$$

We calculate the number of pairs between two nodes satisfy their threshold, and this is the event synchronization between two nodes.

6.2 Calculation of Threshold

To build the graph meaningfully, we need to check whether the number of ES between any pair of two nodes is significantly high, so we need to build a significant test. The following steps are how we get the threshold of the significant number(s) of ES for a pair of nodes:

1. Take the numbers of EREs corresponding to the two nodes, n_1 and n_2.
2. Generate two sequences with length d. One is consist of $(d-n_1)$ 0s and n_1 1s, and the other is consist of $(d-n_2)$ 0s and n_2 1s, where d is the total number of days you considered (the number of days your data covers).
3. Randomly re-permute the two sequences and compute the number of ES between the two nodes/sequences after the random permutation.

4. Repeat the previous step 2000 times and record the number of ES after each random permutation, so you should get 2000 integers after this step.

5. Take the data at the appropriate percentile as the threshold; we took several thresholds corresponding to several percentiles so that we could use different thresholds to test the robustness of our result later. (e.g. we took 90%, 95%, 99%, 99.5%, and 99.9%)

In terms of computation efficiency, we first computed those thresholds based on the above algorithm for any combination of numbers of EREs and stored them in a matrix. We could find the corresponding threshold from the matrix based on the number of EREs two nodes have when we try to figure out whether the two nodes are significantly linked. In this way, we only need to apply the above algorithm $(78 * 78 + 78)/2$ times instead of $(576000 * 576000 + 576000)/2$. (78 because the highest number of ERE among all nodes is NOT greater than 78 in the data we considered.)

6.3 Construct the graph

We first calculate the great circle distance between each two nodes. We then use kernel density estimation to convert the discrete points into a smooth curve.

Then, we calculate the event synchronization between each two nodes, and only save the ones that are higher than the threshold of the signifcance test. Then, we define several range for these distances and calculate how many pairs are there in each range. Note that instead of saving counting 1 distance between each two points, we actually save n pairs of such distance between two nodes, where n is the value of synchronization between these two nodes. In this way, we obtain the distribution of links.

We then plot the distribution of links and the smooth curve of great circle distance between each two nodes into one graph. We fit a power law for distances less than 1000 kilometers. For large distance that larger than 2500 kilometers, we can see that the curve for distribution of links is close to the KDE curve of great circle distance.

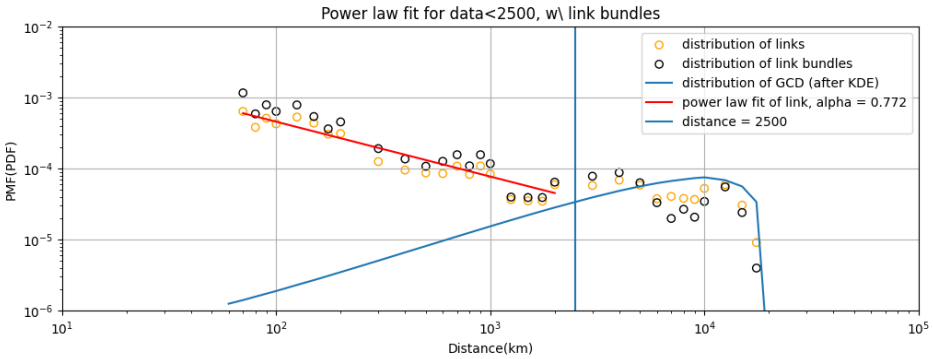

Figure 5: Extended Figure 2(between SCA and the world)

7 Figure 3

7.1 3a

Following steps are how we get fig 3a:

1. Select all links related to South central asia(SCA), and here we consider the area 27.5°N-28.5°N, 78.5°E-79.5°E.

2. Randomly select 10000 links among all links.

3. Set all the ends coordinate of selected links with value 1, and all other coordinates with value 0. According to the values, calculate the kernel density estimation(KDE) with haversine matric and gaussian kernal.

4. With calculated KDE values for all coordinates, plot links for all non-zero KDE value coordinates.

The figure below is Fig 3a we got:

7.2 3b

The way of obtaining fig 3a and 3b are slightly different. We need to generate a null model and compare the KDE value calculated from randomly selected links with null model to get link bundles.

Following steps are how we get the null model and link bundles:

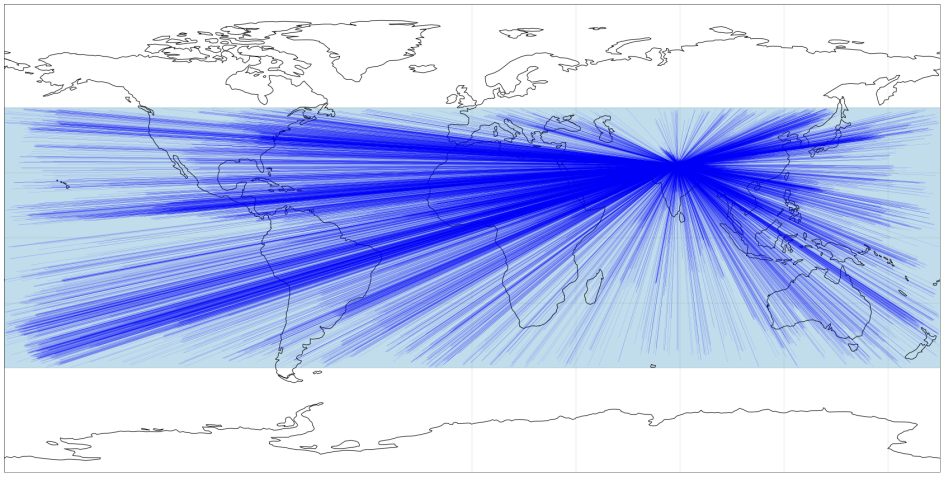

Figure 6: Figure3a

1. Get the number of randomly chosen links. Here we take 10000 links as mentioned in 3a.

2. Randomly choose 10000 coordinates from all ERE¿2 coordinates and record them.

3. Repeat step 2 for 600 times and record them. Then calculate mean, standard deviation, different percentiles according to null model.

4. Compare the KDE value we got from 3a to the null model. With comparing the mean and standard deviation values, keep non-zero values links and these links are link bundles we obtained for SCA area.

The figure below is Fig 3b we got:

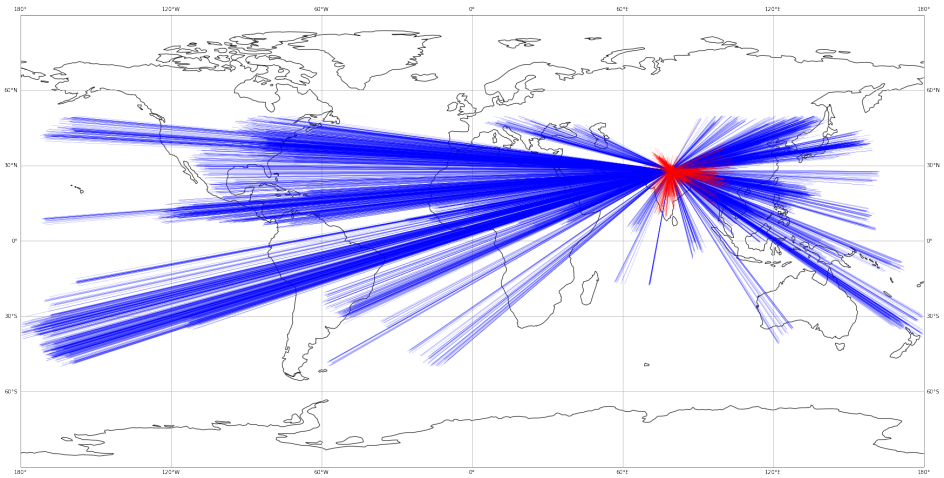

Figure 7: Figure3b

7.3 Extended figures

From these two figures, we can find that the pattern of link. Here we have some more figure which can show the similar pattern.

The following one is the version of plotting link bundles and contour map in the same figure. The following figures are contour maps of mean and standard deviation of

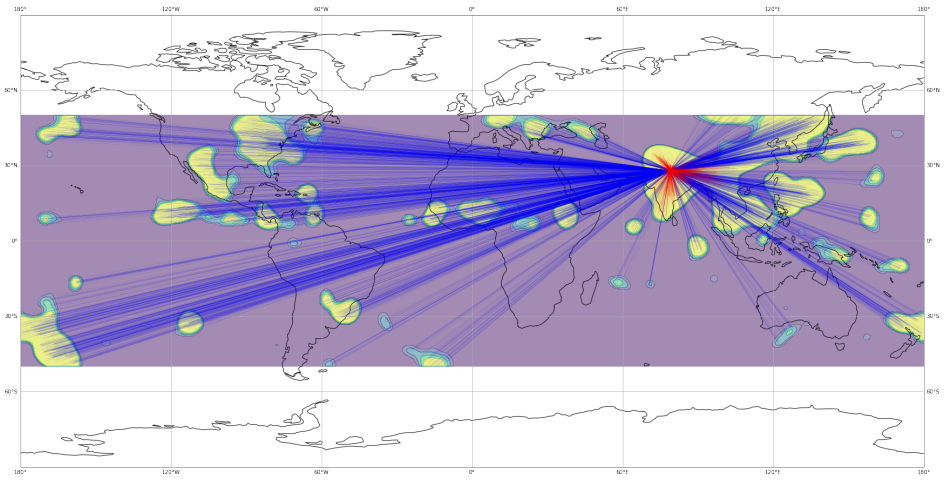

Figure 8: Figure3b with contour and link bundles

null model data we got.

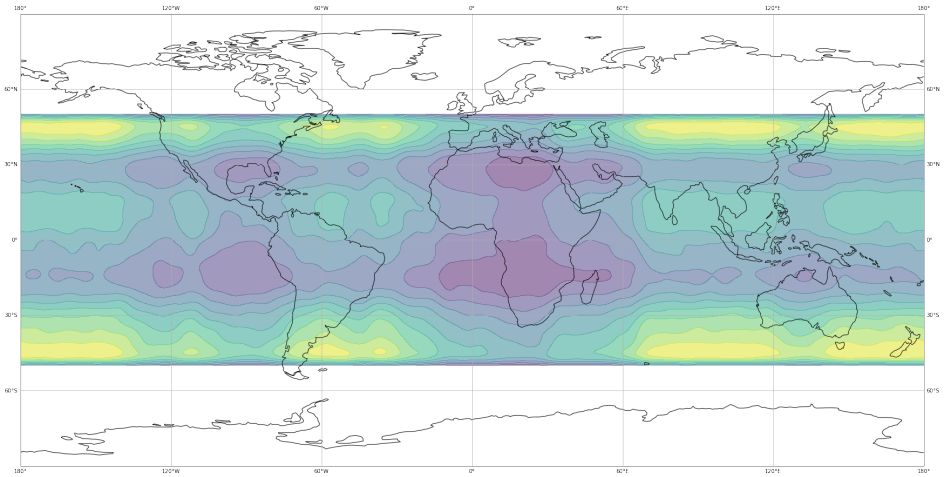

Figure 9: Null model mean

All these figures show similar pattern result as the original author's result.

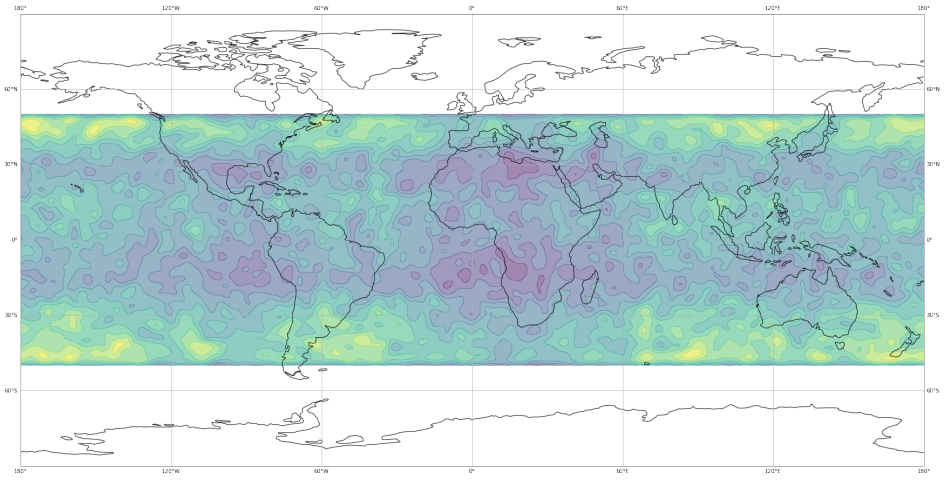

Figure 10: Null model standard deviation

8 Figure 4

8.1 4a

In this section, we are attempting to investigate the teleconnections of extremal rainfall events(EREs) between south central Asia(SCA) and Europe. For the plot 4a, we initially undertake the task of identifying and establishing distinct indices that characterize extreme rainfall events within both SCA and EU, and then compute two time series for these two regions:

1. Time series for SCA: The nth position in this time series represents the number of EREs that occurred in South Central Asia on date n.

2. Time series for Europe: The nth position in this time series represents the number of EREs that occurred in Europe on date n.

The n th position in each time series is the number of EREs happen in that region on date n. We then compute the lead-lag correlation between these two time series. This helps us understand if there is a significant relationship between EREs in South Central Asia and Europe and whether one region influences the other.

By employing these analytical steps, we aim to gain a deeper understanding of the dynamics and connections between extreme rainfall events in the specified regions, ultimately contributing to our knowledge of regional climate patterns and their interrelated

effects.

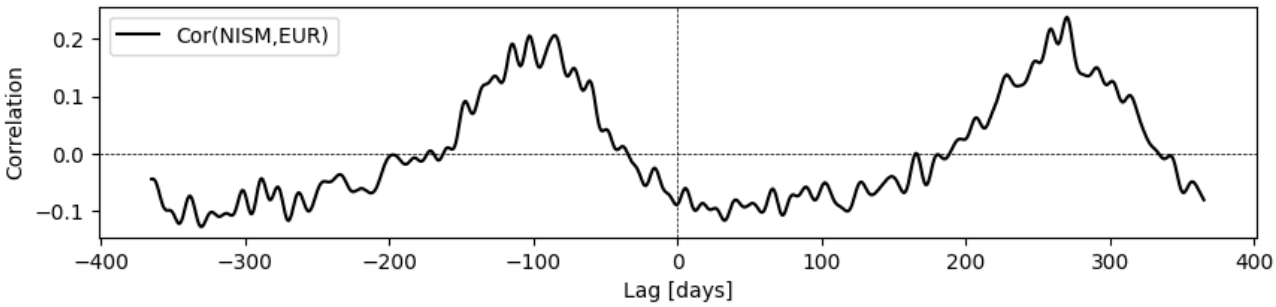

Figure 11: Figure4a

8.2 4b, 4c

The way to obtain 4c is the same as 4b. The steps needed for generating 4b will be introduced first.

First, we need to obtain our interested dates (days with high numbers of EREs in Europe that are followed by associated EREs in SCA). Here are the steps needed for generating the dates.

1. obtain the required time series (Europe to SCA), which is described in the method section (2) and (3)

2. apply the Chebyshev low-pass filter with cutoff period of 10 days to the time series, so that the filtered time series is obtained (Note that the effect of Chebyshev low-pass filter needs further consideration, because it doesn't preserve all the dates that have large values and create local maximums that shouldn't have existed.)

3. identify the local maximums in the filtered time series and select the local maximums within JJA season

4. select again from JJA local maximums so that they pass the 90th percentile of the JJA dates, obtaining our final interested dates in 4b

Note that even though authors have said in the article that in step 4 local maximums need to pass the 90th percentile of entire time series, what they did in their code is using the 90th percentile of JJA dates.

After obtaining the interested dates, the composite anomalies are needed to be computed. Here are the steps for computing anomalies.

1. Obtain the average precipitation from 1998 – 2016 in JJA season
2. compute the difference between the precipitation data of our interested dates and the JJA precipitation average
3. average the differences across the dates, obtaining the composite anomalies

From basic algebra we can notice that swapping step 2 and 3, i.e. averaging the precipitation across the interested dates and then minusing the JJA precipitation average, makes no difference to our results.

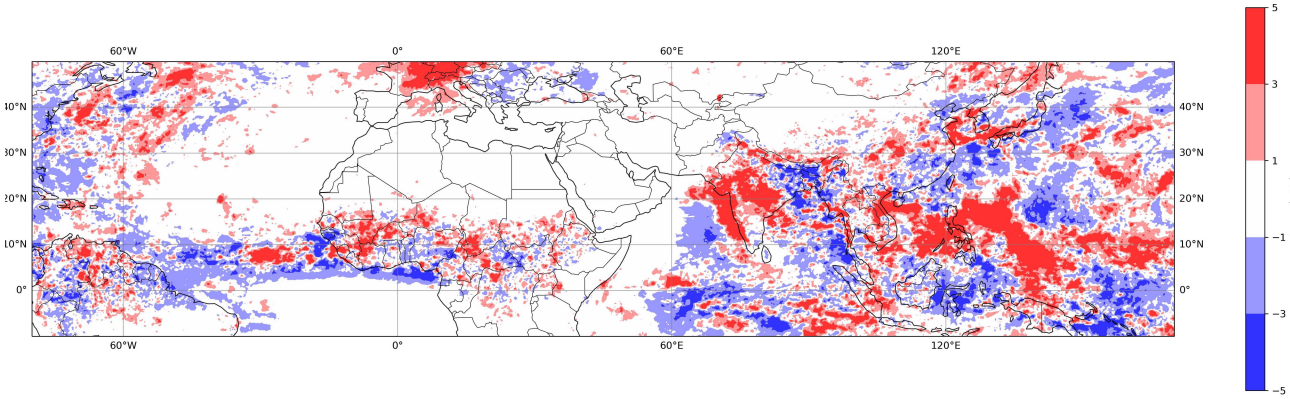

Figure 12: Figure4b

Figure 4c is obtained using the interested dates in figure 4b. By plusing 3 days to the dates in 4b, the interested dates in 4c is obtained. Then following the same steps of computing anomalies, figure 4c is obtained.

We can observe that both figures 4b and 4c display characteristics present in the original figures. For instance, in the original figures, the red color indicating high precipitation gradually diminishes in Europe after 3 days, while simultaneously becoming more prominent and consistent in SCA. This same pattern remains consistent in the figures we created.

At the same time, the differences between the two figures we produced and the original figures are both hard to be identified.

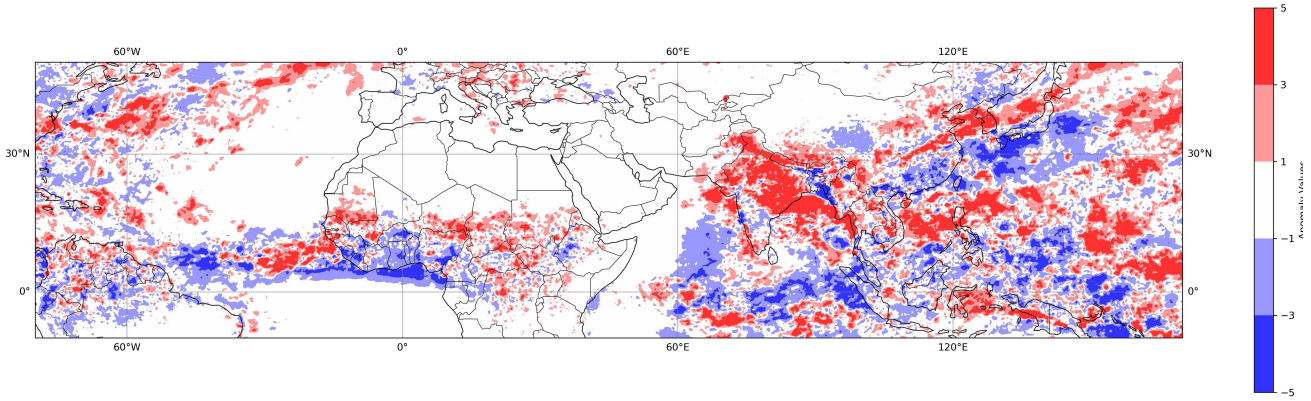

Figure 13: Figure4c

8.3 4d, 4e

Figure 4d and 4e are obtained using the same interested dates as figure 4b and 4c, so the only steps need to be done are step 1 to 3 for computing the anomalies but using v-wind speed data instead of precipitation.

Here are the two figures obtained.

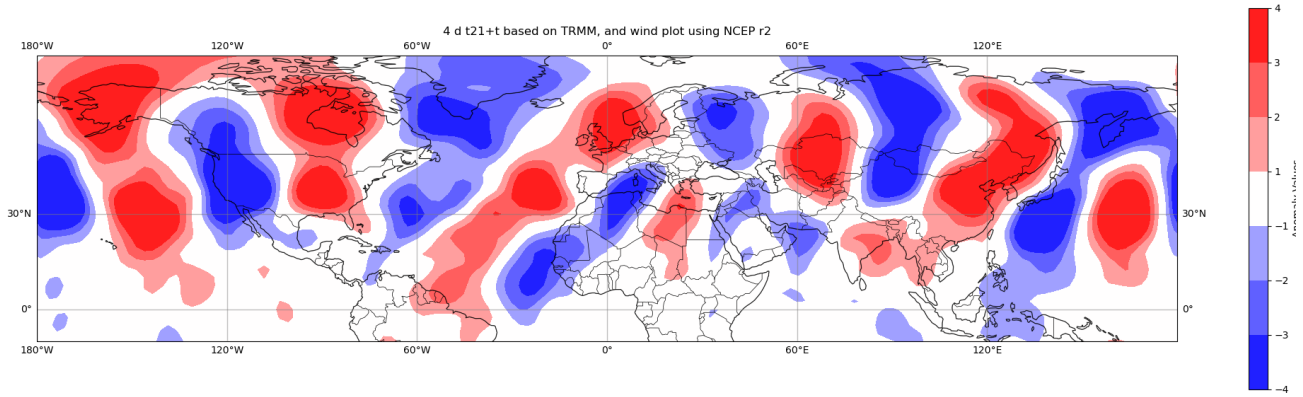

Figure 14: Figure4d

Even though the wave pattern are very significant in both figures but there are many differences between these two and the original figures.

It is possible that such differences are caused by the version difference of our v-wind data. In our code, we tried plotting using TRMM time series with NCEP r1 (reanalysis 1) data instead of r2, as well as time series based on NCEP r2 precipitation with both r1 and r2 v-wind data. However it seems all of the combinations are significantly different from that of the original results.

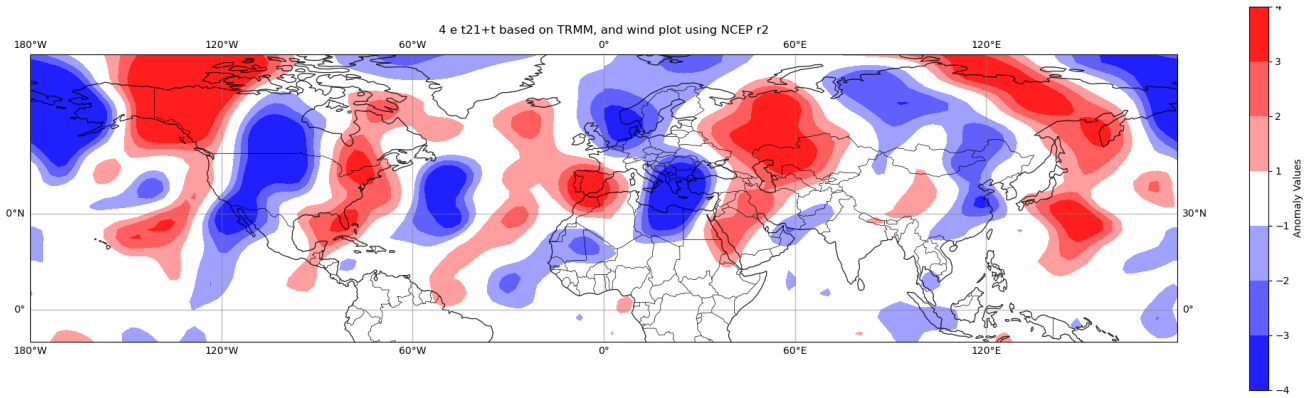

Figure 15: Figure4e

8.4 Extended figure 4b

One possible reason for the difference is due to the difference in the pre-chebychev filtered time series.

Extended figure 4b is a very good auxiliary figure for examining our pre-chebychev filtered time series. One thing to notice is that what we plot is Frequency of EREs in Europe **with synchronous subsequent counterparts in SCA**, which means that we need to exclude the ES happened on the same dates between two regions without any delay. Such time series without delay is denoted as t and the one with delay from Europe to SCA is $t21$. So what plotted here is based on $t21$.

Here is the extended figure 4 we produced. Notice that the time series is produced using TRMM data.

We can see that the percentage of September is significantly higher than that of the authors' time series, and that of July and August are lower.

As for NCEP time series, here is what we obtained.

In terms of the NCEP, percentage of September seems lower than that of original figure and those of July and August seem higher.

However, the algorithm we use to generate the time series are the same as the code authors sent us, and there are no differences found between the inputs of the algorithms. Therefore, it may be a potential task in the future to investigate the reasons behind the variance in our results compared to those of the original authors.

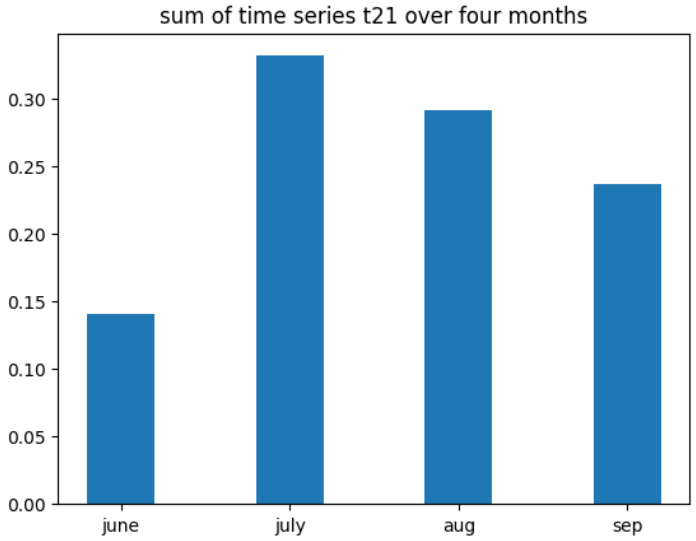

Figure 16: Extended figure4 use TRMM precipitation

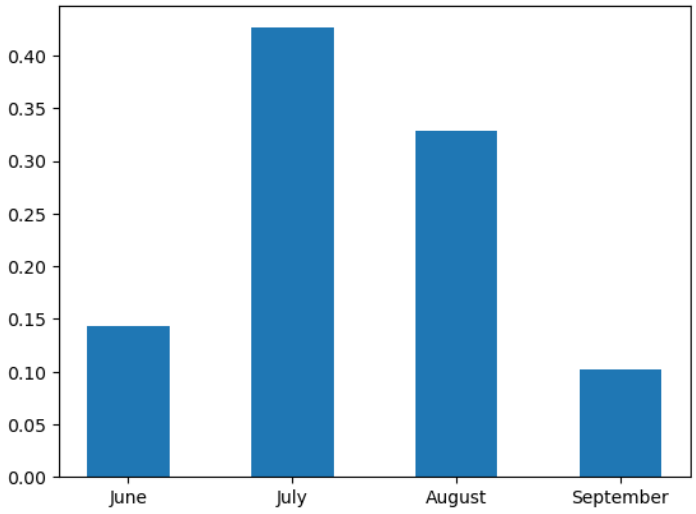

Figure 17: Extended figure4 use NCEP precipitaion

9 Summary

In our endeavor to reproduce the key findings of the article "Complex Networks Reveal Global Pattern of Extreme-rainfall Teleconnections," we embarked on a comprehensive journey that encompassed data acquisition, computational resources utilization, and meticulous figure replication.

Our first step was to secure the requisite datasets, including TRMM precipitation and NCEP reanalysis data. The TRMM dataset, in particular, demanded special attention due to its crucial role in our analysis. With the aid of high-performance computing

clusters, we efficiently managed the substantial computational demands of our project, enhancing our analytical capabilities.

Each key figure in the original article underwent rigorous replication efforts. Figure 1, depicting the global distribution of extreme rainfall events (EREs), was recreated using the cartopy package. Although Figure 1a exhibited minor discrepancies compared to the original, Figure 1b faithfully reproduced the ERE distribution, validating our efforts.

Figure 2 explored event synchronization between locations, highlighting ERE connections. Through meticulous calculations and threshold determination, we successfully generated link distributions, with distances under 1000 kilometers adhering to a power-law distribution. The figure effectively showcased the relationship between geographic distance and event synchronization.

Figures 3a and 3b delved into link bundling and composite anomalies, illustrating the association between EREs in South Central Asia and Europe. Our process of identifying significant dates and computing composite anomalies closely mirrored the original findings.

Figures 4a, 4b, 4c, 4d, and 4e investigated teleconnections between EREs and v-wind speed, involving anomaly computations and lead-lag correlation analysis. While minor variations were observed compared to the original figures, the fundamental patterns and trends remained intact.

Notably, Extended Figure 4b shed light on time series data used in the analysis, revealing differences in percentage values for specific months. These variations warrant further scrutiny.

In summary, our extensive efforts to replicate the main findings of the article yielded results that closely aligned with the original study. Despite minor divergences in some figures, our ability to capture the overarching trends and patterns underscores the significance of transparent and reproducible research. This project underscores the complexities and challenges associated with replicating intricate analyses in the realm of scientific research.

10 Code

Github Page. If you have any questions about the code or the report, you can contact Dr. Ray or any of the Group members through email.