

川渝地区 夏季高温干旱复合天气事件分布图

Distribution map of Summer Heat and Drought Combined Weather

Events in Sichuan and Chongqing Region

过程操作文档

目录

1 制图原因与数据处理原理	2
1.1 制图原因	2
1.2 数据处理原理	2
2 制图流程	2
3 具体制图过程	3
3.1 数据使用	3
3.2 数据处理	3
3.2.1 单月数据提取和按掩膜提取	3
3.2.2 判断高温干旱复合天气事件	5
3.2.3 单月数据联合为夏季	9
3.2.4 频率和频数统计	18
3.3 数据制图	33
文献和数据引用	37

1 制图原因与数据处理原理

1.1 制图原因

进入 21 世纪以来，川渝地区发生了多次极端高温干旱事件，如 2006 年川渝地区遭遇了当时有气象记录以来最严重的高温干旱事件。2022 年夏季，西南地区乃至整个长江流域再次发生了罕见的重大高温干旱复合事件，高温突破了历史记录，干旱事件导致长江中上游干流和主要支流的流量为历史最少。^[1]

高温干旱复合天气事件不仅会诱发其他自然灾害，还会给川渝地区带去严重的经济损失，此外，其中的四川省还是水电大省，高温干旱复合天气事件会导致四川省河流流量减少，水库水位降低，从而导致用电紧张。

因此，研究高温干旱复合天气事件的时空分布规律和变化趋势对帮助干旱减灾和有效应对气候变化具有重要意义。为了帮助研究高温干旱复合天气事件的时空分布规律和变化趋势，需要制作有关地图来帮助研究分析。本次制图的时间范围为 1982 年至 2019 年。

1.2 数据处理原理

研究高温干旱复合天气事件需要使用到标准化温度指数（STI）和标准化降水指数（SPI）来综合判断高温干旱复合天气事件。通过月平均气温来计算 STI 指数的公式如下：

$$STI = (X_t - \mu_t) / \sigma_t$$

其中， X_t 为当前月的平均气温， μ_t 为该栅格点的气温平均值， σ_t 为该栅格点的平均气温标准差。

通过月平均降水来计算 SPI 指数的公式如下：

$$SPI = (X_p - \mu_p) / \sigma_p$$

其中， X_p 为当前月的平均降水， μ_p 为该栅格点的降水平均值， σ_p 为该栅格点的平均降水标准差。

通过 STI 和 SPI 指数判断高温干旱复合天气事件的标准如表 1 所示：

表 1 高温干旱复合天气事件判断依据

等级	SPI	STI	类型
1	$-0.5 < SPI$	$STI \leq 0.5$	无
2	$-1.0 < SPI \leq -0.5$	$0.5 < STI \leq 1.0$	轻度
3	$-1.5 < SPI \leq -1.0$	$1.0 < STI \leq 1.5$	中度
4	$-2.0 < SPI \leq -1.5$	$1.5 < STI \leq 2.0$	重度
5	$SPI \leq -2.0$	$STI > 2.0$	极端

2 制图流程

本次制图将使用 MATLAB 进行数据处理以及使用 ArcMap 进行制图与地图输

出，数据处理与制图流程如图 1 所示。

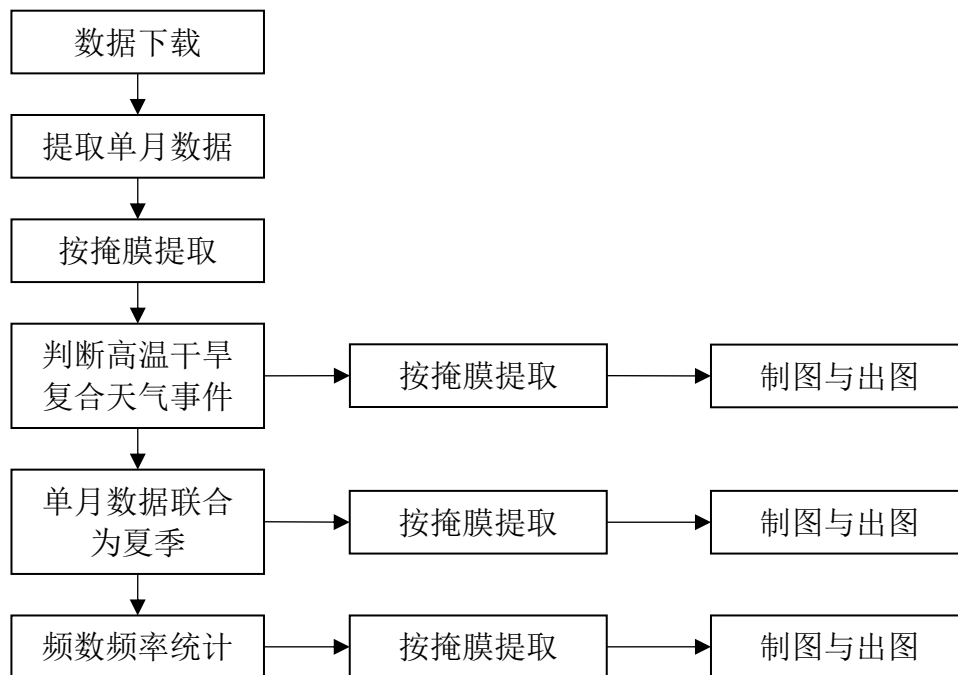


图 1 数据处理与制图流程

3 具体制图过程

3.1 数据使用

(1) 月平均气温和降水数据来源：

本次制图所使用的数据为彭守璋老师在国家青藏高原科学数据中心（TPDC）上发布的中国 1km 分辨率逐月平均气温数据集（1901-2022）和中国 1km 分辨率逐月降水量数据集（1901-2022）。^[2-7]

(2) 川渝地区省级和市级 Shapefile 文件来源：

本次制图从阿里云提供的地图选择器网站来获取川渝地区的省级和市级 Shapefile 文件。^[8]

(3) 川渝地区地级市点集数据 Shapefile 文件来源：

本次制图使用的地级市点集数据来自微信公众号：立方数据学社。

3.2 数据处理

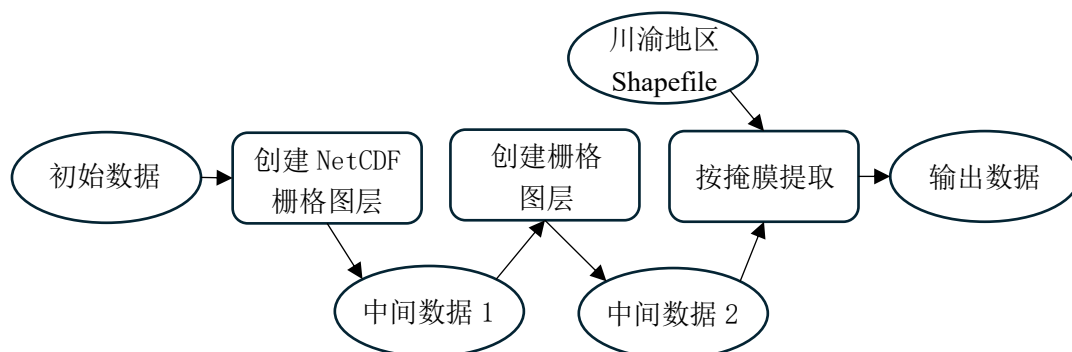
3.2.1 单月数据提取和按掩膜提取

(1) 简介：本次制图使用的数据类型为 NetCDF 文件，该类型的数据一般为三维数据，分别是经度、纬度和时间，因此需要将需要的数据从中提取出来。本次制图使用的数据为全国数据，因此还需以川渝地区的 Shapefile 为掩膜裁切数据。该步骤使用的软件为 ArcMap。

(2) 主要使用工具:

多维工具→创建 NetCDF 栅格图层、数据管理工具→图层和表视图→创建栅格图层、Spatial Analyst 工具→提取分析→按掩膜提取

(3) 操作流程:



(4) 各工具参数设置:

①创建 NetCDF 栅格图层:

1) 输入 netCDF 文件: 输入初始数据

2) 变量: 如果输入的是降水数据则为 pre, 如果是温度则为 tmp, 一般选择完文件后自动输入

3) X 维度: lon (经度)

4) Y 维度: lat (纬度)

5) 输出的栅格图层: 在此输入一个输出栅格的名称。

6) 波段维度 (可选): 在这里选择 time。

其余参数均可保持不变。

②创建栅格图层:

1) 输入栅格: 选择上一步输出的栅格图层。

2) 输出的栅格名称: 在此输入一个输出栅格的名称。

3) 波段: 输入 6 (6 月)。

说明: 本次制图使用的 NetCDF 文件中, 1982-2017 年是每三年一个文件, 2018, 2019 两年均为一年一个文件。因此, 1982-2017 年需要输入 6, 7, 8, 18, 19, 20, 30, 31, 32 (6 代表 6 月, 对应加上 12 则为下一年的 6 月, 7 月, 8 月再下一年也是如此)。2018 年和 2019 年只需要输入 6, 7, 8 即可。注意, 每次操作只能提取一个月的数据, 不能一次性输入全部波段。

其余参数均可保持不变。

③按掩膜提取:

1) 输入栅格: 选择上一步输出的栅格图层。

2) 输入栅格数据或要素掩膜数据: 选择川渝地区的 Shapefile 文件。

3) 输出栅格: 在这里输入输出路径和文件名称。(建议: 将文件保存在地理文件数据库中)

(5) 后续操作：完成上面的操作后，将输出数据加载进图层中，右键点击图层中的文件选择数据导出，将它们输出为 TIF 文件以进行下一步操作。

3.2.2 判断高温干旱复合天气事件

(1) 简介：本次制图所使用的数据为栅格数据，且分辨率为 1km，这意味着数据量非常大，因此接下来的数据处理需要使用 MATLAB 软件来进行。

(2) MATLAB 代码结构：



(3) MATLAB 代码说明：该代码需要输入存有月平均气温和月平均降水的栅格文件 (.tif)，代码将读取栅格文件中的栅格数据和地理信息，随后代码将自动计算 STI 和 SPI 并输出图像供使用者判断是否计算正确，然后通过计算得来的 STI 和 SPI 赋值为 1-5，分别对应着无、轻度、中度、重度、极端，并以此综合判断高温干旱复合天气事件，最终会输出一个新的储存有高温干旱复合天气事件数据的栅格文件。

(4) MATLAB 高温干旱复合天气事件判断代码：

```
clc
clear
%% SPI 部分
% 程序交互
disp('选择文件前，请将 MATLAB 工作目录设置到 TIFF 文件所在目录，否则将报错！');
disp('-----')
tiff_path_SPI = uigetfile('*.tif', '选择一个包含降水数据的 TIFF 文件',
'MultiSelect', 'on');
if iscell(tiff_path_SPI)
    clear disp('多个 TIFF 文件被选择。暂时无法处理多个 TIFF 文件。');
    return;
end

if tiff_path_SPI == 0
    disp('未选择任何文件。');
    return;
end

disp(['已选择的文件： ', tiff_path_SPI]);

if ~exist(tiff_path_SPI, 'file')
    disp(['错误，无法找到 ', tiff_path_SPI, ' 文件 ']);
    disp('程序已结束');
    return;
```

```

end

% 读取 TIFF 文件的信息
tiff_info_SPI = imfinfo(tiff_path_SPI);
num_rows_SPI = tiff_info_SPI.Height;
num_cols_SPI = tiff_info_SPI.Width;
num_bands_SPI = numel(tiff_info_SPI);
data_SPI = zeros(num_rows_SPI, num_cols_SPI, num_bands_SPI, 'double');
[A2,R2] = readgeoraster(tiff_path_SPI);

for band = 1:num_bands_SPI
    data_SPI(:,:,band) = imread(tiff_path_SPI, band);
end

% 自动读取栅格文件中的最值，将其设置为空值
min_val_SPI = min(data_SPI(:));
max_val_SPI = max(data_SPI(:));
data_SPI(data_SPI == min_val_SPI | data_SPI == max_val_SPI) = NaN;

% 计算所有栅格的总的平均值
X_SPI = mean(data_SPI(:), 'omitnan');

% 计算每个栅格数据与总平均值的标准差
e_SPI = std(data_SPI(:), 'omitnan');

% 计算 SPI
SPI = (data_SPI - X_SPI) ./ e_SPI;

% 显示 SPI 结果
disp('-----')
disp('SPI 已计算完毕。');
% 显示 SPI 表格
figure;
imagesc(SPI);
colorbar;
title('Standardized Precipitation Index (SPI) Grid');
xlabel('Columns');
ylabel('Rows');

%% STI 部分
% 程序交互
disp('选择文件前，请将 MATLAB 工作目录设置到 TIFF 文件所在目录，否则将报错！');
disp('-----')

```

```

tiff_path_STI = uigetfile('*.tif', '选择一个包含温度数据 TIFF 文件',
'MultiSelect', 'on');
if iscell(tiff_path_STI)
    disp('多个 TIFF 文件被选择。暂时无法处理多个 TIFF 文件。');
    return;
end

if tiff_path_STI == 0
    disp('未选择任何文件。');
    return;
end

disp(['已选择的文件: ', tiff_path_STI]);

if ~exist(tiff_path_STI, 'file')
    disp(['错误, 无法找到 ', tiff_path_STI, ' 文件 ']);
    disp('程序已结束');
    return;
end

%读取 TIFF 文件的信息
tiff_info_STI = imfinfo(tiff_path_STI);
num_rows_STI = tiff_info_STI.Height;
num_cols_STI = tiff_info_STI.Width;
num_bands_STI = numel(tiff_info_STI);
data_STI = zeros(num_rows_STI, num_cols_STI, num_bands_STI, 'double');
[A1,R1] = readgeoraster(tiff_path_STI);

for band = 1:num_bands_STI
    data_STI(:,:,band) = imread(tiff_path_STI, band);
end

% 自动读取栅格文件中的最值, 将其设置为空值
min_val_STI = min(data_STI(:));
max_val_STI = max(data_STI(:));
data_STI(data_STI == min_val_STI | data_STI == max_val_STI) = NaN;

% 计算所有栅格的总的平均值 (X)
X_STI = mean(data_STI(:), 'omitnan');

% 计算每个栅格数据与总平均值的标准差
e_STI = std(data_STI(:), 'omitnan');

% 计算 STI

```



```

STI = (data_STI - X_STI) ./ e_STI;

% 显示 STI 结果
disp('-----')
disp('STI 已计算完毕。');
% 显示 STI 表格
figure;
imagesc(STI);
colorbar;
title('Standardized Temperature Index (STI) Grid');
xlabel('Columns');
ylabel('Rows');

%% 判断高温干旱复合天气事件
event = ones(num_rows_STI, num_cols_STI);
% 轻度
event((SPI > -1.0 & SPI <= -0.5) & (STI > 0.5 & STI <= 1.0)) = 2;
% 中度
event((SPI > -1.5 & SPI <= -1.0) & (STI > 1.0 & STI <= 1.5)) = 3;
% 重度
event((SPI > -2.0 & SPI <= -1.5) & (STI > 1.5 & STI <= 2.0)) = 4;
% 极端
event(SPI <= -2.0 & STI > 2.0) = 5;

% 显示结果
disp('-----')
disp('高温干旱复合事件等级已判断完毕。');

% 显示结果表格
figure;
imagesc(event);
colorbar('Ticks', [1 2 3 4 5], 'TickLabels', {'正常', '轻度', '中度', '重度', '极端'});
title('高温干旱复合事件等级');
xlabel('Columns');
ylabel('Rows');

%保存高温干旱复合天气事件为 TIFF 文件
[output_filename, output_path] = uinputfile('*.tif', '保存高温干旱复合天气事件数据为 TIFF 文件');
if isequal(output_filename, 0)
    disp('保存操作被取消。');
else
    output_tiff_path = fullfile(output_path, output_filename);

```

```

geotiffwrite(output_tiff_path, event, R2);
disp(['高温干旱复合天气事件数据已成功保存为 TIFF 文件: ',
output_tiff_path]);
end

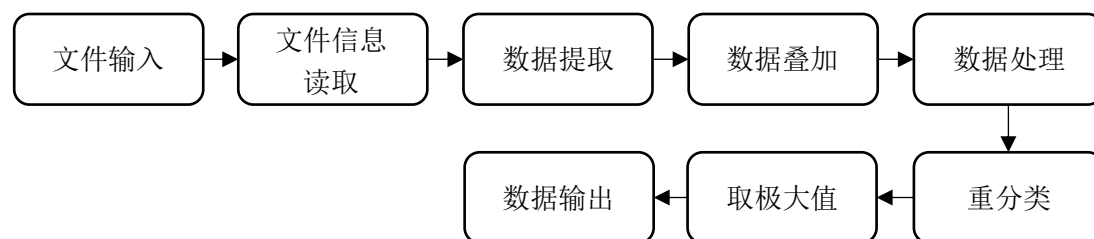
```

3.2.3 单月数据联合为夏季

(1) 简介：刚才那一步只是计算了夏季（6，7，8 月）中一个月的数据，想要综合判断一整个夏季的高温干旱复合天气事件的时空分布和变化趋势，还需要将三个月的数据联合成一个夏季的数据。

(2) MATLAB 代码逻辑：一个栅格点如果三次都为 1（无），那么就认为该栅格点在夏季没有发生过高温干旱复合天气事件，再次赋值为 1；一个栅格点如果在其中某一个月发生过一次轻度、中度、重度或极端就认为该栅格点在夏季发生了轻度、中度、重度或极端高温干旱复合天气事件，发生的等级取最高的那一次。

(3) MATLAB 代码结构：



(4) MATLAB 代码说明：该代码需要输入上一段代码输出的栅格文件(.tif)，需要分别输入 6，7，8 月的栅格文件。不要使用任何 GIS 软件对其进行修改，否则可能会影响计算结果。随后，代码会自动读取栅格文件中的栅格数据以及地理信息并将三个月的高温干旱复合天气事件数据联合成新的栅格数据并输出为栅格文件。

(5) MATLAB 单月联合为季节的代码：

```

clc
clear
%% 提示模块
choice = questdlg('选择文件前，请将 MATLAB 工作目录设置到 TIFF 文件所在目录，否则将报错!', '提示', '继续运行', '停止运行', '停止运行');
switch choice
    case '继续运行'
        %% 文件处理模块(6 月)
        % 交互模块
        tiff_path_6 = uigetfile('*.tif', '选择一个包含高温干旱复合事件数据的 TIFF 文件 (6 月)', 'MultiSelect', 'on');
        if iscell(tiff_path_6)
            disp('多个 TIFF 文件被选择。暂时无法处理多个 TIFF 文件。');
            return;

```

```

end

if tiff_path_6 == 0
    disp('未选择任何文件。');
    return;
end

disp(['已选择的文件： ', tiff_path_6]);

if ~exist(tiff_path_6, 'file')
    disp(['错误，无法找到 ', tiff_path_6, ' 文件 ']);
    disp('程序已结束');
    return;
end

%信息读取模块
tiff_info_6 = imfinfo(tiff_path_6);
num_rows_6 = tiff_info_6.Height;
num_cols_6 = tiff_info_6.Width;
num_bands_6 = numel(tiff_info_6);
data6 = zeros(num_rows_6, num_cols_6, num_bands_6, 'double');
[A6,R6] = readgeoraster(tiff_path_6);

for band = 1:num_bands_6
    data6(:,:,band) = imread(tiff_path_6, band);
end

%% 文件处理模块(7月)
% 交互模块
tiff_path_7 = uigetfile('*.tif', '选择一个包含高温干旱复合事件数据的 TIFF 文件 (7月)', 'MultiSelect', 'on');
if iscell(tiff_path_7)
    disp('多个 TIFF 文件被选择。暂时无法处理多个 TIFF 文件。');
    return;
end

if tiff_path_7 == 0
    disp('未选择任何文件。');
    return;
end

disp(['已选择的文件： ', tiff_path_7]);

if ~exist(tiff_path_7, 'file')

```

```

        disp(['错误, 无法找到 ', tiff_path_7, ' 文件 ']);
        disp('程序已结束');
        return;
    end

% 信息读取模块
tiff_info_7 = imfinfo(tiff_path_7);
num_rows_7 = tiff_info_7.Height;
num_cols_7 = tiff_info_7.Width;
num_bands_7 = numel(tiff_info_7);
data7 = zeros(num_rows_7, num_cols_7, num_bands_7, 'double');
[A7,R7] = readgeoraster(tiff_path_7);

for band = 1:num_bands_7
    data7(:,:,band) = imread(tiff_path_7, band);
end

%% 文件处理模块(8月)
% 交互模块
tiff_path_8 = uigetfile('*.tif', '选择一个包含高温干旱复合事件数据的 TIFF 文件 (8月)', 'MultiSelect', 'on');
if iscell(tiff_path_8)
    disp('多个 TIFF 文件被选择。暂时无法处理多个 TIFF 文件。');
    return;
end

if tiff_path_8 == 0
    disp('未选择任何文件。');
    return;
end

disp(['已选择的文件: ', tiff_path_8]);

if ~exist(tiff_path_8, 'file')
    disp(['错误, 无法找到 ', tiff_path_8, ' 文件 ']);
    disp('程序已结束');
    return;
end

% 信息读取模块
tiff_info_8 = imfinfo(tiff_path_8);
num_rows_8 = tiff_info_8.Height;
num_cols_8 = tiff_info_8.Width;
num_bands_8 = numel(tiff_info_8);

```

```

data8 = zeros(num_rows_8, num_cols_8, num_bands_8, 'double');
[A8,R8] = readgeoraster(tiff_path_8);

for band = 1:num_bands_8
    data8(:, :, band) = imread(tiff_path_8, band);
end

%% 数据处理模块
% 提取数据(6月)
% 正常
data_6n = zeros(size(tiff_path_6));% 新建空白栅格
for i6n = 1:size(data6, 1)
    for j6n = 1:size(data6, 2)
        if data6(i6n, j6n) == 1
            data_6n(i6n, j6n) = data6(i6n, j6n);
        else
            data_6n(i6n, j6n) = 0;
        end
    end
end
end
% 轻度
data_6mi = zeros(size(tiff_path_6));% 新建空白栅格
for i6mi = 1:size(data6, 1)
    for j6mi = 1:size(data6, 2)
        if data6(i6mi, j6mi) == 2
            data_6mi(i6mi, j6mi) = data6(i6mi, j6mi);
        else
            data_6mi(i6mi, j6mi) = 0;
        end
    end
end
end
% 中度
data_6m = zeros(size(tiff_path_6));% 新建空白栅格
for i6m = 1:size(data6, 1)
    for j6m = 1:size(data6, 2)
        if data6(i6m, j6m) == 3
            data_6m(i6m, j6m) = data6(i6m, j6m);
        else
            data_6m(i6m, j6m) = 0;
        end
    end
end
end
% 重度
data_6s = zeros(size(tiff_path_6));% 新建空白栅格

```

```

for i6s = 1:size(data6, 1)
    for j6s = 1:size(data6, 2)
        if data6(i6s, j6s) == 4
            data_6s(i6s, j6s) = data6(i6s, j6s);
        else
            data_6s(i6s, j6s) = 0;
        end
    end
end
end
% 极端
data_6e = zeros(size(tiff_path_6));% 新建空白栅格
for i6e = 1:size(data6, 1)
    for j6e = 1:size(data6, 2)
        if data6(i6e, j6e) == 5
            data_6e(i6e, j6e) = data6(i6e, j6e);
        else
            data_6e(i6e, j6e) = 0;
        end
    end
end
end
% 提取数据(7月)
% 正常
data_7n = zeros(size(tiff_path_7));% 新建空白栅格
for i7n = 1:size(data7, 1)
    for j7n = 1:size(data7, 2)
        if data7(i7n, j7n) == 1
            data_7n(i7n, j7n) = data7(i7n, j7n);
        else
            data_7n(i7n, j7n) = 0;
        end
    end
end
end
% 轻度
data_7mi = zeros(size(tiff_path_7));% 新建空白栅格
for i7mi = 1:size(data7, 1)
    for j7mi = 1:size(data7, 2)
        if data7(i7mi, j7mi) == 2
            data_7mi(i7mi, j7mi) = data7(i7mi, j7mi);
        else
            data_7mi(i7mi, j7mi) = 0;
        end
    end
end
end
end

```

```

% 中度
data_7m = zeros(size(tiff_path_7));% 新建空白栅格
for i7m = 1:size(data7, 1)
    for j7m = 1:size(data7, 2)
        if data7(i7m, j7m) == 3
            data_7m(i7m, j7m) = data7(i7m, j7m);
        else
            data_7m(i7m, j7m) = 0;
        end
    end
end
% 重度
data_7s = zeros(size(tiff_path_7));% 新建空白栅格
for i7s = 1:size(data7, 1)
    for j7s = 1:size(data7, 2)
        if data7(i7s, j7s) == 4
            data_7s(i7s, j7s) = data7(i7s, j7s);
        else
            data_7s(i7s, j7s) = 0;
        end
    end
end
% 极端
data_7e = zeros(size(tiff_path_7));% 新建空白栅格
for i7e = 1:size(data7, 1)
    for j7e = 1:size(data7, 2)
        if data7(i7e, j7e) == 5
            data_7e(i7e, j7e) = data7(i7e, j7e);
        else
            data_7e(i7e, j7e) = 0;
        end
    end
end
% 提取数据(8月)
% 正常
data_8n = zeros(size(tiff_path_8));% 新建空白栅格
for i8n = 1:size(data8, 1)
    for j8n = 1:size(data8, 2)
        if data8(i8n, j8n) == 1
            data_8n(i8n, j8n) = data8(i8n, j8n);
        else
            data_8n(i8n, j8n) = 0;
        end
    end
end

```

```

        end
    end
    % 轻度
    data_8mi = zeros(size(tiff_path_8));% 新建空白栅格
    for i8mi = 1:size(data8, 1)
        for j8mi = 1:size(data8, 2)
            if data8(i8mi, j8mi) == 2
                data_8mi(i8mi, j8mi) = data8(i8mi, j8mi);
            else
                data_8mi(i8mi, j8mi) = 0;
            end
        end
    end
    % 中度
    data_8m = zeros(size(tiff_path_8));% 新建空白栅格
    for i8m = 1:size(data8, 1)
        for j8m = 1:size(data8, 2)
            if data8(i8m, j8m) == 3
                data_8m(i8m, j8m) = data8(i8m, j8m);
            else
                data_8m(i8m, j8m) = 0;
            end
        end
    end
    % 重度
    data_8s = zeros(size(tiff_path_8));% 新建空白栅格
    for i8s = 1:size(data8, 1)
        for j8s = 1:size(data8, 2)
            if data8(i8s, j8s) == 4
                data_8s(i8s, j8s) = data8(i8s, j8s);
            else
                data_8s(i8s, j8s) = 0;
            end
        end
    end
    % 极端
    data_8e = zeros(size(tiff_path_8));% 新建空白栅格
    for i8e = 1:size(data8, 1)
        for j8e = 1:size(data8, 2)
            if data8(i8e, j8e) == 5
                data_8e(i8e, j8e) = data8(i8e, j8e);
            else
                data_8e(i8e, j8e) = 0;
            end
        end
    end

```



```

        end
    end

    % 叠加数据
    data_normal = data_6n + data_7n + data_8n;
    data_mile = data_6mi + data_7mi + data_8mi;
    data_moderate = data_6m + data_7m + data_8m;
    data_serious = data_6s + data_7s + data_8s;
    data_extreme = data_6e + data_7e + data_8e;

    % 叠加数据处理
    % data_normal
    data_normal_R = zeros(size(data_normal));% 新建空白栅格
    for inr = 1:size(data_normal,1)
        for jnr = 1:size(data_normal,2)
            if data_normal(inr,jnr) == 3
                data_normal_R(inr,jnr) = data_normal(inr,jnr);
            else
                data_normal_R(inr,jnr) = 0;
            end
        end
    end

    % data_mile
    data_mile_R = zeros(size(data_mile));% 新建空白栅格
    for imir = 1:size(data_mile,1)
        for jmir = 1:size(data_mile,2)
            if data_mile(imir,jmir) == 2 || data_mile(imir,jmir) == 4 ||
data_mile(imir,jmir) == 6
                data_mile_R(imir,jmir) = data_mile(imir,jmir);
            else
                data_mile_R(imir,jmir) = 0;
            end
        end
    end

    % data_moderate
    data_moderate_R = zeros(size(data_moderate));% 新建空白栅格
    for imr = 1:size(data_moderate,1)
        for jmr = 1:size(data_moderate,2)
            if data_moderate(imr,jmr) == 3 || data_moderate(imr,jmr) == 6 ||
data_moderate(imr,jmr) == 9
                data_moderate_R(imr,jmr) = data_moderate(imr,jmr);
            else
                data_moderate_R(imr,jmr) = 0;
            end
        end
    end

```

```

        end
    end
    % data_serious
    data_serious_R = zeros(size(data_serious));% 新建空白栅格
    for isr = 1:size(data_serious,1)
        for jsr = 1:size(data_serious,2)
            if data_serious(isr,jsr) == 4 || data_serious(isr,jsr) == 8 ||
data_serious(isr,jsr) == 12
                data_serious_R(isr,jsr) = data_serious(isr,jsr);
            else
                data_serious_R(isr,jsr) = 0;
            end
        end
    end
    % data_extreme
    data_extreme_R = zeros(size(data_extreme));% 新建空白栅格
    for ier = 1:size(data_extreme,1)
        for jer = 1:size(data_extreme,2)
            if data_extreme(ier,jer) == 5 || data_extreme(ier,jer) == 10 ||
data_extreme(ier,jer) == 15
                data_extreme_R(ier,jer) = data_extreme(ier,jer);
            else
                data_extreme_R(ier,jer) = 0;
            end
        end
    end
    %% 重分类模块
    data_normal_R(data_normal_R > 0) = 1;
    % 特殊处理
    data_mile_R(data_mile_R > 0) = 2;
    data_moderate_R(data_moderate_R > 0) = 3;
    data_serious_R(data_serious_R > 0) = 4;
    data_extreme_R(data_extreme_R > 0) = 5;

    %% 数据计算模块
    max_event = max(data_mile_R,data_moderate_R);
    max_event2 = max(max_event,data_serious_R);
    max_event3 = max(max_event2,data_extreme_R);
    event_union = data_normal_R + max_event3;

    %% 图像输出模块
    figure;
    imagesc(event_union);
    colorbar;

```

```

title('夏季高温干旱复合事件');
xlabel('Columns');
ylabel('Rows');

%% TIFF 文件输出模块
[output_filename, output_path] = uinputfile('*.tif', '保存夏季高温干旱复合天气
事件数据为 TIFF 文件');
if isequal(output_filename, 0)
    disp('保存操作被取消。');
else
    output_tiff_path = fullfile(output_path, output_filename);
    geotiffwrite(output_tiff_path, event_union, R6);
    disp(['夏季高温干旱复合天气事件数据已成功保存为 TIFF 文件: ',
output_tiff_path]);
end

case '停止运行'
    disp('程序已停止运行')
    return
end

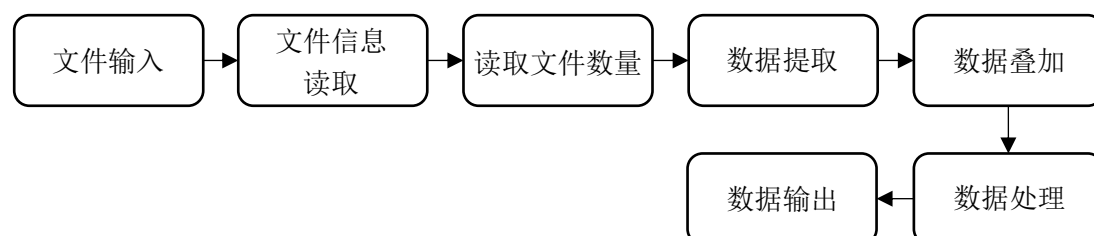
```

3.2.4 频率和频数统计

(1)简介:为了更好的分析高温干旱复合天气事件的时空分布和变化趋势,还需要统计高温干旱复合天气事件各等级的发生频数。

(2)MATLAB 代码逻辑:由于本次制图将无到极端分别赋值为 1-5,因此,将所有的高温干旱复合天气事件的数据相加后,无到极端的值分别是 1、2、3、4、5 的倍数,将各个等级的值提取出来分别相加然后再除以它们的倍数即可得到各个等级发生的频数,将相加后的值除以输入的栅格文件数即可得到各等级高温干旱复合天气事件发生的频率。

(3) MATLAB 代码结构:



(4) MATLAB 代码说明:该代码可以读取多个栅格文件,可以使用单月数据,也可以使用季节数据,但是必须使用 MATLAB 输出的栅格数据,不要使用任何 GIS 软件对其进行修改以免产生错误。输入栅格数据后,代码会自动计算频数和频率并输出数据。其中频数计算代码 A 型是计算除去等级为无的全部等级的频数的代码, F 型是单独计算除去等级为无的全部等级的频数的代码。

(5-1) MATLAB 频数计算代码 A 型:

```
clc
clear
%% 提示模块
choice = questdlg('请将 TIFF 文件保存在一个文件夹中，文件夹中不要有其它文件', '提示', '继续运行', '停止运行', '停止运行');
switch choice
    case '继续运行'
%% 交互模块
folderPath = uigetdir('选择栅格文件所在的文件夹');
if folderPath == 0
    disp('未选择任何文件夹。');
    return;
end
tiffFiles = findFilesRecursive(folderPath, '.tif');
images = cell(numel(tiffFiles), 1);
h = waitbar(0, '进行中', 'Name', '文件读取进度');
for i = 1:numel(tiffFiles)
    images{i} = imread(tiffFiles{i});
    h = waitbar(i/numel(tiffFiles), h, sprintf('文件读取进
度: %d%%', round(i/numel(tiffFiles).*100)));
end
delete(h);

%% 信息读取模块
geoImages = cell(numel(tiffFiles), 1);
for i = 1:numel(tiffFiles)
    geoImages{i} = readgeoraster(tiffFiles{i}, 'OutputType', 'double');
    [A, R] = readgeoraster(tiffFiles{i});
    geoImages{i, 1} = A;
    geoImages{i, 2} = R;
end

%% 数据处理模块
% 无
numImages = numel(images);
for i = 1:numImages
    eval(['TIFFData', num2str(i), ' = images{i};']);
end

for i = 1:numImages
    currentImage_normal = eval(['TIFFData', num2str(i)]);
    currentImage_normal(currentImage_normal ~= 1) = 0;
    eval(['TIFFData_normal', num2str(i), ' = currentImage_normal;']);
end
```

```

end

% 轻度
numImages = numel(images);
for i = 1:numImages
    eval(['TIFFData', num2str(i), ' = images{i};']);
end

for i = 1:numImages
    currentImage_mile = eval(['TIFFData', num2str(i)]);
    currentImage_mile(currentImage_mile ~= 2) = 0;
    eval(['TIFFData_mile', num2str(i), ' = currentImage_mile;']);
end

% 中度
numImages = numel(images);
for i = 1:numImages
    eval(['TIFFData', num2str(i), ' = images{i};']);
end

for i = 1:numImages
    currentImage_moderate = eval(['TIFFData', num2str(i)]);
    currentImage_moderate(currentImage_moderate ~= 3) = 0;
    eval(['TIFFData_moderate', num2str(i), ' = currentImage_moderate;']);
end

% 重度
numImages = numel(images);
for i = 1:numImages
    eval(['TIFFData', num2str(i), ' = images{i};']);
end

for i = 1:numImages
    currentImage_serious = eval(['TIFFData', num2str(i)]);
    currentImage_serious(currentImage_serious ~= 4) = 0;
    eval(['TIFFData_serious', num2str(i), ' = currentImage_serious;']);
end

% 极端
numImages = numel(images);
for i = 1:numImages
    eval(['TIFFData', num2str(i), ' = images{i};']);
end

for i = 1:numImages
    currentImage_extreme = eval(['TIFFData', num2str(i)]);
    currentImage_extreme(currentImage_extreme ~= 5) = 0;

```

```

        eval(['TIFFData_extreme', num2str(i), ' = currentImage_extreme;']);
    end

%% 频数处理模块
% 无
sumData_normal = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage_normal = eval(['TIFFData_normal', num2str(i)]);
    sumData_normal = sumData_normal + currentImage_normal;
end
% 轻度
sumData_mile = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage_mile = eval(['TIFFData_mile', num2str(i)]);
    sumData_mile = sumData_mile + currentImage_mile;
end
sumData_mile_R = sumData_mile./2;
% 中度
sumData_moderate = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage_moderate = eval(['TIFFData_moderate', num2str(i)]);
    sumData_moderate = sumData_moderate + currentImage_moderate;
end
sumData_moderate_R = sumData_moderate./3;
% 重度
sumData_serious = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage_serious = eval(['TIFFData_serious', num2str(i)]);
    sumData_serious = sumData_serious + currentImage_serious;
end
sumData_serious_R = sumData_serious./4;
% 极端
sumData_extreme = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage_extreme = eval(['TIFFData_extreme', num2str(i)]);
    sumData_extreme = sumData_extreme + currentImage_extreme;
end
sumData_extreme_R = sumData_extreme./5;

%% 图像输出模块
% 无
figure;
imagesc(sumData_normal);
colorbar;

```

```

title('夏季高温干旱复合事件发生频数(无)');
xlabel('Columns');
ylabel('Rows');
% 轻度
figure;
imagesc(sumData_mile_R);
colorbar;
title('夏季高温干旱复合事件发生频数(轻度)');
xlabel('Columns');
ylabel('Rows');
% 中度
figure;
imagesc(sumData_moderate_R);
colorbar;
title('夏季高温干旱复合事件发生频数(中度)');
xlabel('Columns');
ylabel('Rows');
% 重度
figure;
imagesc(sumData_serious_R);
colorbar;
title('夏季高温干旱复合事件发生频数(重度)');
xlabel('Columns');
ylabel('Rows');
% 极端
figure;
imagesc(sumData_extreme_R);
colorbar;
title('夏季高温干旱复合事件发生频数(极端)');
xlabel('Columns');
ylabel('Rows');

%% TIFF 文件输出模块
% 正常
saveGeoTiff(sumData_normal, R, '保存高温干旱复合天气事件频数数据（正
常）为 TIFF 文件');
% 轻度
saveGeoTiff(sumData_mile_R, R, '保存高温干旱复合天气事件频数数据（轻
度）为 TIFF 文件');
% 中度
saveGeoTiff(sumData_moderate_R, R, '保存高温干旱复合天气事件频数数据
（中度）为 TIFF 文件');
% 重度

```

```

        saveGeoTiff(sumData_serious_R, R, '保存高温干旱复合天气事件频数数据（重
度）为 TIFF 文件');
        % 极端
        saveGeoTiff(sumData_extreme_R, R, '保存高温干旱复合天气事件频数数据（极
端）为 TIFF 文件');

    case '停止运行'
        disp('程序已停止运行')
        return
end

%% 函数模块：遍历文件夹中的 TIFF 文件
function files = findFilesRecursive(folder, fileExt)
    contents = dir(folder);
    numFiles = sum(~[contents.isdir]);
    files = cell(numFiles, 1);
    idx = 1;
    for i = 1:numel(contents)
        if strcmp(contents(i).name, '.') || strcmp(contents(i).name, '..')
            continue;
        end
        currentPath = fullfile(folder, contents(i).name);
        if contents(i).isdir
            nestedFiles = findFilesRecursive(currentPath, fileExt);
            files(idx:idx+numel(nestedFiles)-1) = nestedFiles;
            idx = idx + numel(nestedFiles);
        else
            [~, ~, ext] = fileparts(contents(i).name);
            if strcmpi(ext, fileExt)
                files{idx} = currentPath;
                idx = idx + 1;
            end
        end
    end
end

%% 函数模块：保存 TIFF 文件
function saveGeoTiff(data, R, dialogTitle)
    [output_filename, output_path] = uiputfile('*.tif', dialogTitle);
    if isequal(output_filename, 0)
        disp('保存操作被取消。');
    else
        output_tiff_path = fullfile(output_path, output_filename);
        geotiffwrite(output_tiff_path, data, R);
        disp(['文件已成功保存为 TIFF 文件: ', output_tiff_path]);
    end
end

```



```

        end
    end
    (5-2) MATLAB 计算代码 F 型:
    clc
    clear
    %% 提示模块
    choice = questdlg('请将 TIFF 文件保存在一个文件夹中，文件夹中不要有其它文件', '提示', '继续运行', '停止运行', '停止运行');
    switch choice
        case '继续运行'
            %% 交互模块
            folderPath = uigetdir('选择栅格文件所在的文件夹');
            if folderPath == 0
                disp('未选择任何文件夹。');
                return;
            end
            tiffFiles = findFilesRecursive(folderPath, '.tif');
            images = cell(numel(tiffFiles), 1);
            h = waitbar(0, '进行中', 'Name', '文件读取进度');
            for i = 1:numel(tiffFiles)
                images{i} = imread(tiffFiles{i});
                h = waitbar(i/numel(tiffFiles), h, sprintf('文件读取进
                度: %d%%', round(i/numel(tiffFiles).*100)));
            end
            delete(h);

            %% 信息读取模块
            geoImages = cell(numel(tiffFiles), 1);
            for i = 1:numel(tiffFiles)
                geoImages{i} = readgeoraster(tiffFiles{i}, 'OutputType', 'double');
                [A, R] = readgeoraster(tiffFiles{i});
                geoImages{i, 1} = A;
                geoImages{i, 2} = R;
            end

            %% 数据处理模块
            % 无
            numImages = numel(images);
            for i = 1:numImages
                eval(['TIFFData', num2str(i), ' = images{i};']);
            end

            for i = 1:numImages
                currentImage = eval(['TIFFData', num2str(i)]);
            end

```

```

        currentImage(currentImage ~= 1) = 0;
        eval(['TIFFData_normal', num2str(i), ' = currentImage;']);
    end

% 轻度
numImages = numel(images);
for i = 1:numImages
    eval(['TIFFData', num2str(i), ' = images{i};']);
end

for i = 1:numImages
    currentImage = eval(['TIFFData', num2str(i)]);
    currentImage(currentImage ~= 2) = 0;
    eval(['TIFFData_mile', num2str(i), ' = currentImage;']);
end

% 中度
numImages = numel(images);
for i = 1:numImages
    eval(['TIFFData', num2str(i), ' = images{i};']);
end

for i = 1:numImages
    currentImage = eval(['TIFFData', num2str(i)]);
    currentImage(currentImage ~= 3) = 0;
    eval(['TIFFData_moderate', num2str(i), ' = currentImage;']);
end

% 重度
numImages = numel(images);
for i = 1:numImages
    eval(['TIFFData', num2str(i), ' = images{i};']);
end

for i = 1:numImages
    currentImage = eval(['TIFFData', num2str(i)]);
    currentImage(currentImage ~= 4) = 0;
    eval(['TIFFData_serious', num2str(i), ' = currentImage;']);
end

% 极端
numImages = numel(images);
for i = 1:numImages
    eval(['TIFFData', num2str(i), ' = images{i};']);
end

for i = 1:numImages

```

```

        currentImage = eval(['TIFFData', num2str(i)]);
        currentImage(currentImage ~= 5) = 0;
        eval(['TIFFData_extreme', num2str(i), ' = currentImage;']);
    end

%% 频数处理模块
% 无
sumData_normal = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage = eval(['TIFFData_normal', num2str(i)]);
    sumData_normal = sumData_normal + currentImage;
end

% 轻度
sumData_mile = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage = eval(['TIFFData_mile', num2str(i)]);
    sumData_mile = sumData_mile + currentImage;
end
sumData_mile_R = sumData_mile./2;

% 中度
sumData_moderate = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage = eval(['TIFFData_moderate', num2str(i)]);
    sumData_moderate = sumData_moderate + currentImage;
end
sumData_moderate_R = sumData_moderate./3;

% 重度
sumData_serious = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage = eval(['TIFFData_serious', num2str(i)]);
    sumData_serious = sumData_serious + currentImage;
end
sumData_serious_R = sumData_serious./4;

% 极端
sumData_extreme = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage = eval(['TIFFData_extreme', num2str(i)]);
    sumData_extreme = sumData_extreme + currentImage;
end
sumData_extreme_R = sumData_extreme./5;

SumData_Union = sumData_mile_R + sumData_moderate_R + sumData_serious_R +
sumData_extreme_R;

```

```

%% 图像输出模块
% 无
figure;
imagesc(SumData_Union);
colorbar;
title('夏季高温干旱复合事件发生频数(全等级)');
xlabel('Columns');
ylabel('Rows');

%% TIFF 文件输出模块
% 全等级
saveGeoTiff(SumData_Union, R, '保存高温干旱复合天气事件频数数据（全等
级）为 TIFF 文件');

case '停止运行'
    disp('程序已停止运行')
    return
end
%% 函数模块：遍历文件夹中的 TIFF 文件
function files = findFilesRecursive(folder, fileExt)
    contents = dir(folder);
    numFiles = sum(~[contents.isdir]);
    files = cell(numFiles, 1);
    idx = 1;
    for i = 1:numel(contents)
        if strcmp(contents(i).name, '.') || strcmp(contents(i).name, '..')
            continue;
        end
        currentPath = fullfile(folder, contents(i).name);
        if contents(i).isdir
            nestedFiles = findFilesRecursive(currentPath, fileExt);
            files(idx:idx+numel(nestedFiles)-1) = nestedFiles;
            idx = idx + numel(nestedFiles);
        else
            [~, ~, ext] = fileparts(contents(i).name);
            if strcmpi(ext, fileExt)
                files{idx} = currentPath;
                idx = idx + 1;
            end
        end
    end
end
end

function saveGeoTiff(data, R, dialogTitle)

```

```

[output_filename, output_path] = uinputfile('*.tif', dialogTitle);
if isequal(output_filename, 0)
    disp('保存操作被取消。');
else
    output_tiff_path = fullfile(output_path, output_filename);
    geotiffwrite(output_tiff_path, data, R);
    disp(['文件已成功保存为 TIFF 文件: ', output_tiff_path]);
end
end

(5-3) MATLAB 频率计算代码:
clc
clear
%% 提示模块
choice = questdlg('请将 TIFF 文件保存在一个文件夹中, 文件夹中不要有其它文件', '提示', '继续运行', '停止运行', '停止运行');
switch choice
    case '继续运行'
        %% 交互模块
        folderPath = uigetdir('选择栅格文件所在的文件夹');
        if folderPath == 0
            disp('未选择任何文件夹。');
            return;
        end
        tiffFiles = findFilesRecursive(folderPath, '.tif');
        images = cell(numel(tiffFiles), 1);
        h = waitbar(0, '进行中', 'Name', '文件读取进度');
        for i = 1:numel(tiffFiles)
            images{i} = imread(tiffFiles{i});
            h = waitbar(i/numel(tiffFiles), h, sprintf('文件读取进
度: %d%%', round(i/numel(tiffFiles).*100)));
        end
        delete(h);

        %% 信息读取模块
        geoImages = cell(numel(tiffFiles), 1);
        for i = 1:numel(tiffFiles)
            geoImages{i} = readgeoraster(tiffFiles{i}, 'OutputType', 'double');
            [A, R] = readgeoraster(tiffFiles{i});
            geoImages{i, 1} = A;
            geoImages{i, 2} = R;
        end

        %% 数据处理模块
        % 无

```

```

numImages = numel(images);
for i = 1:numImages
    eval(['TIFFData', num2str(i), ' = images{i};']);
end

for i = 1:numImages
    currentImage_normal = eval(['TIFFData', num2str(i)]);
    currentImage_normal(currentImage_normal ~= 1) = 0;
    eval(['TIFFData_normal', num2str(i), ' = currentImage_normal;']);
end

% 轻度
numImages = numel(images);
for i = 1:numImages
    eval(['TIFFData', num2str(i), ' = images{i};']);
end

for i = 1:numImages
    currentImage_mile = eval(['TIFFData', num2str(i)]);
    currentImage_mile(currentImage_mile ~= 2) = 0;
    eval(['TIFFData_mile', num2str(i), ' = currentImage_mile;']);
end

% 中度
numImages = numel(images);
for i = 1:numImages
    eval(['TIFFData', num2str(i), ' = images{i};']);
end

for i = 1:numImages
    currentImage_moderate = eval(['TIFFData', num2str(i)]);
    currentImage_moderate(currentImage_moderate ~= 3) = 0;
    eval(['TIFFData_moderate', num2str(i), ' = currentImage_moderate;']);
end

% 重度
numImages = numel(images);
for i = 1:numImages
    eval(['TIFFData', num2str(i), ' = images{i};']);
end

for i = 1:numImages
    currentImage_serious = eval(['TIFFData', num2str(i)]);
    currentImage_serious(currentImage_serious ~= 4) = 0;
    eval(['TIFFData_serious', num2str(i), ' = currentImage_serious;']);
end

```

```

% 极端
numImages = numel(images);
for i = 1:numImages
    eval(['TIFFData', num2str(i), ' = images{i};']);
end

for i = 1:numImages
    currentImage_extreme = eval(['TIFFData', num2str(i)]);
    currentImage_extreme(currentImage_extreme ~= 5) = 0;
    eval(['TIFFData_extreme', num2str(i), ' = currentImage_extreme;']);
end

%% 频数处理模块
% 无
sumData_normal = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage_normal = eval(['TIFFData_normal', num2str(i)]);
    sumData_normal = sumData_normal + currentImage_normal;
end
% 轻度
sumData_mile = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage_mile = eval(['TIFFData_mile', num2str(i)]);
    sumData_mile = sumData_mile + currentImage_mile;
end
sumData_mile_R = sumData_mile./2;
% 中度
sumData_moderate = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage_moderate = eval(['TIFFData_moderate', num2str(i)]);
    sumData_moderate = sumData_moderate + currentImage_moderate;
end
sumData_moderate_R = sumData_moderate./3;
% 重度
sumData_serious = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage_serious = eval(['TIFFData_serious', num2str(i)]);
    sumData_serious = sumData_serious + currentImage_serious;
end
sumData_serious_R = sumData_serious./4;
% 极端
sumData_extreme = zeros(size(TIFFData1));
for i = 1:numImages
    currentImage_extreme = eval(['TIFFData_extreme', num2str(i)]);

```

```

    sumData_extreme = sumData_extreme + currentImage_extreme;
end
sumData_extreme_R = sumData_extreme./5;
% 全部等级
sumData_normal_All = sumData_normal .* 0;
SumData_Union = sumData_normal_All + sumData_mile_R + sumData_moderate_R +
sumData_serious_R + sumData_extreme_R;
%% 频率处理模块
rate_normal = sumData_normal ./ numImages;
rate_mile = sumData_mile_R ./ numImages;
rate_moderate = sumData_moderate_R ./ numImages;
rate_serious = sumData_serious_R ./ numImages;
rate_extreme = sumData_extreme_R ./ numImages;
rate_All = SumData_Union ./ numImages;
%% 图像输出模块
% 无
figure;
imagesc(rate_normal);
colorbar;
title('夏季高温干旱复合事件发生频率(无)');
xlabel('Columns');
ylabel('Rows');
% 轻度
figure;
imagesc(rate_mile);
colorbar;
title('夏季高温干旱复合事件发生频率(轻度)');
xlabel('Columns');
ylabel('Rows');
% 中度
figure;
imagesc(rate_moderate);
colorbar;
title('夏季高温干旱复合事件发生频率(中度)');
xlabel('Columns');
ylabel('Rows');
% 重度
figure;
imagesc(rate_serious);
colorbar;
title('夏季高温干旱复合事件发生频率(重度)');
xlabel('Columns');
ylabel('Rows');
% 极端

```



```

figure;
imagesc(rate_extreme);
colorbar;
title('夏季高温干旱复合事件发生频率(极端)');
xlabel('Columns');
ylabel('Rows');
% All
figure;
imagesc(rate_All);
colorbar;
title('夏季高温干旱复合事件发生频率(所有等级)');
xlabel('Columns');
ylabel('Rows');

%% TIFF 文件输出模块
% 正常
saveGeoTiff(rate_normal, R, '保存高温干旱复合天气事件频率数据（正常）为
TIFF 文件');
% 轻度
saveGeoTiff(rate_mild, R, '保存高温干旱复合天气事件频率数据（轻度）为
TIFF 文件');
% 中度
saveGeoTiff(rate_moderate, R, '保存高温干旱复合天气事件频率数据（中度）
为 TIFF 文件');
% 重度
saveGeoTiff(rate_serious, R, '保存高温干旱复合天气事件频率数据（重度）为
TIFF 文件');
% 极端
saveGeoTiff(rate_extreme, R, '保存高温干旱复合天气事件频率数据（极端）为
TIFF 文件');
% All
saveGeoTiff(rate_All, R, '保存高温干旱复合天气事件频率数据（所有等级）为
TIFF 文件');

case '停止运行'
    disp('程序已停止运行')
    return
end
%% 函数模块：遍历文件夹中的 TIFF 文件
function files = findFilesRecursive(folder, fileExt)
    contents = dir(folder);
    numFiles = sum(~[contents.isdir]);
    files = cell(numFiles, 1);
    idx = 1;

```

```

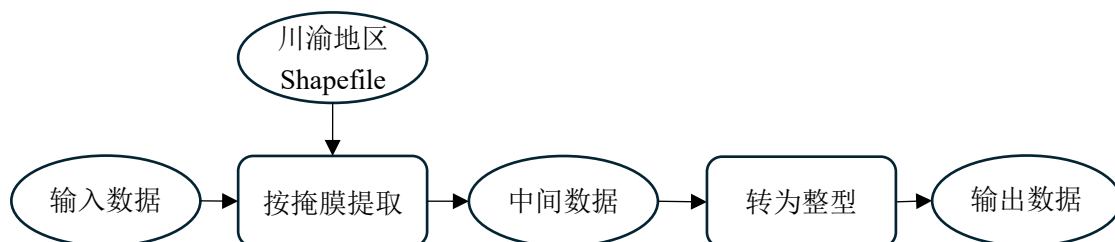
for i = 1:numel(contents)
    if strcmp(contents(i).name, '.') || strcmp(contents(i).name, '..')
        continue;
    end
    currentPath = fullfile(folder, contents(i).name);
    if contents(i).isdir
        nestedFiles = findFilesRecursive(currentPath, fileExt);
        files{idx:idx+numel(nestedFiles)-1} = nestedFiles;
        idx = idx + numel(nestedFiles);
    else
        [~, ~, ext] = fileparts(contents(i).name);
        if strcmpi(ext, fileExt)
            files{idx} = currentPath;
            idx = idx + 1;
        end
    end
end
end

%% 函数模块：保存 TIFF 文件
function saveGeoTiff(data, R, dialogTitle)
    [output_filename, output_path] = uinputfile('*.tif', dialogTitle);
    if isequal(output_filename, 0)
        disp('保存操作被取消。');
    else
        output_tiff_path = fullfile(output_path, output_filename);
        geotiffwrite(output_tiff_path, data, R);
        disp(['文件已成功保存为 TIFF 文件: ', output_tiff_path]);
    end
end

```

3.3 数据制图

(1-1) 简介：通过 MATLAB 代码得到需要的数据后就可以进行制图和出图工作了，由于 MATLAB 代码会将川渝地区之外的栅格数据设置为无效值（本次制图将无效值设置为 NaN），因此还需要对 MATLAB 代码输出的栅格文件进行一次按掩膜提取。通过 MATLAB 输出的栅格文件是不带属性表的，因此还需要将其转为整型，具体操作流程如下：



(1-2) 各工具参数设置:

1) 按掩膜提取:

输入栅格: 输入通过 MATLAB 输出的数据。

输入栅格数据或要素掩膜数据: 川渝地区 Shapefile。

输出栅格: 输入文件保存的路径和名称。(建议: 将文件保存在地理文件数据库中)

2) 转为整型:

输入栅格数据或常量值: 输入上一步输出的栅格图层。

输出栅格: 输入文件保存的路径和名称。(建议: 将文件保存在地理文件数据库中)

(2) 添加需要数据:

川渝地区市级数据和地级市点集数据, 由于下载来的地级市点集数据是全国数据, 还需使用裁剪工具对点集数据进行裁剪。

裁剪工具参数设置:

输入要素: 下载来的全国地级市点集数据。

裁剪要素: 川渝地区市级 Shapefile。

输出要素类: 输入保存路径和名称。

其余参数均可保持默认。

(3) 制图:

1) 切换到布局视图。

2) 创建经纬网: 点击菜单栏中视图→数据框属性, 弹出数据框属性窗口, 然后点击格网选项卡, 点击新建格网, 弹出格网和经纬网向导窗口。在“创建哪项?”中选择“经纬网: 用经线和纬线分割地图”, 点击下一步。在“外观”中选择“仅标注”, 在“间隔”中的放置纬线间隔和放置经线间隔均设置为 5 度, 点击下一步。在“轴”中取消勾选短轴主刻度, 点击下一步, 再点击下一步完成经纬网创建。将格网和地图调整为合适的大小。

3) 插入地图要素: 点击菜单栏中的插入, 这里可以插入许多地图要素, 比如标题、文本、图例、比例尺、比例文本等, 插入进地图后调整它们的位置和大小和颜色, 尽可能让地图更为美观。

(4) 出图:

点击文件→导出地图即可将地图导出为 (.png) 图片文件, 本次制图输出的地图的文件保存路径如表 2 所示, 参数描述如表 3 所示。

(5) 制作动画:

为了更好的分析高温干旱复合天气事件的时空变化规律和变化趋势, 还可以通过制作视频来展示图片。本次制图输出的视频文件保存路径如表 4 所示。

表 2 文件保存路径和描述

编号	文件路径	描述
1	···\Month	单月高温干旱复合天气事件等级分布图 (共 114 张图)
2	···\Summer	夏季高温干旱复合天气事件等级分布图 (共 38 张图)
3	···\Rate\Month	通过单月高温干旱复合天气事件等级数据统计的各等级发生的频率图 (共 7 张图)
4	···\Rate\Summer	通过夏季季节高温干旱复合天气事件等级数据统计的各等级发生的频率图 (共 7 张图)
5	···\Frequency\Month	通过单月高温干旱复合天气事件等级数据统计的各等级发生的频次图 (共 7 张图)
6	···\Frequency\Summer	通过夏季季节高温干旱复合天气事件等级数据统计的各等级发生的频次图 (共 7 张图)
7	···\Monthly\6\Frequency	通过 38 年的 6 月份的数据统计的各等级发生频次 (共 7 张图)
8	···\Monthly\6\Rate	通过 38 年的 6 月份的数据统计的各等级发生频率 (共 7 张图)
9	···\Monthly\7\Frequency	通过 38 年的 7 月份的数据统计的各等级发生频次 (共 5 张图)
10	···\Monthly\7\Rate	通过 38 年的 7 月份的数据统计的各等级发生频率 (共 5 张图)
11	···\Monthly\8\Frequency	通过 38 年的 8 月份的数据统计的各等级发生频次 (共 5 张图)
12	···\Monthly\8\Rate	通过 38 年的 8 月份的数据统计的各等级发生频率 (共 5 张图)

表 3 参数解读

编号	参数	描述
1	PT+xx0x	SPI 和 STI 取中间的字母组成，xx0x 表示 2000 年前的年份和月份，例如 8206 代表 1982 年 6 月
2	PT+20xx0x	20xx0x 表示 2000 年以后的年份和月份，例如 200006 代表 2000 年 6 月
3	PT+年份	代表某一年夏季的高温干旱复合天气事件分布图
4	Mile	轻度
5	Moderate	中度
6	Non_A	等级为无的反色图
7	Non_B	等级为无的正常图
8	Serious	重度
9	Serious_A	等级为重度的局部图
10	Union	除去等级为无的所有等级的高温干旱复合天气事件分布图

表 4 视频文件保存路径

编号	文件路径	描述
1	...\Videos\Month.mp4	通过单月高温干旱复合天气事件等级分布图（共 114 张图）制作的视频
2	...\Videos\Month6.mp4	通过 6 月高温干旱复合天气事件等级分布图（共 38 张图）制作的视频
3	...\Videos\Month7.mp4	通过 7 月高温干旱复合天气事件等级分布图（共 38 张图）制作的视频
4	...\Videos\Month8.mp4	通过 8 月高温干旱复合天气事件等级分布图（共 38 张图）制作的视频
5	...\Videos\Summer.mp4	通过夏季高温干旱复合天气事件等级分布图（共 38 张图）制作的视频

文献和数据引用

- [1]孙昭萱,张强,孙蕊,等.2022 年西南地区极端高温干旱特征及其主要影响[J].干旱气象,2022,40(05):764-770.
- [2]Peng, S.Z., Ding, Y.X., Wen, Z.M., Chen, Y.M., Cao, Y., & Ren, J.Y. (2017). Spatiotemporal change and trend analysis of potential evapotranspiration over the Loess Plateau of China during 2011-2100. *Agricultural and Forest Meteorology*, 233, 183-194. <https://doi.org/10.1016/j.agrformet.2016.11.129>
- [3]Ding, Y.X., & Peng, S.Z. (2020). Spatiotemporal trends and attribution of drought across China from 1901–2100. *Sustainability*, 12(2), 477.
- [4]Peng, S.Z., Ding, Y.X., Liu, W.Z., & Li, Z. (2019). 1 km monthly temperature and precipitation dataset for China from 1901 to 2017. *Earth System Science Data*, 11, 1931–1946. <https://doi.org/10.5194/essd-11-1931-2019>
- [5]Peng, S., Gang, C., Cao, Y., & Chen, Y. (2017). Assessment of climate change trends over the loess plateau in China from 1901 to 2100. *International Journal of Climatology*.
- [6]彭守璋. (2019). 中国 1km 分辨率逐月平均气温数据集（1901-2022）. 国家青藏高原数据中心.
Peng, S. (2019). 1-km monthly mean temperature dataset for China (1901-2022). National Tibetan Plateau / Third Pole Environment Data Center.
<https://doi.org/10.11888/Meteoro.tpdc.270961><https://cstr.cn/18406.11.Meteoro.tpdc.270961>
- [7]彭守璋. (2020). 中国 1km 分辨率逐月降水量数据集（1901-2022）. 国家青藏高原数据中心.
Peng, S. (2020). 1-km monthly precipitation dataset for China (1901-2022). National Tibetan Plateau / Third Pole Environment Data Center.
<https://doi.org/10.5281/zenodo.3185722>
- [8]https://datav.aliyun.com/portal/school/atlas/area_selector