**目录**

**[一、预设置 2](#_Toc30835)**

[1） 文件夹设置 2](#_Toc28124)

[2） 库下载 2](#_Toc12111)

**[二、全部功能 2](#_Toc29216)**

[1. 插值 2](#_Toc4002)

[2. 下载 4](#_Toc21221)

[3. 拼接 4](#_Toc3250)

**[三、结果预览 5](#_Toc4030)**

[1.下载结果 5](#_Toc951)

[2. 插值结果 5](#_Toc17698)

[3. 拼接结果 5](#_Toc10954)

[4. 站点数据 7](#_Toc13896)

**[四、 如果结果精度较低可能发生错误的地方 7](#_Toc28638)**

[1. 局部加权插值函数 7](#_Toc15005)

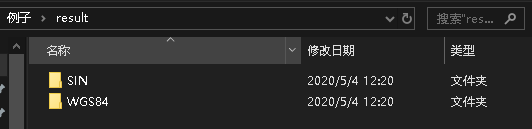
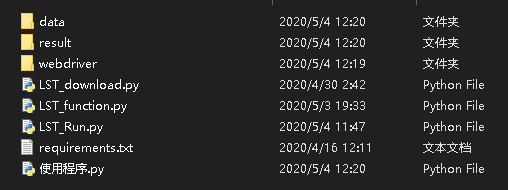
[2. 投影转换 7](#_Toc4298)

**[五、 一些改变的方法 7](#_Toc21561)**

**[六、 一些对使用这个程序有帮助的资料 9](#_Toc15480)**

**一、预设置**

1. 文件夹设置



1. 库下载

进入cmd命令界面

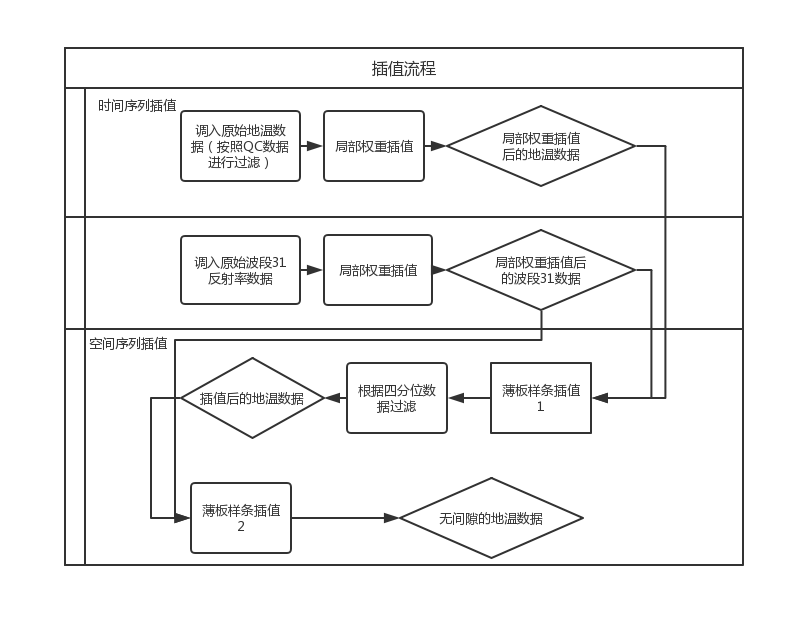
-pip install requirements.txt

这个方法成功几率很低，推荐使用编译器自带的pip插件，如果失败，可以登陆网站[https://www.lfd.uci.edu/~gohlke/pythonlibs/](https://www.lfd.uci.edu/~gohlke/pythonlibs/" \t "http://www.yanglajiao.com/article/GISMiaoYing/_blank)下载wheel文件进行安装

**二、全部功能**

1. **插值**

1）原理



2）使用方法

1）只生成无间隙的日地温平均值数据（正弦投影+WGS84）

from LST\_Run import Run  
  
date = ['2019-01-01','2019-01-03']  
block = ['h24v05']  
dir\_data = r'data'  
filv = 0  
espg = 4326  
  
tryOne = Run(date,block,dir\_data,filv,espg)  
tryOne.main()

2）生成无间隙的日地温平均值数据（正弦投影+WGS84）和站点数据

from LST\_Run import Run\_AC  
  
date = ['2019-01-01','2019-01-03']  
block = ['h24v05']  
dir\_data = r'data'  
filv = 0  
site\_data = [[52602,38.45,93.20],[52787,37.12,102.52]]  
csv\_name = 'Tibet\_2019-01-01~2019-01-03'  
espg = 4326  
  
tryOne = Run\_AC(date,block,dir\_data,filv,site\_data,csv\_name,espg)  
tryOne.main()

1. **下载**
2. **使用方法**
3. **方法一:无预操作（window可行，其他系统请下载2.7.1版本chrome到webdriver文件夹，名称按照文件夹内对应的名称设置）**

from LST\_download import download1  
  
dir\_out = 'data'  
username = 'XXXX'  
password = 'XXXXX'  
time = ['2019-01-01','2019-01-02']  
location = ['70.0','30.0','90.0','20.0']  
product = 'MOD11A1--6'  
  
tryOne = download1(dir\_out,username,password,time,location,product)  
tryOne.main()

1. **方法二：需要登陆earthdata网站，下载所需数据的url，生成txt**

from LST\_download import download2  
  
urlfile = 'data/urlfile.txt'  
dir\_out = 'data'  
username = 'XXXXX'  
password = 'XXXXXX'  
  
tryOne = download2(urlfile,dir\_out,username,password)  
tryOne.main()

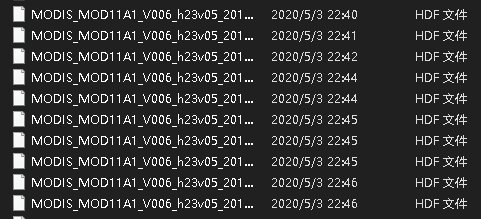
1. **拼接**
2. **使用方法（录入数据必须是正弦投影数据，因为这样拼接过程不会产生错误）**

**（生成拼接后的影像（正弦投影+WGS84））**

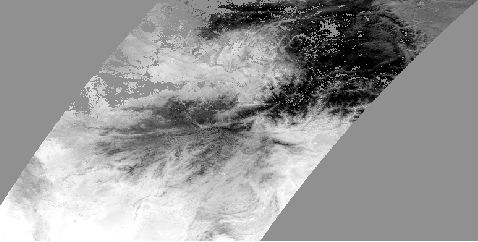
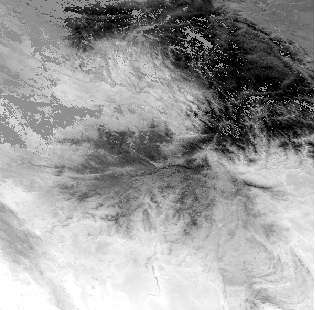
from LST\_Run import Splicing  
  
date = ['2019-01-01','2019-01-01']  
block = ['h23v05','h24v05']  
dir\_data = r'result\SIN'  
dir\_out = r'result'  
filename = 'Tibet'  
Nodata = 0  
  
tryOne = Splicing(date,block,dir\_data,dir\_out,filename,Nodata)  
tryOne.main()

**三、结果预览**

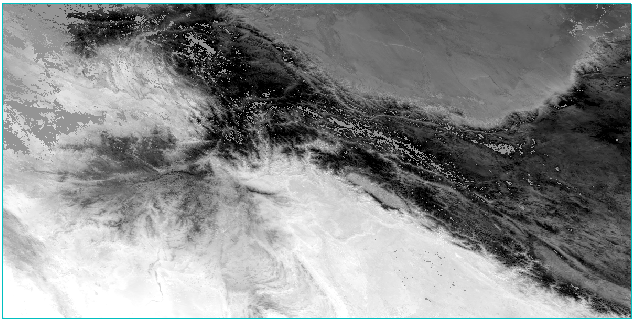
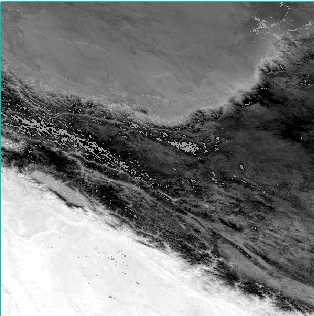
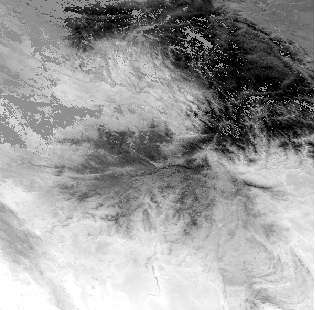
**1.下载结果**

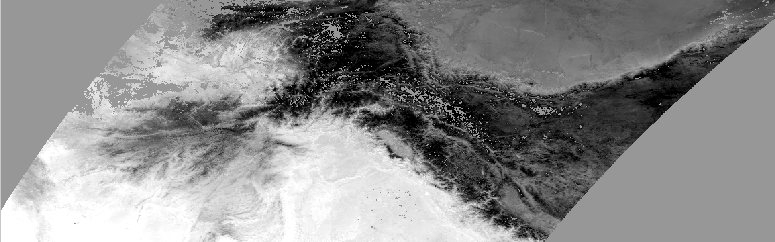
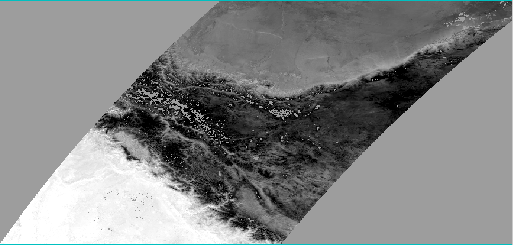
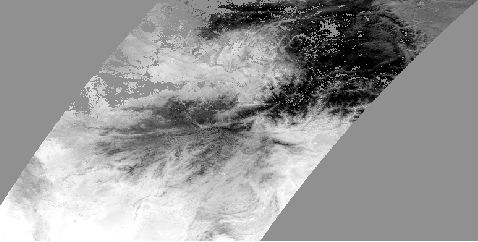
****

1. **插值结果**

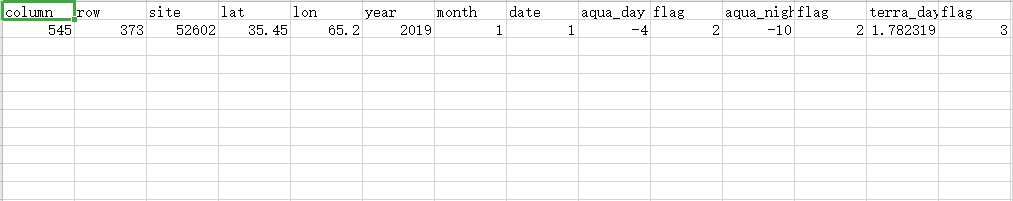
****

1. **拼接结果**

****

****

1. **站点数据**

****

1. **如果结果精度较低可能发生错误的地方**
2. **局部加权插值函数**

**程序中局部加权插值的思路如下**

def lwlr(testPoint,xArr,yArr,k=1.0):  
 xMat = np.mat(xArr); yMat = np.mat(yArr).T  
 m = np.shape(xMat)[0]  
 weights = np.mat(np.eye((m)))  
 for j in range(m): #next 2 lines create weights matrix  
 diffMat = testPoint - xMat[j,:] #difference matrix  
 weights[j,j] = np.exp(diffMat\*diffMat.T/(-2.0\*k\*\*2)) #weighted matrix  
 xTx = xMat.T \* (weights \* xMat)  
 if np.linalg.det(xTx) == 0.0:  
 print ("This matrix is singular, cannot do inverse")  
 return  
 ws = xTx.I \* (xMat.T \* (weights \* yMat)) #normal equation  
 return testPoint \* ws

1. **投影转换**

**程序中modis影像的投影信息,全部来自于已经预处理过的GMTED2010文件的投影,该文件的投影是通过envi软件根据h24v05影像中的参数设置的,也许不同景的投影有小的不同，如果是这个问题,解决方法是想办法读出modis影像自身的投影信息,代替从GMTED中读取**

**还有一个可能就是程序中所有影像都转换成espg4326投影,如果想要把不同块的数据转换成不同zone的WGS84,可以建立一个块编号与对应投影espg码的字典**

1. **一些改变的方法**
2. 原程序不能保证结果是无间隙的,关键在BAND31数据的插值只使用了7个最近邻,如果要得到完全无间隙的结果,建议使用30个最近邻,但是这样处理速度的减缓和结果的收益增加比很不理想,修改的方法是：(同时修改)

1)LST\_function 第44行修改为（-15,16）

def getFile(self,year,month,date):  
 #获取距插值日期最近的7天数据名称  
 list\_sort = []  
 filename = '{}-{}-{}'.format(year,month,date)  
 d = datetime.datetime.strptime(filename, '%Y-%m-%d')  
 for i in range(-3,4): #44行  
 delta = datetime.timedelta(days=i)  
 list\_sort.append(str(d + delta)[:10])  
  
 return list\_sort

def getFile(self,year,month,date):  
 #获取距插值日期最近的7天数据名称  
 list\_sort = []  
 filename = '{}-{}-{}'.format(year,month,date)  
 d = datetime.datetime.strptime(filename, '%Y-%m-%d')  
 for i in range(-15,16): #44行  
 delta = datetime.timedelta(days=i)  
 list\_sort.append(str(d + delta)[:10])  
  
 return list\_sort

1. LST\_Run 第148、290行（相同代码）改为[14:19]

for i in File\_list[1:6]: #148、290行  
 LST\_data = processor.getData('MODIS\_{}\_V006\_{}\_{}.hdf'.format(satellite,block,i), 'LST', DayNightFlag=daynight)  
 QC\_data = processor.getData('MODIS\_{}\_V006\_{}\_{}.hdf'.format(satellite,block,i), 'QC', DayNightFlag=daynight)  
 data = processor.QualityControl(LST\_data, QC\_data)  
 tem\_LstData.append(data)

=======================================================================

for i in File\_list[14:19]: #148、290行  
 LST\_data = processor.getData('MODIS\_{}\_V006\_{}\_{}.hdf'.format(satellite,block,i), 'LST', DayNightFlag=daynight)  
 QC\_data = processor.getData('MODIS\_{}\_V006\_{}\_{}.hdf'.format(satellite,block,i), 'QC', DayNightFlag=daynight)  
 data = processor.QualityControl(LST\_data, QC\_data)  
 tem\_LstData.append(data)

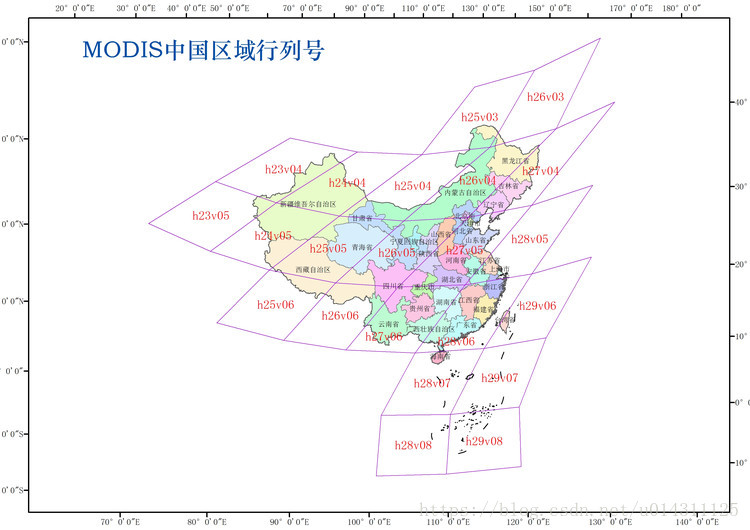
1. LST\_Run 第127、269行（相同代码）改为[[1,1],.........,[1,31]]

INTLWR1 = [[1, 1], [1, 2], [1, 3], [1, 4], [1, 5]]  
INTLWR2 = [[1, 1], [1, 2], [1, 3], [1, 4], [1, 5], [1, 6], [1, 7]]#127、269行

1. LST\_Run 第167、309行（相同代码）改为[1,16]

BAND31\_lwrData = INT.lwr([1, 4], INTLWR2, tem\_Band31Data) # 167、309行  
BAND31\_lwrData = np.array(BAND31\_lwrData).reshape(1200, 1200).T.tolist()

BAND31\_lwrData = INT.lwr([1, 16], INTLWR2, tem\_Band31Data) # 167、309行  
BAND31\_lwrData = np.array(BAND31\_lwrData).reshape(1200, 1200).T.tolist()

1. **一些对使用这个程序有帮助的资料**
2. 

H24V05某日的交叉验证结果

空间序列插值（现在的程序经过了优化）

TPS模拟点个数为：500

TPS异常值点数目为：4115

平均误差值:5.594751206665908

时间序列插值

最大值:16.149999999999995

最小值：0.0

平均误差值：0.8028812649781175