

# CYGNSS 反射测量估计海面风速

星载全球导航卫星系统反射测量(GNSS-R)是一种费效比极高的对地遥感观测技术,通过搭载在低轨卫星(LEO)上的时延/多普勒映射设备(DDMI)接收海面散射的 GNSS 导航定位信号并同接收机本地复制码(或者直射信号)进行相关处理,获得海面时延多普勒图(DDM),采用理论模型匹配或者提取 DDM 海洋要素指示计建立统计的量化模型可以反演全球尺度的高时空分辨率的海洋表面风速信息,同时对 DDM 观测过程中的信噪比(SNR)信息进行建模,可同时反演海面的有效波高(SWH)信息。

## 1.引言

海洋风场在天气和气候研究、洋流、海-气相互作用及海洋航运安全中均有非常重要的作用。星载 GNSS-R 海洋遥感技术利用 GNSS 全球导航卫星系统发射的导航定位信号作为双基地(多基地)雷达发射机,利用 LEO 卫星上搭载的 DDMI 遥感海洋表面的前向散射功率 DDM,散射功率主要受海洋表面风速(粗糙度)影响,通过双基地雷达理论模型、观测几何、观测系统的硬件及噪声信息建模模拟或者通过最优的风速指示计量化风速和指示计间的映射关系建立类似后向雷达散射计的 GMF 函数,就可以进行海洋风速和 SWH 的反演。GNSS-R 海洋遥感技术使用空间中已经普遍存在的 GNSS 导航定位信号,所以不需要另行设计遥感设备发射端系统,DDMI 的核心是时延多普勒映射接收机,用于接收 GNSS 直射信号和地球表面反射的 GNSS 信号,通过在不同时延和多普勒频移区间内对反射信号和直射信号,或者接收机本地复制码进行互相关处理。

天基 GNSS-R 的工作模式显著降低了遥感海洋风场的部署及业务化的成本。同时 GNSS 全球导航卫星系统使用 L 波段信号,其不受云雨的影响可以实现全天候作业。因此在对地遥感领域拥有非常广阔的应用前景,近年来引起了科研和商业领域的广泛关注。目前通过 DDM 进行星载 GNSS-R 海洋风速遥感的方法主要包括两类,一种基于利用 GMF 函数,量化观测量和风速的映射关系建立经验的 GMF 模型,建立模型过程中观测量对应的风速信息通过内插并置的其他观测系统获得风速。另一种方法主要通过优化双基地雷达方程,根据实际观测 DDM 过程中双基地遥感系统的几何配置、发射功率、天线增益、系统噪声和风速(海

面状态)等信息模拟建立理论的 DDM 或者时延波形,通过不断的迭代匹配观测量和理论量进行参数反演。由于后者对系统本身标定信息及计算量要求极高,本研究主要采用第一种方法进行 DDM 后处理风速反演,利用 DDM 随着不同的风速和海面粗糙度的变化,通过 DDM 派生的风速指示计,包括镜面反射点附近的归一化双基地雷达散射面积(NBRCS)和前缘坡度(LES)等对风场变化最敏感同时受观测噪声影响最弱的区域同其他观测技术获取的地表真实风速进行时空匹配建立 GMF 学习模型,学习模型总会受到残余观测噪声和训练数据量不足的影响出现异常抖动,算法实现中通过合理的平滑和约束条件可以有效剔除模型异常的同时保证模型的精度。最后采用最小方差估计器获得不同观测量对应反演风速的权系数,建立起了一套完整的星载 GNSS-R 风速反演程序。

## 2.算法说明

星载 GNSS-R 技术的直接观测量为 DDM,实际的风速反演算法是通过提取 DDM 中对风速敏感的观测量匹配其他风速观测系统的对应风速回归获得经验映射关系 GMF。在建立经验模型之前,首先需要根据双基地雷达方程对原始的 DDM 进行标定,去除非地球物理参数效应的影响,通常将其标定为雷达散射截面,在部分空基或者天基中,如 UK TDS-1 项目,当系统标定信息还不完善的情况下也可以直接使用 DDM 的 SNR 观测量,根据反演空间分辨率的要求提取 DDM 中受观测噪声影响小且对反演物理量敏感的 DDM 派生量建立反演模型。反演风速除了受到 DDM 时延和多普勒分辨率的影响外,还受到双基地系统几何配置的影响。为了精化模型,实际建立的 GMF 实际上是一个函数族,即我们在不同的高度角下分别建立一组 GMF,在风速预测阶段根据观测 DDM 的几何配置选择合适的 GMF,或者对齐其进行内插再将观测量映射到风速。

观测数据会受到不同源噪声的影响,通过对 DDM 观测量的质量控制,也可以进行时间平均处理,在满足反演空间分辨率的前提下,继续对 DDM 进行非相干求和提高 DDM 的信噪比,匹配多源地表参考风速建立二维的经验 GMF,为了剔除观测噪声和训练数据不足引起的模型异常,再通过分段非线性函数进行平滑。在平滑过程中添加多项约束条件以保证在模型平滑处理中可以有效剔除模型异常的同时保证模型的精度,包括风速小于分段点的离散点进行非线性最小二乘拟合,并要求函数的所有系数均大于零;对风速大于分段点的离散点,抛物线函数需要满足两个函数在分段点处函数值相等,一阶导数值相等,抛物线函数开口向下,对称轴位于分段点左侧。最后通过最小方差估计不同观测量映射风速的加

权系数。

星载 GNSS-R 风速反演过程如图 1 包括：(1) 计算 DDM 观测量；(2) 在满足反演空间分辨率要求的情况下通过时间平均(TA)提高观测量的信噪比；(3) DDM 观测量匹配地表真值风速建立训练数据集；(4) 根据 DDM 观测几何关系建立风速和 DDM 观测量、入射角的二维 GMF；(5) 平滑因噪声和训练数据不足引起的 GMF 抖动；(6) 通过最小方差估计不同观测量风速权系数；(7) 应用 GMF 进行风速反演。其中前六步用于建立反演模型，最后一步是通过模型进行风速的预测。

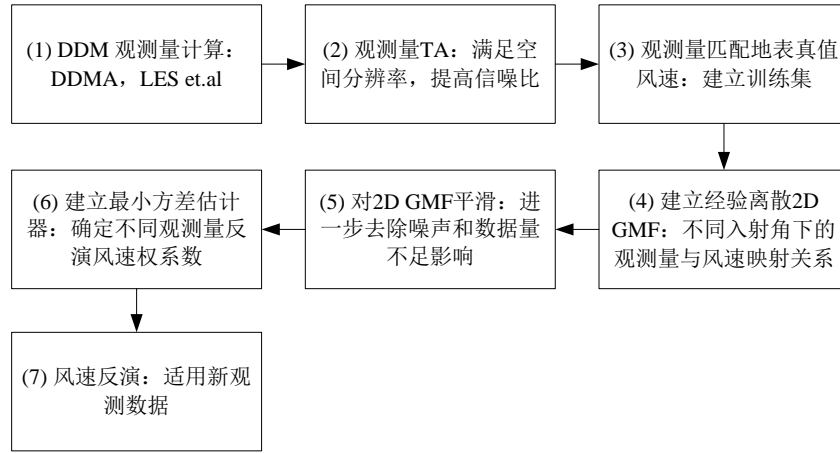


图 1 天基 GNSS-R 风速反演流程图

### 3.设计说明

#### 3.1 GNSS-R 数据预处理

当前主要的星载 GNSS-R 项目包括 UK TDS-1 和 NASA 的 CYGNSS，两者均提供了大量观测自海洋表面的 DDM 数据，可以通过标定的 DDM 计算两个风速指示计 DDMA 和 LES 反演风速。通过 CYGNSS Level1 数据进行海风反演，首先计算 DDM 对应的距离改正增益(RCG),用以指示观测量的质量，剔除掉 RCG 小于 1 的观测量，RCG 计算入下式：

$$RCG = \frac{G_r^{sp}}{(R_t^{sp} R_r^{sp})^2} 10^{27}$$

式中， $G_r^{sp}$ 表示镜面点的接收机天线增益，单位 dB。 $R_t^{sp}$ ， $R_r^{sp}$ 分别表示发射机到镜面的距离和镜面点至接收机的距离。DDM 对应的空间分辨率受到双基地雷几何结构的影响，在满足地表反演风速的空间分辨率达到 25km 指标时，可进一步

对 DDMA 和 LES 进行时间平均，提高观测量的信噪比。以镜面点入射角为自变量，在不同的入射角区间对观测量前后连续历元的采样求平均，时间平均规则如下：

表 1 样本时间平均策略

时间平均样本数量	入射角区间 (°)
5	$0^{\circ} < \theta \leq 17^{\circ}$
4	$17^{\circ} < \theta \leq 31^{\circ}$
3	$31^{\circ} < \theta \leq 41^{\circ}$
2	$41^{\circ} < \theta \leq 48^{\circ}$
1	$> 48^{\circ}$

3.2 地表真值风匹配

参考风速为其他观测技术或者数值天气预报模型同化输出的风速产品，如星载散射计，海洋锚定浮标观测数据，ECMWF ERA5 地表风速产品，GDAS 数值天气预报风速数据，CCMP 风速产品。部分数据提供单点不同时刻的风速数据，如锚定浮标。数值天气预报则提供格网化的风速产品。匹配不同源数据集作为 GNSS-R 地表真值风速，需要顾及不同数据源数据的时空分辨率进行时空插值。对离散参考风速数据需要设计巧妙的数据结构便于在时间上进行线性插值，并考虑参考数据中断时不再进行外推。对格网参考风速数据在空间上进行双线性内插，时间上进行线性内插。

3.3 建立 GMF

反演模型所建立的 GMF 实际上是一个函数簇，即在不同的入射角下分别建立一组 DDMA、LES 与风速的映射关系，以一定步长风速及对应观测量的散点表示。在建立 GMF 时，训练数据入射角范围取  $0^{\circ}$  到  $68^{\circ}$ ，风速范围取  $0\text{ m/s}$  到  $35\text{m/s}$ 。在入射角维上，从  $0.5^{\circ}$  开始以  $1^{\circ}$  为步长在不同的入射角下建立 GMF。在某一入射角下，再以  $0.1\text{m/s}$  的步长由  $0.05\text{m/s}$  风速起始至  $35\text{m/s}$ ，计算不同风速下的加权平均观测量，建立离散的经验映射关系。为了扩充训练集，允许建立 GMF 时训练数据可以相互重叠。在某一入射角下，取左右两个步长(步长为  $1^{\circ}$ )区间内的所有数据作为该入射角下的模型训练数据。在某一风速下，同样取左右两个步长区间内的数据计算加权 DDM 观测量，需要注意的是此时风速区间步长需要根据风速概率密度重新确定步长的大小，风速步长的选取策略如表 2。

表 2 风速步长选取策略

步长 (m/s)	风速区间 (m/s)
0.4	$U_{10} < 2$
0.3	$2 < U_{10} < 5$
0.2	$5 < U_{10} < 9$
0.4	$9 < U_{10} < 11$
0.6	$11 < U_{10} < 14$
0.8	$14 < U_{10} < 17$
1.0	$17 < U_{10}$

在入射角维和风速维，不同的区间段使用不同的权值，距离某一入射角/风速一倍步长区间的观测值取两倍的权值，不同入射角和风速下经验 GMF 散点的加权策略如图 3。

	$\frac{2}{14}$	$\frac{4}{14}$	$\frac{4}{14}$	$\frac{2}{14}$
Incidence	$\frac{4}{14}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{4}{14}$
	$\frac{4}{14}$	$\frac{8}{14}$	$\frac{8}{14}$	$\frac{4}{14}$
	$\frac{1}{14}$	$\frac{4}{14}$	$\frac{4}{14}$	$\frac{2}{14}$
	Wind speed			

图 3 不同区间观测值加权策略

为了保证经验模型的准确性，首先计算各个入射角下训练数据风速分布的最大概率密度区间。为了去除因训练数据量不足和噪声影响引起的模型抖动，由最大风速概率密度区间对应的风速开始计算对应的加权 DDM 观测量。再以 0.1m/s 的步长依次计算大于和小于该风速对应的观测量，并强制离散 GMF 为单调函数。再通过分段非线性函数对经验 GMF 进行平滑。风速小于分段点和大于分段点的平滑函数如(2)，(3)式：

$$\text{obs} = a_0 + a_1 u^{-1} + a_2 u^{-2} \quad (2)$$

$$\text{obs} = b_0 + b_1 u + b_2 u^2 \quad (3)$$

式中，obs 表示 DDM 观测量，u 表示真实风速。主要问题是要确定合理的分段点，保证模型最优拟合的同时在过度点处平滑。首先在不同的离散点上首先进行平滑，寻找到拟合残差最小的离散点，然后在离散点周围设置搜索区间，以更小的步长再次寻找拟合残差最小的点作为最终的过渡点。

### 3.4 最小方差估计

通过 MV 估计方法动态调整因观测几何变化引起的 DDM 信噪比变化，即在

不同的 RCG 区间，加权不同 DDMA 和 LES 映射的风速获得最优风速估计量。RCG 可先选取较小的区间根据估计结果迭代调整。

$$\mathbf{m} = \left( \sum_{i=1}^N \sum_{j=1}^N c_{i,j}^{-1} \right)^{-1} \mathbf{C}^{-1} \mathbf{I} \quad (4)$$

$$\sigma_{MV}^2 = \left( \sum_{i=1}^N \sum_{j=1}^N c_{i,j}^{-1} \right)^{-1} \quad (5)$$

$$\mathbf{C} = \mathbf{S} \mathbf{R} \mathbf{S} \quad (6)$$

$$\mathbf{U}_{10} = \mathbf{m} \cdot \mathbf{u} \quad (7)$$

最小方差估计器是确定不同观测量映射风速  $\mathbf{u}$  的最优权系数  $\mathbf{m}$ ，(6)式中  $\mathbf{S}$  为对角阵，对角元素为观测量反演误差的方差， $\mathbf{R}$  为观测量相关系数构成的矩阵。(4)式中  $\mathbf{I}$  为单位向量， $c_{i,j}^{-1}$  为  $\mathbf{C}$  逆阵元素。

## 4.程序说明

该程序基于 Python3.7 环境搭建，除主反演程序之外，还提供辅助的数据下载函数反演结果评定程序。项目组织结构如图 4：

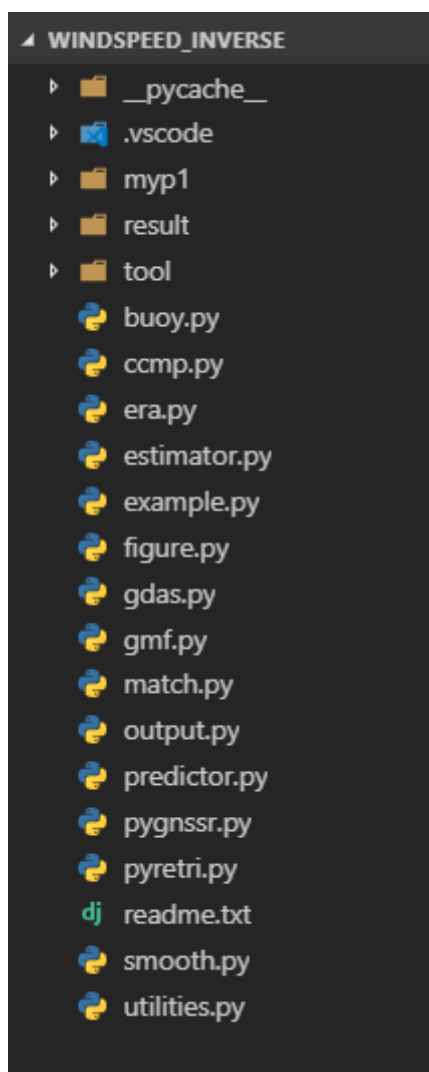


图 4 反演程序项目结构

项目所需要的 python 标准库包括: numpy, pandas, scipy, cvxopt, skilearn, pickle, julian, collections, netCDF4, h5py, Basemap, matplotlib。myp1 文件目录用于存储中间过程文件, result 文件目录保存程序输出结果文件, tool 文件目录保存 CYGNSS Level1 数据下载程序, 提供 GUI 界面, 如图 5。

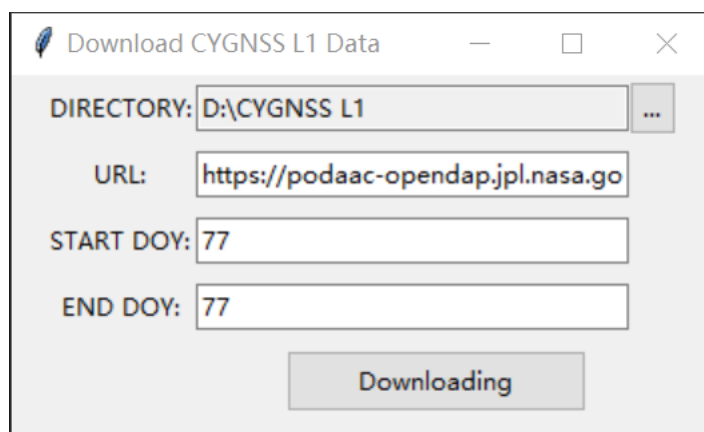


图 5 辅助数据下载工具

## 4.1 程序 GUI 设计

程序 GUI 界面通过 Qt5 设计实现，主要包括三部分：参数设置、数据输入、处理过程及结果展示，反演程序主界面如图 6。

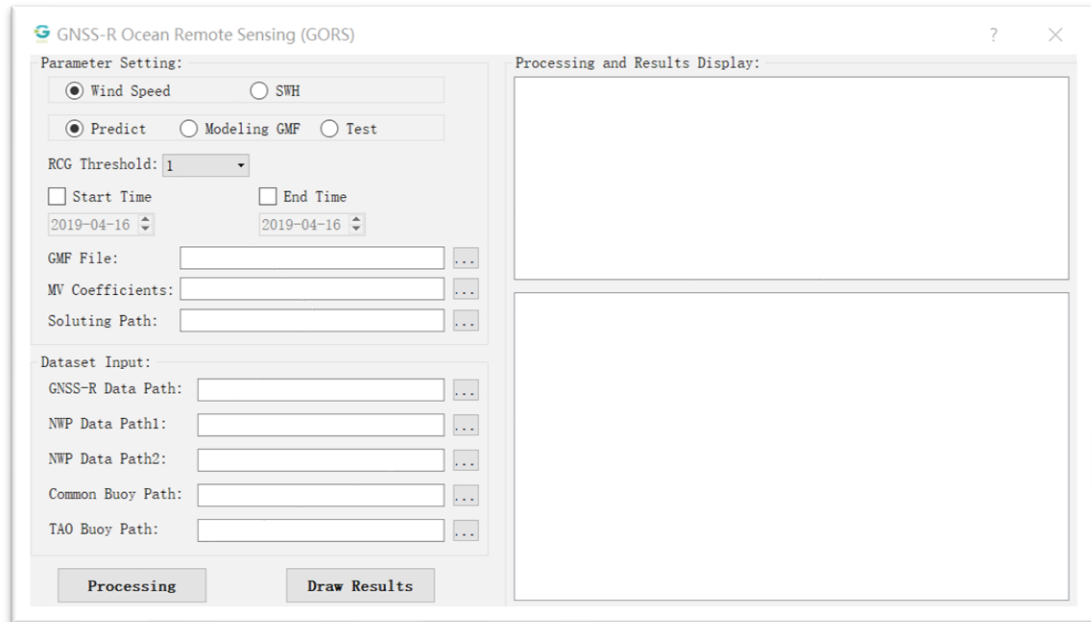


图 6 程序主界面

当前算法及程序均已实现独立的风速和有效波高的反演，在参数设置界面反演对象包括风速和有效波高两种，处理模式则分为三类：建模、模型检验、预测。其中建模是采用第三部分介绍的模型实现过程建立 GMF 函数簇和最小方差估计器。预测模式采用已经建立的 GMF 函数和最小方差估计器对新观测数据进行预测。模型检验是对已经建立的模型进行精度评估，此时同样需要外部的参考风速数据。有效波高反演类似，也分为建模、模型检验和预测三类处理模式。在风速反演中通过 RCG 区分不同的观测数据质量，程序可以选择不同 RCG 控制模型及预测数据的质量。截面单独实现了在数据池中选择指定的时间段进行建模、模型检验和预测。如果不指定起始时间和结束时间，则会对数据输入路径下的所有数据进行处理。GMF 文件、最小方差文件输入在预测和模型检验模式下使用，输入已经建立的 GMF 模型和最小方差估计系数对新数据进行预测和模型检验。结果路径用于存储所有输出结果。选定结果路径后程序会在该路径下新建立两个文件夹 myp1 和 result，用于存储中间过程文件和结果文件。数据输入路径指原始的 L1 GNSS-R 观测文件路径，nwp1 路径为 ECMWF(GMF 或 CCMP)提供的参考风速数据路径，nwp2 为另外一种的数值天气预报参考数据，如果两者均提供的情况下，会对输入参考风速进行相互检校，对参考风速内插获得的两组结



果大于 3m/s 的观测数据进行剔除，需要注意的是所有这些参考文件均需要使用标准的 “.nc” 文件，建议直接使用 ECMWF 和 ADS 提供的标准格式文件。浮标数据主要用于模型的检验，原因是将浮标数据作为训练数据时，训练样本太了。由于 TAO 浮标阵列提供的数据格式和普通的浮标文件稍有差异，因此在数据输入阶段需要加以区分。在处理过程和结果展示部分，上部文本框提示处理过程中的参数设置信息，正在进行的数据处理步骤等。下部的视图框展示反演模型和统计反演结果。绘制结果按钮在模型检验模式下会提供反演结果和参考风速的散点密度图。

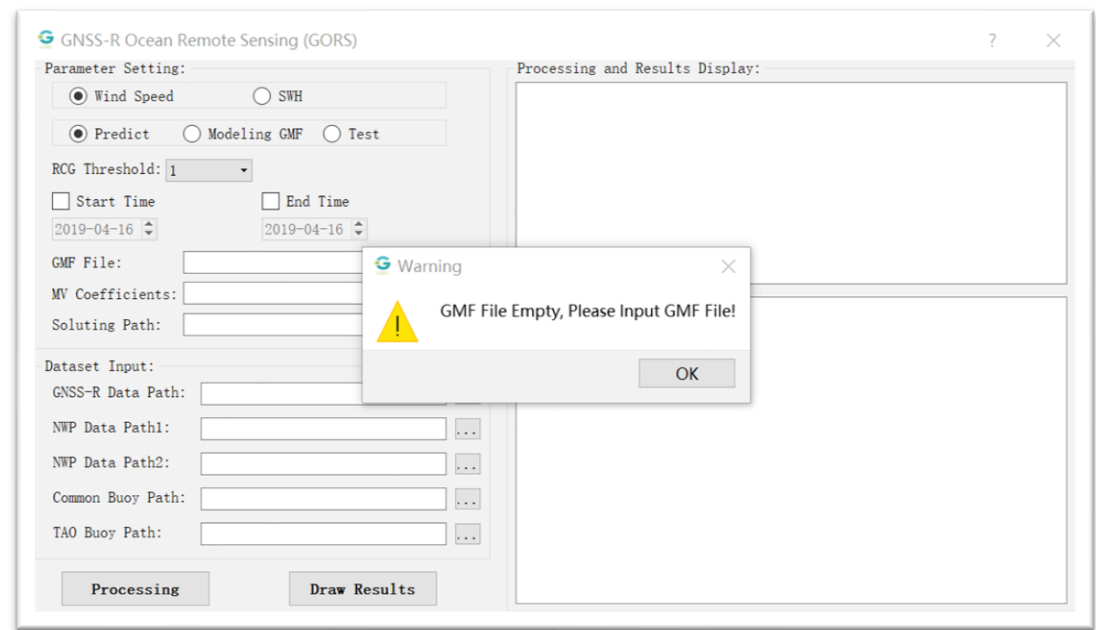


图 7 交互提示

为帮助用户更好的使用本程序，软件设计过程中也顾及到了交互提示的功能，在不同的模式下帮助用户检查输入项信息，以便更快的进行业务应用如图 7。在反演风速模式下程序运行的状态如图 8，其中视图窗口展示的是 2 维 GMF 模型。

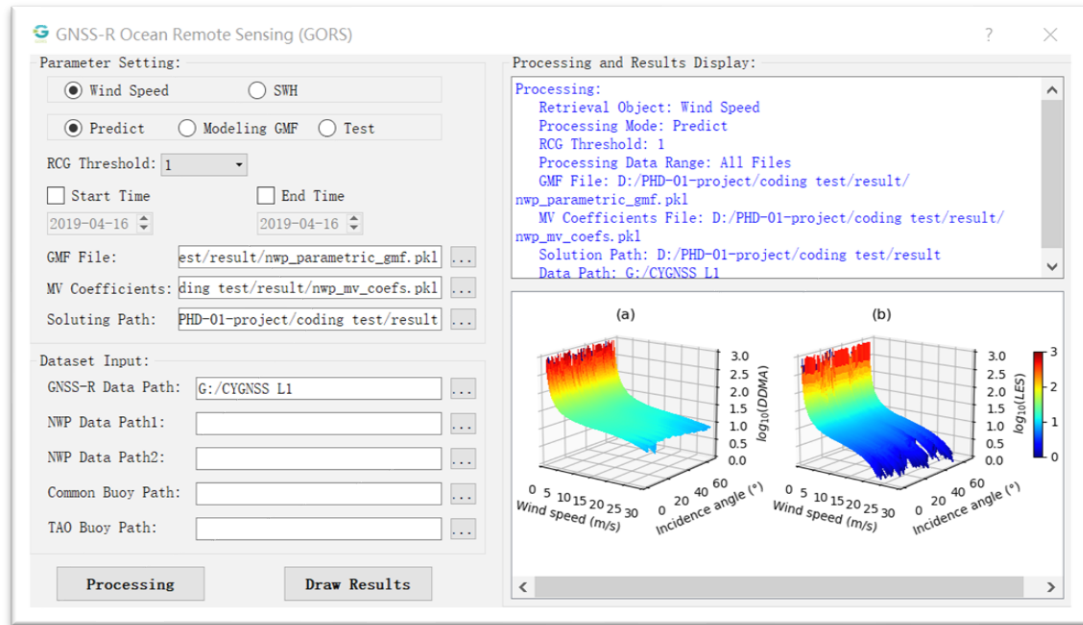


图 8 运行结果展示

## 4.2 主程序模块简介

库 `utilities.py` 和 `figure.py` 提供了独立的工具函数。结果评定在 `output.py` 中实现。`pygnssr.py` 用于读取 CYGNSS L1 文件，提取所需的风速反演数据。其中 `buoy.py`, `era.py`, `gdas.py`, `ccmp.py` 模块分别用于读取浮标、ECMWF ERA5、GDAS 和 CCMP 地表参考真值风速。`match.py` 实现数据匹配。上述所有的模块均为辅助模块，具体实现数据读取、质量控制、数据匹配和结果输出与评定等。

`pyretri.py` 批实现 GNSS-R 风速数据准备，`gmf.py` 实现经验 GMF 函数的建立，`smooth.py` 实现经验 GMF 的平滑，`estimator.py` 实现 DDM 观测量到风速的映射和最小方差估计权系数，`predictor.py` 实现风速预测。上述模块完成风速反演模型的建立和风速预测。

### `utilities.py` 核心函数简介：

```
def write_pkl(filename, data): 将 data 写入文件
def file_list(path): 返回 path 路径下的所有文件
def haversin(organ, lat, lon): 计算球面距离
def shift_lon(lon): 经度由[0,360]范围转换为[-180,180]
```

### `figure.py` 核心函数简介：

```
def data_combine(data, x_var, y_var, flag, type): 将 dict={cygnss ID: pandas.DataFrame}数据组合成单个 pandas.DataFrame。
```

**`era.py`, `gdas.py`, `ccmp.py` 模块：**均以面向对象的方式设计，形式基本相同：

`self.pickfile(date, path)`: 根据输入 `date` 对象选择对应的文件。

`self.readnc(filename)`: 读取文件数据。

`self.show()`: 数据展示。

**buoy.py 模块**: 基于面向对象的方式设计但稍复杂, 包括不同类型文件数据读取和存储, 风速规化为 U10。

**pygnssr.py 模块**: 基于面向对象的方式设计, 用于读取 CYGNSS L1 观测文件。

`self.readnc(filename)`: 文件读取。

`self.getVariables()`: 变量选择。

`self.oceanData(type)`: 观测数据整理为 `pandas.DataFrame` 格式。

此外还包括 `self.observators`, `self.signalSource`, `self.addRCG`, `self.timeAveraging` 等函数负责数据类型判读, RCG 计算和 DDM 的时间平均。

**match.py 模块**: 基于面向对象的方式设计。对不同的地表真值风速(浮标、数值天气预报)匹配类设计存在差异。

`self.matchup`: 执行匹配操作, `self.spatialFilter` 和 `self.temperalFilter` 完成时空匹配。此外, 还包括不同的时空插值方法的实现。

**pyretri.py 核心函数简介**:

`def load_cyg_dataset(path, flag=None, start=None, end=None)`: 选择路径 `path` 下起始日期和终止日期内的 GNSS-R 数据。

`def py_retrieval_prepare(file_mag, path)`: 提取反演变量

`def py_retrieval_model(g_path, ccmp_path, filename)`: 匹配格网真值风

**gmf.py 模块**: 基于面向对象的方式设计, 建立离散的 GMF 函数。

**smooth.py 模块**: 基于面向对象的方式设计, 对经验 GMF 函数进行平滑。

**estimator.py 库核心函数简介**:

`def wind_mapping(input)`: 将 DDM 观测量映射为对应的风速。

`def mv_estimator(data, filename)`: 最小方差估计器。

此外包含一系列 2D GMF 的内插方法, 由于此部分执行速度比较慢, 设计了多进程函数, 通过并行处理提升程序运行速度。当训练数据集比较大时, 建议在服务器或者超算上执行该步骤。

**predictor.py 模块**: 基于面向对象的方式设计, 对不同 DDM 观测量对应的风速进行加权, 反演最终风速。

## 5.使用说明

源码程序调用需要输入 GNSS-R 观测数据和地表真值风速数据，可直接读取 CYGNSS L1 文件（TDS 数据需要额外的整理）和 ERA5、GDAS、CCMP、浮标参考风速数据。反演模型及中间过程文件也可以相应输出，均使用的 pickle 模块的高精度模式存储。包括匹配的训练数据集、测试数据集、经验 GMF、平滑 GMF、权系数。

主模块程序执行示例如下：

```
# =====
import os
import numpy as np
from datetime import datetime
from gmfr import EGMF_generator
from smooth import paras_gmf
from estimator import multiproc_mapping, mv_estimator
from predictor import Predictor
from pyretri import load_cyg_dataset, py_retrieval_prepare, py_retrieval_model
np.warnings.filterwarnings('ignore')    # ignore the invalid value warning

TEMP_PATH = r'.\myp1'    # template file of L1 file for wind retrieval
RESULT_PATH = r'.\result'    # result directory
if not os.path.exists(RESULT_PATH):
    os.mkdir(RESULT_PATH)

# =====
## (1) chose the cygnss l1 files as train dataset matchup referenced wind speed
cyg_path = r'.\data\cygl1'    # CYGNSS L1 file saving dir
ccmp_path = r'.\data\ccmp'
start_date = datetime(2017, 12, 1)    # start date of training data
end_date = datetime(2017, 12, 5)    # end date of training data
## pickle training data files
file_mag = load_cyg_dataset(cyg_path, "date", start_date, end_date)
py_retrieval_prepare(file_mag)
## matchup ground true wind speed
train_filename = "train_dataset.pkl"
py_retrieval_model(TEMP_PATH, ccmp_path, train_filename)

# =====
## (2) generate empirical 2D GMF
filename = os.path.join(RESULT_PATH, train_filename)
emp_gmf = EGMF_generator(filename)
emp_gmf.gmf_soc3()
```

```

emp_gmf_filename = "empirical_gmf.pkl"
emp_gmf.save2file(filename=emp_gmf_filename)
# =====
# # (3) smooth emperical 2D GMF
smooth_gmf_filename = "parametric_gmf.pkl"
paras_gmf(emp_gmf_filename, smooth_gmf_filename)
# =====
# # (4)mapping wind of DDM observales
obs_wind_filename = os.path.join(RESULT_PATH, "train_obs_wind.pkl")
multiproc_mapping(train_filename, smooth_gmf_filename, obs_wind_filename)
# =====
# # (5)mv estimate the weighting coefficents
coefs_filename = "ccmp_mv_coefs.pkl"
mv_estimator(obs_wind_filename, coefs_filename)
# =====
# # (6)retrieval final wind speed
wp_predict = Predictor(coefs_filename)
wp_predict.predictor(obs_wind_filename)
wp_predict.saveData("train_final_wind.pkl")

```