Supporting Information for

"**Assessing the Variations of Water Use Efficiency and its Link-age with Land-use Changes with three Different Data Sources: A Case Study of Yellow River, China**", by Huaiwei Sun [1, *], Lin Chen [1], Yong Yang [1], Mengge Lu [1], Hui Qin [1], Bingqian Zhao [1], Mengtian Lu [2], Jie Xue [3, 4, *], Dong Yan [1], submitted to ***Remote Sensing***.

Text S1. The "GEEMEP" package

GEE (Google Earth Engine), a cloud computing platform that stores and processes huge datasets, is now widely used in hydrological parameter calculations. The data in the GEE platform is usually composed of images or features. Each of them can provide us convenience in the calculation.

The "GEEMEP" package was designed by following the proposed framework introduced in the manuscript. The GEE platform facilities the preparation of GEE datasets, bands selection, data processing, and calculation processes. By employing the "gee-python API" in the GEE platform and Python 3.9, a Python package named "GEEMEP" was designed and formed to implement the steps in the proposed framework. Table 1 gives the information related to versions in the "GEEMEP" package.

Table S1. The versions of Python and main modules are applied in the "GEEMEP" package.

Module	Version
Python	3.9
earthengine-api	0.1.292
numpy	1.20.3
scipy	1.7.1
gdal	3.3.3
rasterio	1.2.10

Text S2. Usage of the "GEEMEP" package

1. Introduction

The purpose of this article is to introduce how to implement the maximum entropy generation (MEP) model by invoking three data sources on GEE platform (ERA5, FLDAS, glDAS-2.1) using “GEEMEP” package. Compared with the classical ET model, MEP model can calculate surface heat flux with less input and ensure surface energy balance. “GEEMEP” package can be used to calculate the latent heat flux, sensible heat flux and geothermal flux of different types of soil layers in the three data sources, and realize data pretreatment, statistical analysis and visualization.

GEE（Google Earth Engine）, a cloud computing platform that stores and processes huge datasets, is now widely used in hydrological parameter calculations. GEE's ability to call in Python or Java Script also facilitates the calculation of evapotranspiration.

The data in the GEE platform usually composed of images or features. Each of them can provide us a convenience to computing parameter we want to know. In this study, we use some datasets in the platform to combining MEP methods with GEE to calculate energy flux and assess the variations of water use efficiency.

---

2. Description of "GEEMEP"

2.1 Parameter description of "GEEMEP" package

Rn: net radiation(unit:W/m2)
RnL: net long-wave radiation(unit:W/m2)
Ts: surface temperature(unit:Celsius)
qs: specific humidity(unit:kg/kg)

EMEP: latent heat flux

HMEP: sensible heat flux

GMEP: ground heat flux

type 1 for bare soil surface or short canopy, 2 for dense canopy and 3 for Water-snow-ice surface

2.2 Application

Step1: Import packages

```
import geemep as gm
```

Step 2: Set regions

Users can both import boundary files from GEE database or upload their own boundary files.

```
roi_1 = "USDOS/LSIB_SIMPLE/2017"       # Import boundary files from GEE database
roi_0 = "users/user_name/region_file"  # Import user's private assets
```

Step 3: Computation of heat flux of with MEP model

```
heat_flux_array = gm.mep(
    startDate="2001-01-01",     # Start time of calculation, "YYYY-MM-DD"
    endDate="2020-12-31",       # Start time of calculation, "YYYY-MM-DD"
    it=600,                     # Material thermal inertia parameter, 600-900 is the recommended
value
    roi=roi_0,                  # Research region
    timeRes='year',             # Calculation time step, "year" / "month" / "day"
    spatRes = 10000,            # Spatial sampling scale (spatial resolution), m
    src = 'fldas'               # Data source, "era5" / "fldas" / "gldas"
    )

#  heat_flux_array: An array concludes sensible heat flux(HMEP),latent heat flux(EMEP),ground
heat flux(QMEP)
```

Step 4: Computation of actual evapotranspiration

```
et_a = gm.ETa(heat_flux_array)      # Convert energy flux to actual evapotranspiration, mm, 1Wm-
2=0.0864mmd-1
```

Step 5: Visualization of results

```
gm.showfig(eta)
```