



Medium- to long-range hydrological forecasting system tutorial for Ganges-Brahmaputra-Meghna River Basins

Made by Byeong-Hee Kim

(email : kbhee87@postech.ac.kr)

Division of Environmental Science and Engineering, POSTECH

1. Input Data Download & Preprocessing

- How to download the ERA5L reanalysis data and preprocessing
- How to download the S2S forecast data and preprocessing

2. VIC-River Routing Model Install & Running Process

- Install the VIC-River Routing Model (VIC-RRM)
- Running process

3. Preprocessing of output data

- Preprocessing of ERA5L streamflow data (reanalysis)
- Preprocessing of S2S streamflow data (forecast)
- Time-series data at gauge station

Part 1. Input Data Download & Preprocessing

Preparing to use the VIC-River Routing Model



- ① Sets up your environment (i.e. `.bashrc` or `.bash_profile`) to use **Fortran, Python, NCL, Perl and Grads** in your account
- ② Unzip 'Routing.gz' in **your home directory**

```
gzip -d Routing.gz
```

- ③ Now you can download input datasets and use the VIC-River Routing Model in your account

****Notice : the red content is a command entered in the shell of Linux**

Steps to produce streamflow forecast data in Bangladesh



- ① Download the recently released ERA5L and ECMWF S2S data (Part 1)
- ② Convert the downloaded data into the input data format of the model (Part 1)
- ③ Build the VIC-River Routing Model in your account (Part 2)
- ④ Generating gridded observed streamflow data using the ERA5L reanalysis data (Part 2)
- ⑤ Generating gridded forecasted streamflow data using the ECMWF S2S data (Part 2)
- ⑥ Preprocessing of output datasets (observed and forecasted streamflow data) (Part 3)

Information of input data for VIC-RRM



	ERA5L (reanalysis)	ECMWF S2S (forecast)
Period	1951~Present	2015~Present
Resolution	10km x 10km	0.1° x 0.1°
Origin data format	netcdf	grib
Forecast period	-	32 days
Ensemble member	-	50
Notice		<ul style="list-style-type: none">• Before July 2023, the forecast data was produced once every three or four days, but currently it is produced every day.

Why we choose the ECMWF S2S forecast data?



Institution	Total runoff Surface runoff Sub-surface runoff	Forecast period	Forecast interval	Ensemble number	Forecast data release date
CMA	○	2015.01-Present	3 or 4 days	3	After 1 month
CNRM	○	2015.05-Present	7 days	24	After 1 month
ECMWF	○	2015.01-Present	daily	50	Within 3 days

- The total runoff and surface runoff (sub-surface runoff) data are required to produce daily streamflow using the VIC-River Routing Model.
- In the S2S forecast datasets, only 3 institutions (CMA, CNRM, and ECMWF) release the forecasted them.
- The ECMWF, only, produces the medium- to long-term forecast (~30 days) every day and publishes them within 3 days. This continues to be helpful not only for short-term forecasting but also for medium- to long-term forecasting.

1. Input Data Download & Preprocessing



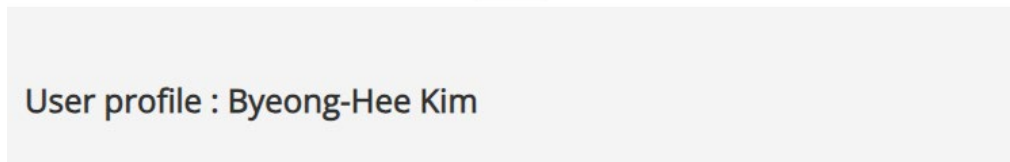
1-1) How to download ERA5L reanalysis data using the python code

- ① Go to 'Climate Data Store' site (<https://cds.climate.copernicus.eu/>)
- ② Log in at this website
- ③ Check your API key



(i) Click your name

(ii) Scroll down and check your UID and API Key



A new CDS soon to be launched - check our informative page [to get ready](#). Meanwhile, expect some disruptions on current CDS - see here [for details](#).

First name Byeong-Hee

Surname Kim

Country Korea, Republic of

Sector Academic/Research

I am registering on behalf of an organisation

API key

Data download requests can be sent programatically via our API. However, a user ID and API key must be sent using HTTP basic authentication.

UID

API Key

1. Input Data Download & Preprocessing

1-1) How to download ERA5L reanalysis data using the python code

④ Make the '.cdsapirc' file at your home directory in your Unix/Linux environment.

(i) `vi $HOME/.cdsapirc`

(ii) Enter as follows in the file `$HOME/.cdsapirc`

```
url: https://cds.climate.copernicus.ed/api/v2
key: {your UID}:{your API key}
verify: 0
```

(iii) Save and out from that file

* If you don't know this, please refer this site

<https://cds.climate.copernicus.eu/api-how-to>

1. Input Data Download & Preprocessing

1-1) How to download ERA5L reanalysis data using the python code

⑤ Edit the python code for downloading ERA5L reanalysis data

(i) `vi $HOME/ROUTING/input/download_code/01-1_ERA5L_runoff_down.py`

(ii) Set download period by changing the 'sday' and 'eday' values in this python code

```
#!/usr/bin/env python

import os
import urllib3
import calendar
import cdsapi
from datetime import datetime, timedelta

urllib3.disable_warnings()

#===== Setting =====
sday = '20221229' # startdate YYYYMMDD
eday = '20221231' # end date  YYYYMMDD
#===== Main program =====
fdir = "../ERA5L/netcdf/"
day_int = 1
```

Change just these

1. Input Data Download & Preprocessing



1-1) How to download ERA5L reanalysis data using the python code

⑤ Edit the python code for downloading ERA5L reanalysis data

(iii) Save and out from the python code

(iv) Run the python code

```
python $HOME/ROUTING/input/download_code/01-1_ERA5L_runoff_down.py
```

```
../ERA5L/netcdf/2022/runoff_daily_20221229.nc
2024-07-15 22:17:13,044 INFO Welcome to the CDS
2024-07-15 22:17:13,044 INFO Sending request to https://cds.climate.copernicus.eu/api/v2/resources/reanalysis-era5-land
2024-07-15 22:17:13,366 INFO Request is queued
2024-07-15 22:17:14,656 INFO Request is running
2024-07-15 22:17:18,987 INFO Request is completed
2024-07-15 22:17:18,987 INFO Downloading https://download-0002-clone.copernicus-climate.eu/cache-compute-0002/cache/data
nal-1721049206.1107361-474-16-930c7398-9319-45f5-a8fb-df07f9477e5e.nc to ../ERA5L/netcdf/2022/runoff_daily_20221229.nc (
2024-07-15 22:17:21,933 INFO Download rate 91.5K/s
../ERA5L/netcdf/2022/runoff_daily_20221230.nc
2024-07-15 22:17:22,521 INFO Welcome to the CDS
2024-07-15 22:17:22,521 INFO Sending request to https://cds.climate.copernicus.eu/api/v2/resources/reanalysis-era5-land
2024-07-15 22:17:22,838 INFO Request is queued
2024-07-15 22:17:25,934 INFO Request is running
2024-07-15 22:17:28,474 INFO Request is completed
2024-07-15 22:17:28,474 INFO Downloading https://download-0001-clone.copernicus-climate.eu/cache-compute-0001/cache/data
nal-1721049215.2724652-21692-10-7f2df556-7388-40d7-b9ac-c361377fd6a6.nc to ../ERA5L/netcdf/2022/runoff_daily_20221230.nc
2024-07-15 22:17:32,229 INFO Download rate 71.8K/s
```

1. Input Data Download & Preprocessing

1-1) How to download ERA5L reanalysis data using the python code

⑤ Edit the python code for downloading ERA5L reanalysis data

(v) The downloaded data is located in the path below.

`$HOME/ROUTING/input/ERA5L/netcdf/{YYYY}/runoff_daily_{YYYYMMDD}.nc`

```
[guest@water2]: [/home/guest/bhkim/ROUTING/input/ERA5L/netcdf/2022]$ ll
total 544
-rw-rw-r-- 1 guest guest 276052 2024-07-15 22:17 runoff_daily_20221229.nc
-rw-rw-r-- 1 guest guest 276056 2024-07-15 22:17 runoff_daily_20221230.nc
```

1. Input Data Download & Preprocessing

1-2) Preprocessing the ERA5L reanalysis data to convert the input data format for the VIC-RRM (netcdf file → binary & ctl file)

- ① Open the NCL code for the ERA5L preprocess

`vi $HOME/ROUTING/input/download_code/01-2_ERA5L_Preprocess.ncl`

- ② Set 'sday' and 'eday' variables according to the period of downloaded ERA5L data

sday : start date of ERA5L

eday : end date of ERA5L

opt : option to set whether to recreate already existing data

opt = 0 ← default

opt = 1 ← recreate

```
==== Setting =====
sday = 20230101
eday = 20230105
; If you want to remake existed dataset, you need to set opt = 1
opt = 0 ; 0 : default option, 1 : remake option
==== Main program =====
dmiss = -999.9
fdir = "../ERA5L/netcdf/"

siy = sday/10000
eiy = eday/10000

yyyymmdd = yyyymmdd_time(siy,eiy,"integer")

nsid = ind(yyyymmdd .eq. sday)
neid = ind(vvvymmdd .eq. eday)
```

Change just these

1. Input Data Download & Preprocessing

1-2) Preprocessing the ERA5L reanalysis data to convert the input data format for the VIC-RRM (netcdf file → binary & ctl file)

③ Run this NCL script

```
ncl $HOME/Routing/input/download_code/01-2_ERA5L_Preprocess.ncl
```

```
[guest@water2]:[/home/guest/bhkim/Routing/input/download_code]$ncl 03_ERA5L_Preprocess.ncl
Copyright (C) 1995-2019 - All Rights Reserved
University Corporation for Atmospheric Research
NCAR Command Language Version 6.6.2
The use of this software is governed by a License Agreement.
See http://www.ncl.ucar.edu/ for more details.
(0)      Make Bin file : ../ERA5L/bin/2024/ERA5L_runoff_daily_2024010100
(0)      Make Bin file : ../ERA5L/bin/2024/ERA5L_runoff_daily_2024010200
(0)      Make Bin file : ../ERA5L/bin/2024/ERA5L_runoff_daily_2024010300
(0)      Make Bin file : ../ERA5L/bin/2024/ERA5L_runoff_daily_2024010400
(0)      Make Bin file : ../ERA5L/bin/2024/ERA5L_runoff_daily_2024010500
```

④ The converted input data are in \$HOME/Routing/input/ERA5L/bin/

```
[guest@water2]:[/home/guest/bhkim/Routing/input/ERA5L/bin/2024]$ll
total 2680
-rw-rw-r-- 1 guest guest 545412 2024-07-16 13:58 ERA5L_runoff_daily_2024010100
-rw-rw-r-- 1 guest guest 545412 2024-07-16 13:58 ERA5L_runoff_daily_2024010200
```

1. Input Data Download & Preprocessing

2-1) How to download S2S forecast data using the python code

- ① Go to 'ECMWF' site (https://ecmwf.int/)
- ② Log in at this website
- ③ Check your ECMWF API key

- (i) Click 'Computing'
- (ii) Click 'Software'
- (iii) Click 'The ECMWF Web API'
- (iv) Click 'ECMWF API key' link
- (v) You can see your API Key

The screenshot shows the ECMWF website interface. The 'Computing' menu item is highlighted in red. Below it, the 'Software' sub-menu is also highlighted in red. Under 'Software', the link 'The ECMWF Web API' is highlighted in red. A diagram below the text shows 'My app' connected to the 'ECMWF data archive' via a 'server.retrieve()' function call. The diagram also shows data being received in NetCDF, GRIB, or JSON formats. A 'Prerequisites' section lists: 'An ECMWF account (this determines which data you can receive)', 'An ECMWF API key you need to be logged in to access this URL', and 'Access to the internet'.

The screenshot shows the 'Your API key' page on the ECMWF website. It displays the user's registered email and API key in input fields. Below these fields, the content of the \$HOME/.ecmwfapi.rc file is shown, which is a JSON object containing the API URL, key, and email. The JSON object is highlighted in red.

```

{
  "url" : "https://api.ecmwf.int/v1",
  "key" : {your API key}
  "email" : {your email}
}

```

1. Input Data Download & Preprocessing

2-1) How to download S2S forecast data using the python code

④ Make the '.ecmwfapirc' file at your home directory in your Unix/Linux environment.

- (i) `vi $HOME/.ecmwfapirc`
- (ii) Enter as follows in the file `$HOME/.ecmwfapirc`

```
{  
  "url" : "https://api.ecmwf.int/v1",  
  "key" : "{your API Key}",  
  "email" : "{your email} "  
}
```

(iii) Save and out from that file

ECMWF Help Byeong-Hee Kim

Your API key

Your registered email:

Your API key:

Content of `$HOME/.ecmwfapirc`

```
{  
  "url" : "https://api.ecmwf.int/v1",  
  "key" : ,  
  "email" :   
}
```

1. Input Data Download & Preprocessing

2-1) How to download S2S forecast data using the python code

⑤ Edit the python code for downloading S2S forecast data

(i) `vi $HOME/ROUTING/input/download_code/02-1_S2S_runoff_down.py`

(ii) Set download period by changing the 'sday', 'eday', and 'nens' values in this python code

```
#!/usr/bin/env python

import sys
import os
import calendar
import cdsapi
from datetime import datetime, timedelta
from ecmwfapi import ECMWFDataServer

#===== Setting =====
sday = "2024-01-01" # start date YYYY-MM-DD
eday = "2024-01-05" # end date YYYY-MM-DD
nens = 5 # forecast ensemble number. The 'nens' must be less than 50. (1~50)
#===== Main Program =====
if nens > 50:
```

Change just these

sday = "2024-01-01"
eday = "2024-01-05"
nens = 5

1. Input Data Download & Preprocessing

2-1) How to download S2S forecast data using the python code

⑤ Edit the python code for downloading S2S forecast data

(iii) Save and out from the python code

(iv) Run the python code

```
python $HOME/ROUTING/input/download_code/02-1_S2S_runoff_down.py
```

```
[guest@water2]:[/home/guest/bhkim/ROUTING/input/download_code]$python 02_ECMWF_S2S_runoff_down.py
2023-07-01
../S2S/grib/ECMWF/ro/ro_2023-07-01
2024-07-15 23:11:55 ECMWF API python library 1.6.3
2024-07-15 23:11:55 ECMWF API at https://api.ecmwf.int/v1
2024-07-15 23:11:56 Welcome Byeong-Hee Kim
2024-07-15 23:12:00 In case of problems, please check https://confluence.ecmwf.int/display/WEBAPI/Web+API+FAQ or contact
servicedesk@ecmwf.int
2024-07-15 23:12:00 ----- WARNING -----
2024-07-15 23:12:00 Access to this dataset is transitioning to a new interface, dates to be announced soon
2024-07-15 23:12:00 For more information on how to access this data in the future, visit https://confluence.ecmwf.int/x/
-wUjEw
2024-07-15 23:12:00 -----
2024-07-15 23:12:01 Request submitted
2024-07-15 23:12:01 Request id: 66952d4a77b8712b97c97d31
2024-07-15 23:12:01 Request is submitted
2024-07-15 23:12:03 Calling 'nice mars /tmp/20240715-1400/20/tmp-_mars-q4csI2.req'
2024-07-15 23:12:03 Forcing MIR_CACHE_PATH=/data/ec_coeff
2024-07-15 23:12:03 mars - WARN -
2024-07-15 23:12:03 mars - WARN -
2024-07-15 23:12:03 MIR environment variables:
2024-07-15 23:12:03 MIR_CACHE_PATH=/data/ec_coeff
2024-07-15 23:12:03 MIR_LSM_NAMED=1km.climate.v013
2024-07-15 23:12:03 Using MARS binary: /usr/local/apps/mars/versions/6.33.17.3/bin/mars.bin
```

1. Input Data Download & Preprocessing

2-1) How to download S2S forecast data using the python code

⑤ Edit the python code for downloading ERA5L reanalysis data

(v) The downloaded data is located in the path below.

`$HOME/ROUTING/input/S2S/ECMWF/grib/{YYYY}/ro_{YYYY}-{MM}-{DD}`

```
[guest@water2]: [/home/guest/bhkim/ROUTING/input/S2S/ECMWF/grib/2024]$ll
total 540
-rw-rw-r-- 1 guest guest 108215 2024-07-16 14:45 ro_2024-01-01
-rw-rw-r-- 1 guest guest 108215 2024-07-16 14:45 ro_2024-01-02
-rw-rw-r-- 1 guest guest 108215 2024-07-16 14:45 ro_2024-01-03
-rw-rw-r-- 1 guest guest 108215 2024-07-16 14:45 ro_2024-01-04
-rw-rw-r-- 1 guest guest 108215 2024-07-16 14:46 ro_2024-01-05
```

1. Input Data Download & Preprocessing

2-2) Preprocessing the S2S forecast data to convert the input data format for the VIC-RRM (grib file → netcdf file → binary & ctl file)

- ① Open the NCL code to convert from grib to netcdf in the S2S forecast data
vi \$HOME/ROUTING/input/download_code/02-2_S2S_grib_to_netcdf.ncl
- ② Set 'sday' and 'eday' variables according to the period of downloaded S2S forecast data

sday : start date of S2S

eday : end date of S2S

opt : option to set whether to recreate already existing data

opt = 0 ← default

opt = 1 ← recreate

```
===== Setting =====
sday = 20240101
eday = 20240105
; If you want to remake existed dataset, you need to set opt = 1
opt = 0 ; 0 : default option, 1 : remake option
----- Main Program -----
model = "ECMWF"
varn = "WROD_P11_L1_GLL0_acc"
new_var = "ro"

sy = sday/10000
ey = eday/10000

yyyymmdd = yyyymmdd_time(sy,ey,"integer")
sind = ind(yyyymmdd .eq. sday)
eind = ind(yyyymmdd .eq. eday)
```

Change just these

1. Input Data Download & Preprocessing

2-2) Preprocessing the S2S forecast data to convert the input data format for the VIC-RRM (grib file → netcdf file → binary & ctl file)

③ Run this NCL script

```
ncl $HOME/ROUTING/input/download_code/02-2_S2S_grib_to_netcdf.ncl
```

```
[guest@water2]: [/home/guest/bhkim/ROUTING/input/download_code]$ ncl 04-1_S2S_grib_to_netcdf.ncl
Copyright (C) 1995-2019 - All Rights Reserved
University Corporation for Atmospheric Research
NCAR Command Language Version 6.6.2
The use of this software is governed by a License Agreement.
See http://www.ncl.ucar.edu/ for more details.
(0)      ... Make new netcdf file
(0)      ... Make new netcdf file
(0)      ... Make new netcdf file
```

④ The converted input data are in \$HOME/ROUTING/input/S2S/ECMWF/netcdf/{YYYY}/

```
[guest@water2]: [/home/guest/bhkim/ROUTING/input/S2S/ECMWF/netcdf/2024]$ ll
total 760
-rw-rw-r-- 1 guest guest 154340 2024-07-16 14:51 S2S_runoff_daily_20240101.nc
-rw-rw-r-- 1 guest guest 154340 2024-07-16 14:51 S2S_runoff_daily_20240102.nc
-rw-rw-r-- 1 guest guest 154340 2024-07-16 14:51 S2S_runoff_daily_20240103.nc
-rw-rw-r-- 1 guest guest 154340 2024-07-16 14:51 S2S_runoff_daily_20240104.nc
```

1. Input Data Download & Preprocessing

2-2) Preprocessing the S2S forecast data to convert the input data format for the VIC-RRM (grib file → netcdf file → binary & ctl file)

- ⑤ Open the NCL code to convert from netcdf to bin in the S2S forecast data
vi \$HOME/ROUTING/input/download_code/02-3_S2S_Preprocess.ncl

sday : start date of S2S

eday : end date of S2S

opt : option to set whether to recreate already existing data

opt = 0 ← default

opt = 1 ← recreate

```
===== Setting =====
sday = 20240101
eday = 20240104
; If you want to remake existed dataset, you need to set opt
opt = 0 ; 0 : default option, 1 : remake option
===== Main Program =====

model = "ECMWF"
sy = sday/10000
ey = eday/10000

yyyymmdd = yyyymmdd_time(sy,ey,"integer")
sind = ind(yyyymmdd .eq. sday)
eind = ind(yyyymmdd .eq. eday)
dmiss = -999.9
```

Change just these

1. Input Data Download & Preprocessing

2-2) Preprocessing the S2S forecast data to convert the input data format for the VIC-RRM (grib file → netcdf file → binary & ctl file)

⑥ Run this NCL script

```
ncl $HOME/Routing/input/download_code/02-3_S2S_Preprocess.ncl
```

```
Copyright (C) 1995-2019 - All Rights Reserved
University Corporation for Atmospheric Research
NCAR Command Language Version 6.6.2
The use of this software is governed by a License Agreement.
See http://www.ncl.ucar.edu/ for more details.
(0)      ===== Make a new directory for the initial time 20240101
(0)      --- Make a new bin file for 2024010200, ensemble : 01
(0)      --- Make a new bin file for 2024010300, ensemble : 01
```

⑦ The converted input data are in

```
$HOME/Routing/input/S2S/ECMWF/bin/{YYYY}/{forecast_date}/{ensemble_number}/
```

```
[guest@water2]: [/home/guest/bhkim/Routing/input/S2S/ECMWF/bin/2024/20240101/01]$ ll
total 128
-rw-rw-r-- 1 guest guest 1848 2024-07-16 15:58 S2S_runoff_01_daily_2024010200
-rw-rw-r-- 1 guest guest 1848 2024-07-16 15:58 S2S_runoff_01_daily_2024010300
-rw-rw-r-- 1 guest guest 1848 2024-07-16 15:58 S2S_runoff_01_daily_2024010400
-rw-rw-r-- 1 guest guest 1848 2024-07-16 15:58 S2S_runoff_01_daily_2024010500
```

Part 2. VIC-River Routing Model Install & Running

2. VIC-River Routing Model Install & Running Process



1) VIC-River Routing Model configuration

```
[guest@water2]: [/home/guest/bhkim/Routing]$ls
GIS  input  Output_codes  scripts  src  Workspace
```

Directory name	Clarification
GIS	<ul style="list-style-type: none">• High-resolution Digital Elevation Model (DEM) data• Mask-out data to distinguish between ocean and land for input data
src	<ul style="list-style-type: none">• Fortran codes for streamflow calculation (Model body)
input	<ul style="list-style-type: none">• Input datasets for the VIC-RRM
scripts	<ul style="list-style-type: none">• Perl scripts to run the VIC-RRM
Workspace	<ul style="list-style-type: none">• Model output path
Output_codes	<ul style="list-style-type: none">• NCL scripts that preprocess model output data

2. VIC-River Routing Model Install & Running Process



1) VIC-River Routing Model configuration

1.1) What is in 'GIS'? – DEM data

\$HOME/Routing/GIS/BG_300s : DEM grid data at 10km resolution for Bangladesh

BG_300s_*.bin	Meaning	Units
ACUM	Accumulation area	mile ²
DEM	Elevation	meter
DIR	Flow direction	-
NCNT	Number of grids cells that flow into the target grid cell	-
SLP	Slope	°
SO	Stream order	-
TI	Topographic index of gridcell	-

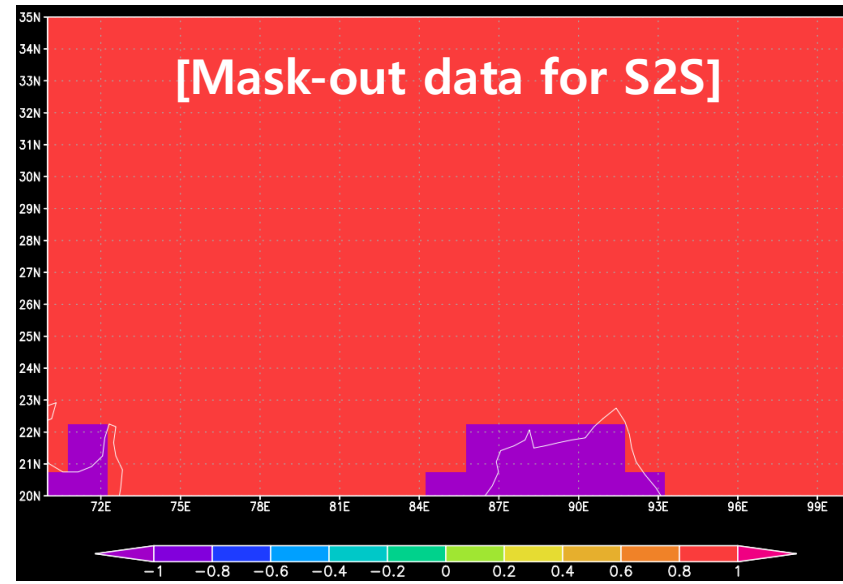
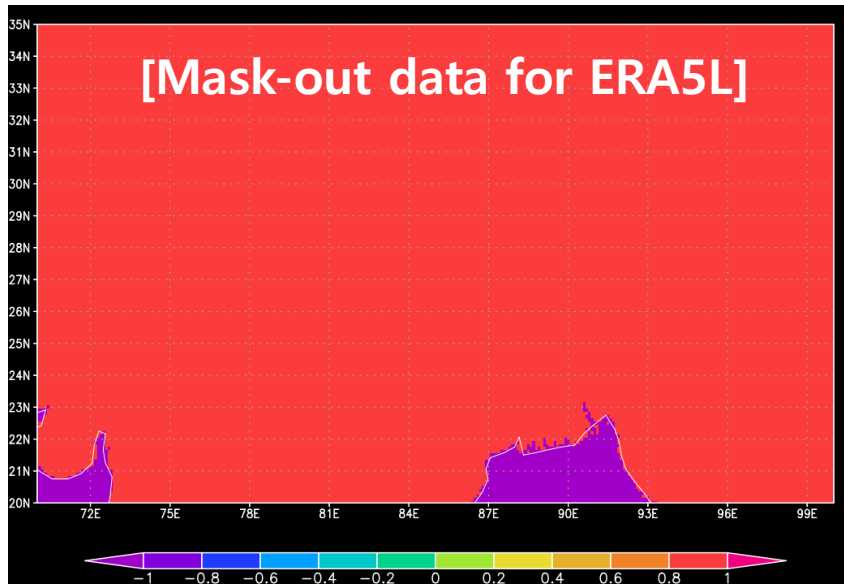
2. VIC-River Routing Model Install & Running Process

1) VIC-River Routing Model configuration

1.1) What is in 'GIS'? – Mask data

\$HOME/Routing/GIS/Mask/ : Mask-out data for input datasets

```
[guest@water2]: [/home/guest/bhkim/Routing/GIS/Mask]$ ll
total 192
-rw-rw-r-- 1 guest guest 181804 2024-07-15 14:33 ERA5_BG_10km.bin
-rw-r--r-- 1 guest guest    225 2024-07-15 14:33 ERA5_BG_10kmctl
-rw-r--r-- 1 guest guest    924 2024-07-16 20:07 S2S_BG_0.1.bin
-rw-r--r-- 1 guest guest    221 2024-07-16 20:07 S2S_BG_0.1ctl
```



2. VIC-River Routing Model Install & Running Process



1) VIC-River Routing Model configuration

1.2) What is in 'src'? – lib, Prep_Routing, Grid_Routing

Name	Meaning
lib	Fortran subroutine codes
Prep_Routing	A Fortran code for calculating parameters including water velocity, length to next grid, and grid area based on the Bangladesh DEM data
Grid_Routing	A Fortran code for calculating streamflow at each grid cell using parameters calculated by the 'Prep_Routing' code and input data (total runoff)

2. VIC-River Routing Model Install & Running Process



2) Build process for the VIC-RRM

① Enter the commands below one by one

cd \$HOME/Routing/src/lib/
make clean
make

cd \$HOME/Routing/src/Prep_Routing
make clean
make

cd \$HOME/Routing/src/Grid_Routing
make clean
make

② Done

(example of build process)

```
[guest@water2]:[/home/guest/bhkim/Routing/src/lib]$ls
area_m.f90  CountDays.o  daysMonth.f90  dist.o        isleap.f90    Makefile       Rout_Grid.f90  zeroMonth.o
area_m.o    createDate.f90 daysMonth.o    grid2River.f90 isleap.o      point2grid.f90 Rout_Grid.o
CountDays.f90 createDate.o  dist.f90       grid2River.o  libRouting.a  point2grid.o   zeroMonth.f90
[guest@water2]:[/home/guest/bhkim/Routing/src/lib]$make clean
rm -rf *.o *.a *.mod
[guest@water2]:[/home/guest/bhkim/Routing/src/lib]$ls
area_m.f90  createDate.f90  dist.f90        isleap.f90  point2grid.f90  zeroMonth.f90
CountDays.f90 daysMonth.f90  grid2River.f90  Makefile    Rout_Grid.f90
[guest@water2]:[/home/guest/bhkim/Routing/src/lib]$make
ifort -c -CB -check all -03 isleap.f90
ifort: warning #10182: disabling optimization; runtime debug checks enabled
ifort -c -CB -check all -03 area_m.f90
ifort: warning #10182: disabling optimization; runtime debug checks enabled
ifort -c -CB -check all -03 dist.f90
ifort: warning #10182: disabling optimization; runtime debug checks enabled
ifort -c -CB -check all -03 point2grid.f90
ifort: warning #10182: disabling optimization; runtime debug checks enabled
ifort -c -CB -check all -03 daysMonth.f90
ifort: warning #10182: disabling optimization; runtime debug checks enabled
ifort -c -CB -check all -03 zeroMonth.f90
ifort: warning #10182: disabling optimization; runtime debug checks enabled
ifort -c -CB -check all -03 CountDays.f90
ifort: warning #10182: disabling optimization; runtime debug checks enabled
ifort -c -CB -check all -03 createDate.f90
ifort: warning #10182: disabling optimization; runtime debug checks enabled
ifort -c -CB -check all -03 Rout_Grid.f90
ifort: warning #10182: disabling optimization; runtime debug checks enabled
ifort -c -CB -check all -03 grid2River.f90
ifort: warning #10182: disabling optimization; runtime debug checks enabled
ar -rcs libRouting.a isleap.o area_m.o dist.o point2grid.o daysMonth.o zeroMonth.o CountDays.o createDate.o Rout_Grid.o
grid2River.o
[guest@water2]:[/home/guest/bhkim/Routing/src/lib]$ls
area_m.f90  CountDays.o  daysMonth.f90  dist.o        isleap.f90    Makefile       Rout_Grid.f90  zeroMonth.o
area_m.o    createDate.f90 daysMonth.o    grid2River.f90 isleap.o      point2grid.f90 Rout_Grid.o
CountDays.f90 createDate.o  dist.f90       grid2River.o  libRouting.a  point2grid.o   zeroMonth.f90
```

2. VIC-River Routing Model Install & Running Process

3) Generate process of observed gridded streamflow data using the ERA5L data

- ① Edit and run the Perl script to create parameters suitable for ERA5L

`vi $HOME/Routing/scripts/ERA5_10km/01_Run_Prep_Routing_ERA5L.pl`

```
#!/usr/bin/perl
# Runs the Preparation codes for the grid routing
# 18 May 2012 jroundy@princeton.edu
use strict;
use warnings;
use FileHandle;
use Term::ANSIColor qw(:constants);
my $start = time();

my $baseDir = "/home/guest/bhkim/Routing";

# Basin related variables
```

Edit to fit your 'Routing' path

`perl $HOME/Routing/scripts/ERA5_10km/01_Run_Prep_Routing_ERA5L.pl`

```
[guest@water2]: [/home/guest/bhkim/Routing/scripts/ERA5_10km]$perl 01_Run_Prep_Routing_ERA5L.pl
/home/guest/bhkim/Routing/GIS/BG_300s/BG_300s_Model_Grid.ctl
#####
Start Prep Routing .....

      0 velocities reset to minimum travel distance of  0.0 of length

End of Prep_Routing
#####
```

2. VIC-River Routing Model Install & Running Process

3) Generate process of observed gridded streamflow data using the ERA5L data

② Open the Perl script to generate observed streamflow data from the ERA5L data

`vi $HOME/Routing/scripts/ERA5_10km/02_Run_Grid_Routing_ERA5L.pl`

③ Edit this script

\$baseDir : path for Routing directory

\$syr : start year for running

\$smn : start month for running

\$sdy : start day for running

\$nt : running days from start date

(ex) If you want to run from Jan 3 to Feb 28,

`$syr = 2024`

`$smn = 1`

`$sdy = 3`

`$nt = 57`

```
##### Setting #####
my $baseDir = "/home/guest/bhkim/Routing";

# Start date setting
my $syr = 2024; #Start year
my $smn = 1; #Start month of simulation
my $sdy = 1; #Start Day of simulation

#Running days from start date
my $nt = 5;

##### Main Program #####
```

Change just these

2. VIC-River Routing Model Install & Running Process

3) Generate process of observed gridded streamflow data using the ERA5L data

- ④ Save and out this Perl script
- ⑤ Run this script

```
perl $HOME/Routing/scripts/ERA5_10km/02_Run_Grid_Routing_ERA5L.pl
```

```
StreamkDir : /home/guest/bhkim/Routing/Workspace/ERA5_BG_flow/Flow
#####
Start Grid Routing ....

Routing BG_300s for          1  days or   -858993460  Months.....

                Land      Ocean      Total      UR-Total % Diff

End Grid_Routing
#####
Time taken was 4 seconds
StreamkDir : /home/guest/bhkim/Routing/Workspace/ERA5_BG_flow/Flow
#####
Start Grid Routing ....
```

2. VIC-River Routing Model Install & Running Process

3) Generate process of observed gridded streamflow data using the ERA5L data

- ⑥ The output data is located in the path below.
 \$HOME/Routing/Workspace/ERA5_BG_flow/Flow/

```
-rw-rw-r-- 1 guest guest 28M 2024-07-16 21:30 Streamflow_20240101.bin
-rw-rw-r-- 1 guest guest 28M 2024-07-16 21:30 Streamflow_20240102.bin
-rw-rw-r-- 1 guest guest 28M 2024-07-16 21:30 Streamflow_20240103.bin
-rw-rw-r-- 1 guest guest 28M 2024-07-16 21:30 Streamflow_20240104.bin
-rw-rw-r-- 1 guest guest 28M 2024-07-16 21:30 Streamflow_20240105.bin
```

<Caution !!>

If the data you are trying to create already exists, an error will appear as shown below. Check the start and end dates of the data you are trying to create.

```
#####
Start Grid Routing ....

Routing BG_300s for          1 days or  -858993460 Months.....

          Land      Ocean      Total      UR-Total % Diff
forrtl: File exists
forrtl: severe (10): cannot overwrite existing file, unit 99, file /home/guest/bhkim/Routing/Workspace/ERA5_BG_flow/Flow/Streamflow_20240105.bin
Image      PC          Routine      Line      Source
Grid_Routing.exe 0000000000429F3B Unknown      Unknown      Unknown
Grid_Routing.exe 00000000004398B0 Unknown      Unknown      Unknown
Grid_Routing.exe 000000000040F941 Unknown      Unknown      Unknown
Grid_Routing.exe 0000000000404462 Unknown      Unknown      Unknown
libc-2.17.so  00002B6717BD8555 __libc_start_main Unknown      Unknown
Grid_Routing.exe 0000000000404369 Unknown      Unknown      Unknown
```

2. VIC-River Routing Model Install & Running Process

4) Generate process of forecasted gridded streamflow data using the ECMWF S2S data

- ① Edit and run the Perl script to create parameters suitable for S2S

vi \$HOME/Routing/scripts/S2S_forecast_100km/01_Run_Prep_Routing_S2S.pl

```
#!/usr/bin/perl
# Runs the Prepartion codes for the grid routing
# 18 May 2012 jroundy@princeton.edu
use strict;
use warnings;
use FileHandle;
use Term::ANSIColor qw(:constants);
my $start = time();

my $baseDir = "/home/guest/bhkim/Routing";
```

Edit to fit your 'Routing' path

perl \$HOME/Routing/scripts/S2S_forecast_100km/01_Run_Prep_Routing_S2S.pl

```
[guest@water2]: [/home/guest/bhkim/Routing/scripts/S2S_forecast_100km]$perl 01_Run_Prep_Routing_S2S.pl
/home/guest/bhkim/Routing/GIS/BG_300s/BG_300s_Model_Grid.ctl
#####
Start Prep Routing .....

      0 velocities reset to minimum travel distance of  0.0 of length

End of Prep_Routing
#####
```

2. VIC-River Routing Model Install & Running Process

4) Generate process of forecasted gridded streamflow data using the ECMWF S2S data

② Open the Perl script to generate forecasted streamflow data from S2S data

`vi $HOME/Routing/scripts/S2S_forecast_100km/02_Run_Grid_Routing_S2S.pl`

```
#!/usr/bin/perl
# Runs the Grid Routing
# Original - 18 May 2012 jroundy@princeton.edu
# Updated - 17 Jul 2024 kbhee87@postech.ac.kr
use strict;
use warnings;
use DateTime;
use FileHandle;
use POSIX qw(strftime);
my $start = time();

my $baseDir = "/home/guest/bhkim/Routing";

my $start_date = "20240101"; # First date of forecast initial time
my $end_date   = "20240103"; # End date of forecast initial time

my $flag = 0;
```

2. VIC-River Routing Model Install & Running Process



4) Generate process of forecasted gridded streamflow data using the ECMWF S2S data

```
#!/usr/bin/perl
# Runs the Grid Routing
# Original - 18 May 2012 jroundy@princeton.edu
# Updated - 17 Jul 2024 kbhee87@postech.ac.kr
use strict;
use warnings;
use DateTime;
use FileHandle;
use POSIX qw(strftime);
my $start = time();

my $baseDir = "/home/guest/bhkim/Routing";

my $start_date = "20240101"; # First date of forecast initial time
my $end_date = "20240103"; # End date of forecast initial time

my $flag = 0;

while ($start_date <= $end_date) {
    my $nt = 30; #Forecast time period (unit : days)
    my $endens = 5; # ensemble number
```

③ Edit this script

Edit to fit your 'Routing' path

Start date of the initial time range for the forecast

End date of the initial time range for the forecast

Enter the forecast duration (days)
(ex. 3, 15, ..., 30)

Enter the ensemble number

2. VIC-River Routing Model Install & Running Process

4) Generate process of forecasted gridded streamflow data using the ECMWF S2S data

(Example1) Situation : You need forecast streamflow data for **10 days** with **the initial condition starting 3 Feb 2024**, and **the number of ensemble is 20**.

```
#!/usr/bin/perl
# Runs the Grid Routing
# Original - 18 May 2012 jroundy@princeton.edu
# Updated - 17 Jul 2024 kbhee87@postech.ac.kr
use strict;
use warnings;
use DateTime;
use FileHandle;
use POSIX qw(strftime);
my $start = time();

my $baseDir = "/home/guest/bhkim/Routing";

my $start_date = "20240203"; # First date of forecast initial time
my $end_date   = "20240203"; # End date of forecast initial time

my $flag = 0;

while ($start_date <= $end_date) {
    my $nt = 10; # Forecast time period (unit: days)
    my $endens = 20; # ensemble number
```

2. VIC-River Routing Model Install & Running Process

4) Generate process of forecasted gridded streamflow data using the ECMWF S2S data

(Example2) Situation : You need forecast streamflow data for **25 days** with **the initial conditions from 15 Feb 2024 to 20 Feb 2024**, and **the number of ensemble is 50**.

```
#!/usr/bin/perl
# Runs the Grid Routing
# Original - 18 May 2012 jroundy@princeton.edu
# Updated - 17 Jul 2024 kbhee87@postech.ac.kr
use strict;
use warnings;
use DateTime;
use FileHandle;
use POSIX qw(strftime);
my $start = time();

my $baseDir = "/home/guest/bhkim/Routing";

my $start_date = "20240215"; # First date of forecast initial time
my $end_date   = "20240220"; # End date of forecast initial time

my $flag = 0;

while ($start_date <= $end_date) {
    my $nt = 25; #Forecast time period (unit : days)
    my $ensens = 50; # ensemble number
```

2. VIC-River Routing Model Install & Running Process

4) Generate process of forecasted gridded streamflow data using the ECMWF S2S data

- ④ Save and out this Perl script
- ⑤ Run this script

```
perl $HOME/Routing/scripts/S2S_forecast_100km/02_Run_Grid_Routing_S2S.pl
```

```
/home/guest/bhkim/Routing/Workspace/S2S_BG_flow/20240101/02/Flow/streamflow_BG_300s.ctl
#####
Start Grid Routing ....

Routing BG_300s for          30  days or   -858993460 Months.....

                Land          Ocean          Total          UR-Total % Diff
2024-01 Totals:    6604.        261017.        267621.        298605.    0.896

End Grid_Routing
#####
Time taken was 24 seconds
StreamkDir : /home/guest/bhkim/Routing/Workspace/S2S_BG_flow/20240101/03/Flow
/home/guest/bhkim/Routing/Workspace/S2S_BG_flow/20240101/03/Flow/streamflow_BG_300s.ctl
#####
Start Grid Routing ....
```

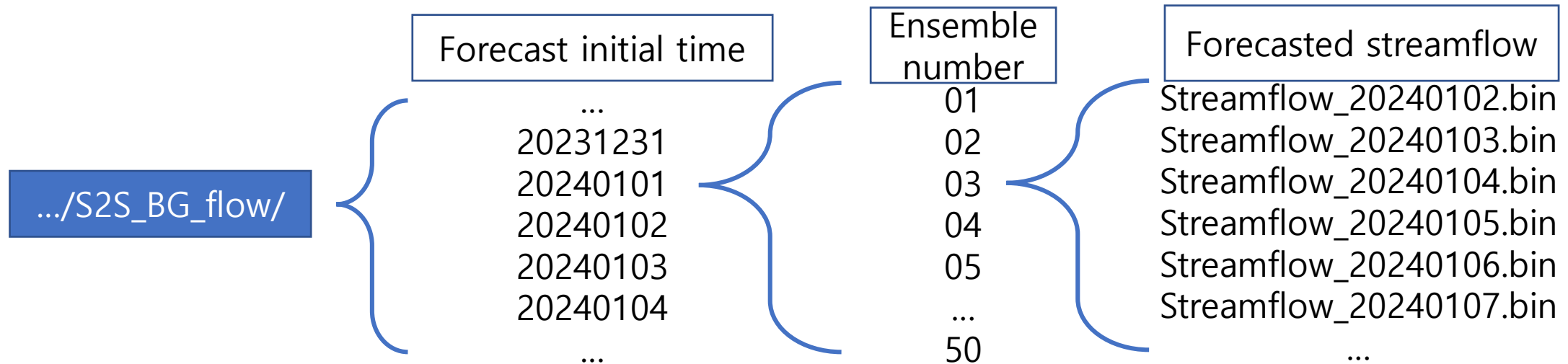
2. VIC-River Routing Model Install & Running Process

4) Generate process of forecasted gridded streamflow data using the ECMWF S2S data

⑥ The output data is located in the path below.

\$HOME/Routing/Workspace/S2S_BG_flow/{Forecast initial time}/{Ensemble number}/Flow/

```
[guest@water2]:[/home/guest/bhkim/Routing/Workspace/S2S_BG_flow/20240101/01/Flow]$ls
Streamflow_20240102.bin  Streamflow_20240110.bin  Streamflow_20240118.bin  Streamflow_20240126.bin
Streamflow_20240103.bin  Streamflow_20240111.bin  Streamflow_20240119.bin  Streamflow_20240127.bin
Streamflow_20240104.bin  Streamflow_20240112.bin  Streamflow_20240120.bin  Streamflow_20240128.bin
Streamflow_20240105.bin  Streamflow_20240113.bin  Streamflow_20240121.bin  Streamflow_20240129.bin
Streamflow_20240106.bin  Streamflow_20240114.bin  Streamflow_20240122.bin  Streamflow_20240130.bin
Streamflow_20240107.bin  Streamflow_20240115.bin  Streamflow_20240123.bin  Streamflow_20240131.bin
Streamflow_20240108.bin  Streamflow_20240116.bin  Streamflow_20240124.bin  streamflow_BG_300s.ctl
Streamflow_20240109.bin  Streamflow_20240117.bin  Streamflow_20240125.bin
```



Part 3. Preprocessing of output data

3. Preprocessing of output data

- **Why the preprocessing process for streamflow output data is needed ?**

1) The VIC-River Routing model's basic output file format has a large data capacity, so it needs to be converted to NetCDF, an efficient compression format.

ex) Binary format data : 28 MB → Compressed NetCDF format data : 245 KB

2) The output files for streamflow are grid data, so it is easy to understand the spatial flow. However, it is difficult to intuitively understand the temporal figures at a particular point.

Therefore, in this third part, we show how to convert data formats for efficient storage and extract time-series for a specific point, such as a gauge station.

- **Programming languages for this preprocessing**

Shell (bash shell), Grads, and NCL

3. Preprocessing of output data

1) Preprocessing of ERA5L streamflow data (reanalysis)

① Update the CTL file of ERA5L streamflow data

i) Open the shell script

`vi $HOME/Routing/Output_codes/code/00_Update_ctl_info_for_ERA5L.sh`

RPATH : path for Routing directory
end_date : End date of newly created ERA5L streamflow data

*** Why we need to update this CTL file ?**
: A CTL file containing spatiotemporal information is required to convert binary data to NetCDF data. So, this file must be updated.

```
#!/bin/bash
#--- Original - 15 Jul 2024 kbhee87@postech.ac.kr
#===== Setting =====
RPATH=${HOME} '/bhkim/Routing' # Routing path
end_date="20240104" # end date of ERA5
#===== Main program =====

sdate="19510101"
start_seconds=$(date -d "$sdate" +%s)
end_seconds=$(date -d "$edate" +%s)
diff_seconds=$((end_seconds - start_seconds))
diff_days=$((diff_seconds / 86400))
```

Change just these

3. Preprocessing of output data

1) Preprocessing of ERA5L streamflow data (reanalysis)

① Update the CTL file of ERA5L streamflow data

ii) Save and out this shell script

iii) Run this shell script

```
sh $HOME/Routing/Output_codes/code/00_Update_ctl_info_for_ERA5L.sh
```

```
[guest@water2]:[/home/guest/bhkim/Routing/Output_codes/code]$sh 00_Update_ctl_info_for_ERA5L.sh
Update the CTL file for ERA5L streamflow data
[guest@water2]:[/home/guest/bhkim/Routing/Output_codes/code]$
```

iv) Path of the updated CTL file

```
/home/guest/bhkim/Routing/Workspace/ERA5_BG_flow/Flow/streamflow_BG_300s.ctl
```

3. Preprocessing of output data

1) Preprocessing of ERA5L streamflow data (reanalysis)

② Change data format from binary to NetCDF using shell and grads

i) Open the shell script

`vi $HOME/Routing/Output_codes/code/01_ERA5L_streamflow_convert_from_bin_to_netcdf.sh`

Start_date : Start date of newly created ERA5L streamflow data

end_date : End date of newly created ERA5L streamflow data

```
#!/bin/bash
#--- Original - 15 Jul 2024 kbhee87@postech.ac.kr

#===== Setting =====
start_date="20240101" # start date
end_date="20240104" # end date

#===== Main program =====

current_date=$(date -d "$start_date" +%Y%m%d)
end_date=$(date -d "$end_date" +%Y%m%d)

while [ "$current_date" -le "$end_date" ]; do
    fdir='../Workspace/ERA5_BG_flow/Flow/streamflow_BG_30
    fout='../Output_codes/output/netcdf/ERA5L/Streamflow_
```

Change just these

3. Preprocessing of output data



1) Preprocessing of ERA5L streamflow data (reanalysis)

② Change data format from binary to NetCDF using shell and grads

ii) Save and out this shell script

iii) Run this shell script

```
sh $HOME/ROUTING/Output_codes/code/01_ERA5L_streamflow_convert_from_bin_to_netcdf.sh
```

```
[guest@water2]: [/home/guest/bhkim/ROUTING/Output_codes/code]$sh 01-1_ERA5L_streamflow_convert_from_bin_to_netcdf.sh

Welcome to the OpenGrADS Bundle Distribution
-----

For additional information enter "grads -h".

Starting "/usr/local/grads/2.1.0.oga.1/bin/Linux/Versions/2.1.0.oga.1/x86_64/grads -blc 'run make_netcdf_ERA5.gs' " .
.

Grid Analysis and Display System (GrADS) Version 2.1.0.oga.1
Copyright (C) 1988-2016 by George Mason University
GrADS comes with ABSOLUTELY NO WARRANTY
See file COPYRIGHT for more information

Config: v2.1.0.oga.1 little-endian readline grib2 netcdf hdf4-sds hdf5 opendap-grids, stn athena geotiff shapefile cairo
Issue 'q config' command for more detailed configuration information
Loading User Defined Extensions table </usr/local/grads/2.1.0.oga.1/bin/Linux/Versions/2.1.0.oga.1/x86_64/gex/udxt> ...
ok.
GX Package Initialization: Size = 11 8.5
Running in Batch mode
All files closed; all defined objects released;
All GrADS attributes have been reinitialized
ga-> [EOF]
GX Package Terminated
01-1_ERA5L_streamflow_convert_from_bin_to_netcdf.sh: line 36: kill: (249743) - No such process
```

3. Preprocessing of output data

1) Preprocessing of ERA5L streamflow data (reanalysis)

② Change data format from binary to NetCDF using shell and grads

iv) Path of the converted NetCDF file for the ERA5L streamflow data
\$HOME/ROUTING/Output_codes/output/netcdf/ERA5L

```
[guest@water2]: [/home/guest/bhkim/ROUTING/Output_codes/output/netcdf/ERA5L]$ ll
total 992
-rw-rw-r-- 1 guest guest 250111 2024-07-21 21:32 Streamflow_20240101.nc
-rw-rw-r-- 1 guest guest 249967 2024-07-21 21:32 Streamflow_20240102.nc
-rw-rw-r-- 1 guest guest 249906 2024-07-21 21:32 Streamflow_20240103.nc
-rw-rw-r-- 1 guest guest 250163 2024-07-21 21:32 Streamflow_20240104.nc
```

3. Preprocessing of output data

1) Preprocessing of ERA5L streamflow data (reanalysis)

③ After the converting process for ERA5L streamflow data

You can delete the binary files in `$HOME/Routing/Workspace/ERA5_BG_flow/Flow/` after the converting process (i.e. ②)

If you want to delete them, use the command below

```
rm -rf $HOME/Routing/Workspace/ERA5_BG_flow/Flow/Streamflow_*.bin
```

<Caution !!>

- Never delete state binary files in `$HOME/Routing/Workspace/ERA5_BG_flow/State/`. (These are named 'State_{YYYYMMDD}.bin')
- These are used when generating forecast data or regenerating the ERA5L streamflow data.

3. Preprocessing of output data

2) Preprocessing of S2S streamflow data (forecast)

① Change data format from binary to NetCDF using shell and grads

i) Open the shell script

`vi $HOME/Routing/Output_codes/code/02_S2S_forecast_streamflow_convert_from_bin_to_netcdf.sh`

start_date : Start date of the initial time range for generating forecast streamflow data

end_date : End date of the initial time range for generating forecast streamflow data

end_ens : Last ensemble number

```
#!/bin/bash
#-- Original - 15 Jul 2024 kbhee87@postech.ac.kr
#----- Setting -----
start_date=20240101 #start date of the initial time
end_date=20240103 #end date of the initial time
end_ens=50 #Last ensemble number
#===== Main program =====
```

Change just these

3. Preprocessing of output data

2) Preprocessing of S2S streamflow data (forecast)

① Change data format from binary to NetCDF using shell and grads

ii) Save and out this shell script

iii) Run this shell script

`sh $HOME/Routing/Output_codes/code/02_S2S_forecast_streamflow_convert_from_bin_to_netcdf.sh`

```
Make file...
...../Output_codes/output/netcdf/S2S/20240103/S2S_streamflow_20240103_19.nc

Welcome to the OpenGrADS Bundle Distribution
-----

For additional information enter "grads -h".

Starting "/usr/local/grads/2.1.0.oga.1/bin/Linux/Versions/2.1.0.oga.1/x86_64/grads -blc 'run make_netcdf_s2s.gs' " ...

Grid Analysis and Display System (GrADS) Version 2.1.0.oga.1
Copyright (C) 1988-2016 by George Mason University
GrADS comes with ABSOLUTELY NO WARRANTY
See file COPYRIGHT for more information

Config: v2.1.0.oga.1 little-endian readline grib2 netcdf hdf4-sds hdf5 opendap-grids, stn athena geotiff shapefile cairo
Issue 'q config' command for more detailed configuration information
Loading User Defined Extensions table </usr/local/grads/2.1.0.oga.1/bin/Linux/Versions/2.1.0.oga.1/x86_64/gex/udxt> ...
ok.
GX Package Initialization: Size = 11 8.5
Running in Batch mode
All files closed; all defined objects released;
All GrADS attributes have been reinitialized
```

3. Preprocessing of output data

2) Preprocessing of S2S streamflow data (forecast)

① Change data format from binary to NetCDF using shell and grads

iv) Path of the converted NetCDF file for the S2S forecast streamflow data

`$HOME/ROUTING/output_codes/output/netcdf/S2S/{YYYYMMDD}/S2S_streamflow_{YYYYMMDD}_{ENS}.nc`

Forecast initial time

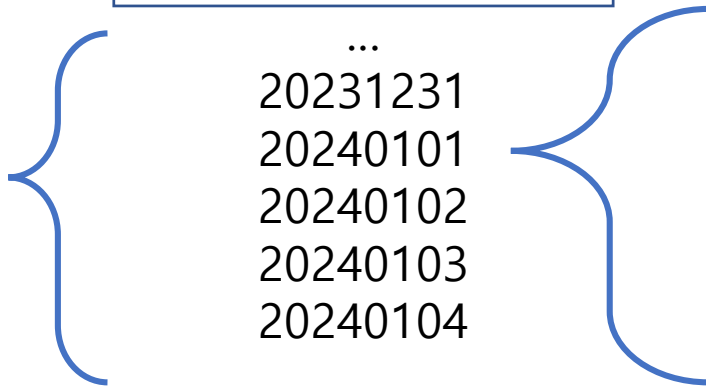
Ensemble number

```
[guest@water2]: [/home/guest/bhkim/ROUTING/output_codes/output/netcdf/S2S]$ ll
total 0
drwxrwxr-x 2 guest guest 51 2024-07-21 22:00 20240101
drwxrwxr-x 2 guest guest 51 2024-07-21 22:01 20240102
drwxrwxr-x 2 guest guest 51 2024-07-21 22:01 20240103
```

Forecast initial time

Forecasted streamflow (NetCDF)

.../S2S/



- S2S_streamflow_20240101_01.nc
- S2S_streamflow_20240101_02.nc
- S2S_streamflow_20240101_03.nc
- S2S_streamflow_20240101_04.nc
- S2S_streamflow_20240101_05.nc

3. Preprocessing of output data

2) Preprocessing of S2S streamflow data (forecast)

② After the converting process for S2S forecast streamflow data

You can delete the binary files in `$HOME/Routing/Workspace/S2S_BG_flow` after the converting process (i.e. ①)

If you want to delete them, use the command below

```
rm -rf $HOME/Routing/Workspace/S2S_BG_flow/*
```

Forecast data is typically generated when new forecast data is needed. Thus, large binary files are not required after conversion to NetCDF.

3. Preprocessing of output data

3) Time-series data at gauge station

① Extract time-series data at a specific point from ERA5L streamflow data

i) Open the NCL script

`vi $HOME/Routing/Output_codes/code/03_ERA5L_gauge_station_streamflow.ncl`

Stat_name : name or ID for gauge station

Stat_lon : Longitude value of a gauge station

Stat_lat : Latitude value of a gauge station

syr : The starting year for which you want to create time-series data

eyr : The ending year for which you want to create time-series data

```
!-- Original - 15 Jul 2024 kbhee87@postech.ac.kr
load "./re_station_lat_lon_using_so.ncl"

===== Setting =====
Stat_name = "Ganges" ; station name
Stat_lon = 89.0255 ; longitude of station
Stat_lat = 24.064 ; latitude of station

syr = 2024 ; start year
eyr = 2024 ; end year

===== Main program =====
```

Change just these

3. Preprocessing of output data

3) Time-series data at gauge station

① Extract time-series data at a specific point from ERA5L streamflow data

ii) Save and out this NCL script

iii) Run this NCL script

```
ncl $HOME/Routing/Output_codes/code/03_ERA5L_gauge_station_streamflow.ncl
```

```
[guest@water2]:[/home/guest/bhkim/Routing/Output_codes/code]$ncl 03_ERA5L_gauge_station_streamflow.ncl
Copyright (C) 1995-2019 - All Rights Reserved
University Corporation for Atmospheric Research
NCAR Command Language Version 6.6.2
The use of this software is governed by a License Agreement.
See http://www.ncl.ucar.edu/ for more details.
(0)      OLAT : 24.064, OLON : 89.0255
(0)      mlat : 24.0417, mlon : 89.0417
(0)      oS0 : 7, rS0 : 7
```

iv) Path of the time-series data at a gauge station from the ERA5L streamflow data

```
$HOME/Routing/Output_codes/output/station/EAR5L/{Stat_name}_ERA5L_daily_streamflow_in_{YYYY}.csv
```

3. Preprocessing of output data



3) Time-series data at gauge station

① Extract time-series data at a specific point from ERA5L streamflow data

v) Structure of the time-series file

First row : Header

First column : Date (YYYYMMDD)

Second column : Streamflow values (m³/s)

(Notice) Missing value = -999.000000

```
Date(YYYYMMDD) , streamflow(m3/s)
20240101 , 4527.273926
20240102 , 4453.748535
20240103 , 4384.325195
20240104 , 4317.537109
20240105 , -999.000000
20240106 , -999.000000
20240107 , -999.000000
20240108 , -999.000000
20240109 , -999.000000
20240110 , -999.000000
20240111 , -999.000000
20240112 , -999.000000
20240113 , -999.000000
20240114 , -999.000000
```

3. Preprocessing of output data

3) Time-series data at gauge station

② Extract time-series data at a specific point from S2S forecast streamflow data

i) Open the NCL script

`vi $HOME/Routing/Output_codes/code/04_S2S_gauge_station_forecast_streamflow.ncl`

Stat_name : name or ID for gauge station

Stat_lon : Longitude value of a gauge station

Stat_lat : Latitude value of a gauge station

forecast_ini : Initial time for forecast

ens : number of ensemble member

```
#!/bin/bash
--- Original - 15 Jul 2024 kbhee87@postech.ac.kr

load "./re_station_lat_lon_using_so.ncl"

;===== Setting =====
Stat_name = "Ganges" ; station name
Stat_lon = 89.0255 ; longitude of station
Stat_lat = 24.064 ; latitude of station

forecast_ini = 20240101 ; forecast initial time
ens = 50 ; number of ensemble member

;===== Main program =====
```

Change just these

3. Preprocessing of output data

3) Time-series data at gauge station

② Extract time-series data at a specific point from S2S forecast streamflow data

ii) Save and out this NCL script

iii) Run this NCL script

```
ncl $HOME/Routing/Output_codes/code/04_S2S_gauge_station_forecast_streamflow.ncl
```

```
[guest@water2]:[/home/guest/bhkim/Routing/Output_codes/code]$ncl 04_S2S_gauge_station_forecast_streamflow.ncl
Copyright (C) 1995-2019 - All Rights Reserved
University Corporation for Atmospheric Research
NCAR Command Language Version 6.6.2
The use of this software is governed by a License Agreement.
See http://www.ncl.ucar.edu/ for more details.
(0)      OLAT : 24.064, OLON : 89.0255
(0)      mlat : 24.0417, mlon : 89.0417
(0)      oS0 : 7, rS0 : 7
```

3. Preprocessing of output data

3) Time-series data at gauge station

② Extract time-series data at a specific point from S2S forecast streamflow data

iv) Path of the time-series data at a gauge station from the S2S forecast streamflow data

`$HOME/Routing/Output_codes/output/station/S2S/{YYYYMMDD}/{Stat_name}_S2S_daily_forecast_streamflow_from_{YYYYMMDD}.csv`



v) Structure of the time-series file (ex. Initial time = 20240101, nens = 50)

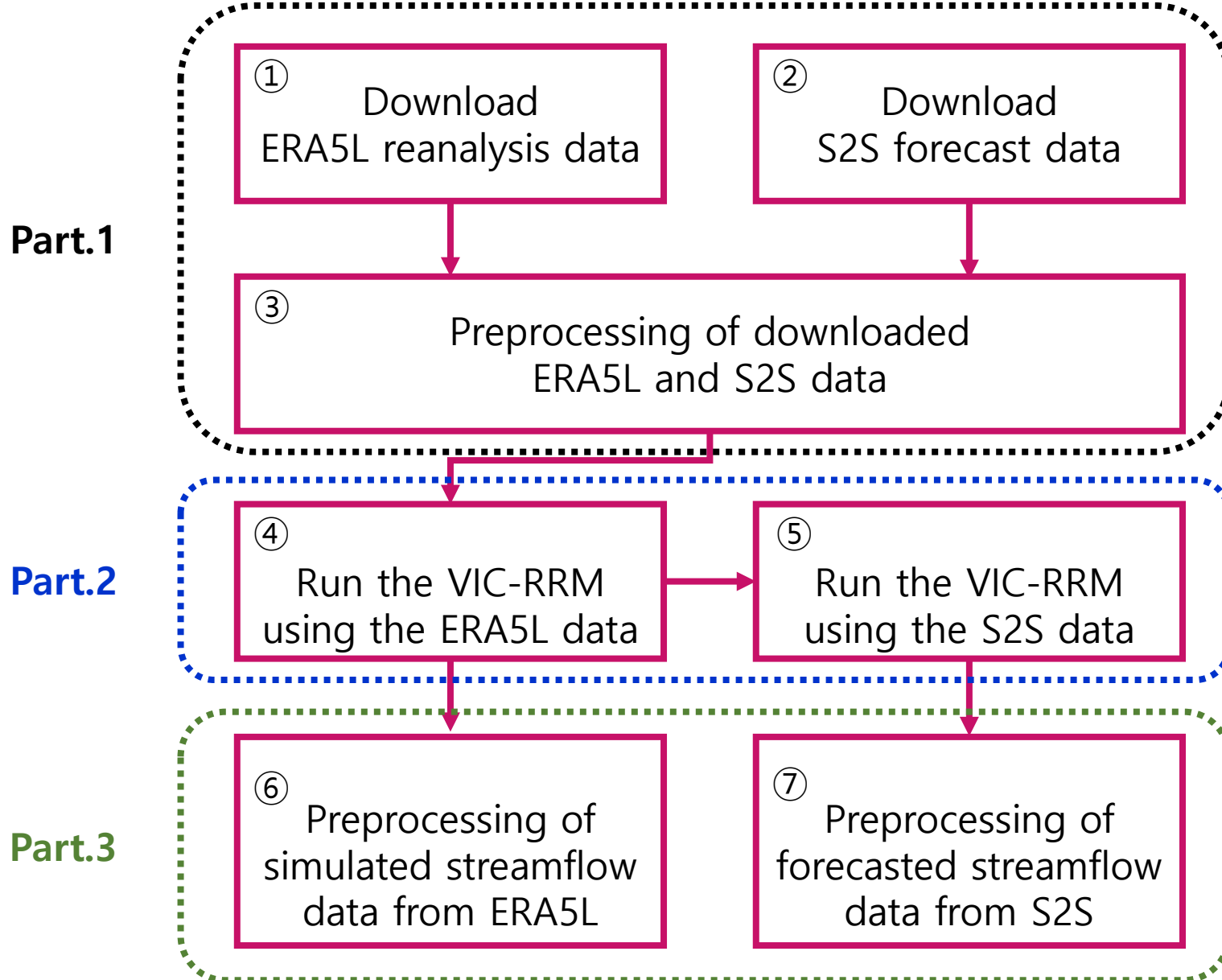
Date(YYYYMMDD)	01	02	...	50
20240102	2818.561	2745.331	...	3154.849
20240103	2615.549	2991.789	...	3354.156
...
20240131	-999.000000	-999.000000	...	-999.000000

Forecast dates

Ensemble number

Forecast streamflow (m³/s)
Missing value = -999.000000

Summary of forecast system process



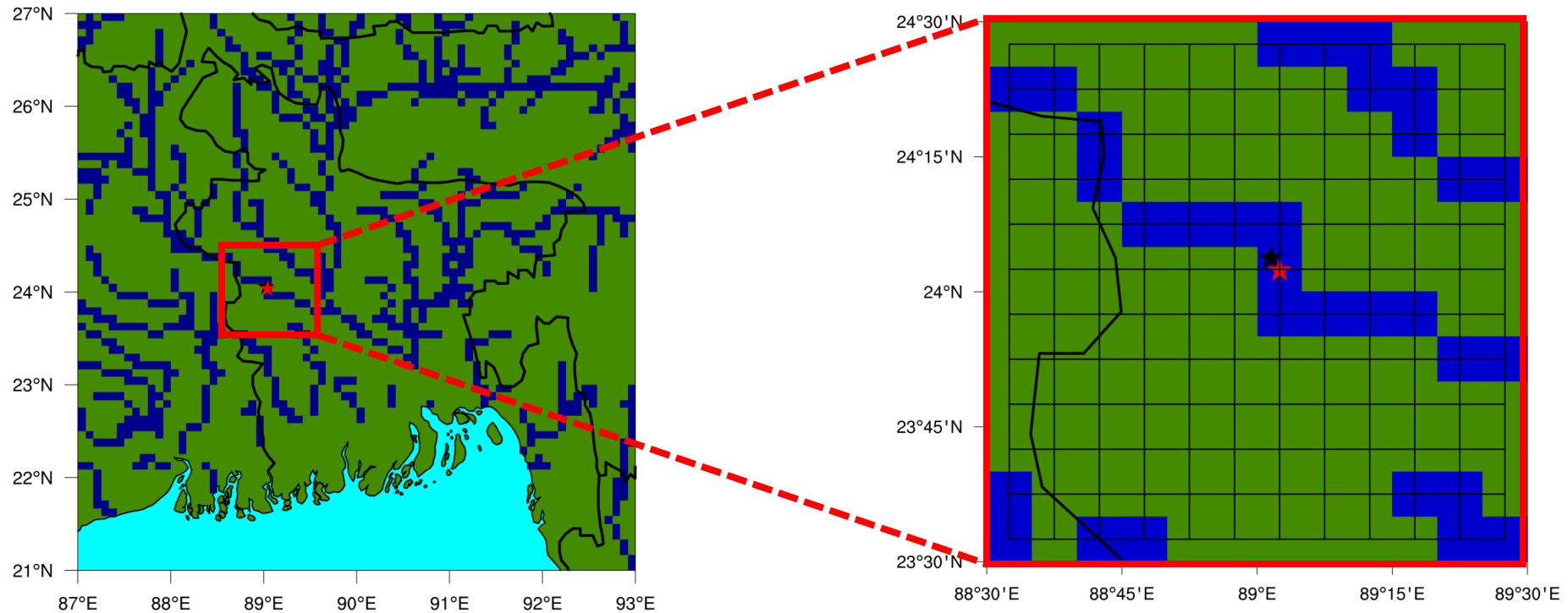
<Caution !!>

Before generating forecasted streamflow data, simulated streamflow data using the ERA5L reanalysis data is required for the initial condition data.

Therefore, ⑤ must be executed after ④

`$HOME/Routing/Output_codes/code/re_station_lat_lon_using_so.ncl`

: A function code to reconstruct the longitude and latitude of the nearest grid point where rivers flow based on the DEM data using the longitude and latitude values of the actual observation station.



★ Actual observation station ★ Nearest grid point from the observation station ■ River network

Thank you