

# Workflow Example

2023-11-14

## Quick checklist summary

1. Setup a repository
  - a. create a repository from template
  - b. add the configuration files
  - c. validate the configuration files
2. Add Submission files
3. Load the data
4. Calculate the ensembles
5. Plot the output

## Library and System setup:

To use full administrator functionality please ensure you install full list of package dependencies including Suggests with:

```
> remotes::install_github("Infectious-Disease-Modeling-Hubs/hubAdmin")
> remotes::install_github("Infectious-Disease-Modeling-Hubs/hubData")
> remotes::install_github("Infectious-Disease-Modeling-Hubs/hubEnsembles")
> remotes::install_github("Infectious-Disease-Modeling-Hubs/hubVis")
```

```
> library(hubAdmin)
> library(hubData)
> library(hubEnsembles)
> library(hubVis)
>
> library(arrow)
> library(dplyr)
```

```
> # Store the path of the hub
> hub_path <- getwd()
```

## Setup a repository

See vignette “hub-setup” on hubUtils package

Create the config files: hub-config/admin.json and hub\_config/tasks.json and validate them:

```
> hubAdmin::validate_config(hub_path)
```

Loading required namespace: jsonvalidate

v Successfully validated config file '/Users/contamin/Documents/SMH/hub\_infrastructure/example-complex-

```
[1] TRUE
```

```
attr(,"config_path")
```

```

/Users/contamin/Documents/SMH/hub_infrastructure/example-complex-scenario-hub/hub-config/tasks.json
attr("schema_version")
[1] "v2.0.0"
attr("schema_url")
https://raw.githubusercontent.com/Infectious-Disease-Modeling-Hubs/schemas/main/v2.0.0/tasks-schema.json
> hubAdmin::validate_config(hub_path, config = "admin")

v Successfully validated config file '/Users/contamin/Documents/SMH/hub_infrastructure/example-complex-
[1] TRUE
attr("config_path")
/Users/contamin/Documents/SMH/hub_infrastructure/example-complex-scenario-hub/hub-config/admin.json
attr("schema_version")
[1] "v2.0.0"
attr("schema_url")
https://raw.githubusercontent.com/Infectious-Disease-Modeling-Hubs/schemas/main/v2.0.0/admin-schema.json
or
> hubAdmin::validate_hub_config(hub_path)

v Hub correctly configured!
'admin.json', 'tasks.json' and 'model-metadata-schema.json' all valid.

$tasks
[1] TRUE
attr("config_path")
/Users/contamin/Documents/SMH/hub_infrastructure/example-complex-scenario-hub/hub-config/tasks.json
attr("schema_version")
[1] "v2.0.0"
attr("schema_url")
https://raw.githubusercontent.com/Infectious-Disease-Modeling-Hubs/schemas/main/v2.0.0/tasks-schema.json

$admin
[1] TRUE
attr("config_path")
/Users/contamin/Documents/SMH/hub_infrastructure/example-complex-scenario-hub/hub-config/admin.json
attr("schema_version")
[1] "v2.0.0"
attr("schema_url")
https://raw.githubusercontent.com/Infectious-Disease-Modeling-Hubs/schemas/main/v2.0.0/admin-schema.json

$model-metadata-schema`
[1] TRUE
attr("config_path")
/Users/contamin/Documents/SMH/hub_infrastructure/example-complex-scenario-hub/hub-config/model-metadata-

attr("config_dir")
/Users/contamin/Documents/SMH/hub_infrastructure/example-complex-scenario-hub/hub-config
attr("schema_version")
[1] "v2.0.0"
attr("schema_url")
[1] "https://github.com/Infectious-Disease-Modeling-Hubs/schemas/tree/main/v2.0.0"

```

## Load the submission files

```
> hub_con <- hubData::connect_hub(hub_path)
> hub_con

-- <hub_connection/UnionDataset> --

* hub_name: "Complex Scenario Hub"
* hub_path:
  '/Users/contamin/Documents/SMH/hub_infrastructure/example-complex-scenario-hub'
* file_format: "csv(2/2)" and "parquet(4/4)"
* file_system: "LocalFileSystem"
* model_output_dir:
  "/Users/contamin/Documents/SMH/hub_infrastructure/example-complex-scenario-hub/model-output"
* config_admin: 'hub-config/admin.json'
* config_tasks: 'hub-config/tasks.json'

-- Connection schema
hub_connection
origin_date: date32[day]
scenario_id: string
location: string
target: string
horizon: int32
output_type: string
output_type_id: double
value: double
model_id: string
age_group: string
target_date: date32[day]

> # Round 1 for example
> round1 <- hub_con %>%
+   dplyr::filter(origin_date == as.Date("2021-03-07")) %>%
+   dplyr::collect()
> head(round1)

# A tibble: 6 x 11
  origin_date scenario_id location target    horizon output_type output_type_id
  <date>      <chr>      <chr>  <chr>      <int>  <chr>          <dbl>
1 2021-03-07  A-2021-03-05  02      inc death        1 quantile         0.01
2 2021-03-07  A-2021-03-05  02      inc death        1 quantile         0.025
3 2021-03-07  A-2021-03-05  02      inc death        1 quantile         0.05
4 2021-03-07  A-2021-03-05  02      inc death        1 quantile         0.1
5 2021-03-07  A-2021-03-05  02      inc death        1 quantile         0.15
6 2021-03-07  A-2021-03-05  02      inc death        1 quantile         0.2
# i 4 more variables: value <dbl>, model_id <chr>, age_group <chr>,
#   target_date <date>
```

## Calculate ensemble

See hubEnsembles package for more information

```
> # Mean ensemble
> round1_ens <- hubEnsembles::simple_ensemble(round1)
> head(round1_ens)

# A tibble: 6 x 11
  model_id origin_date scenario_id location target horizon age_group target_date
  <chr>    <date>      <chr>      <chr>    <chr>    <int> <chr>    <date>
1 hub-ens~ 2021-03-07 A-2021-03-~ 01      cum c~      1 <NA>    NA
2 hub-ens~ 2021-03-07 A-2021-03-~ 01      cum c~      1 <NA>    NA
3 hub-ens~ 2021-03-07 A-2021-03-~ 01      cum c~      1 <NA>    NA
4 hub-ens~ 2021-03-07 A-2021-03-~ 01      cum c~      1 <NA>    NA
5 hub-ens~ 2021-03-07 A-2021-03-~ 01      cum c~      1 <NA>    NA
6 hub-ens~ 2021-03-07 A-2021-03-~ 01      cum c~      1 <NA>    NA
# i 3 more variables: output_type <chr>, output_type_id <dbl>, value <dbl>
```

## Plot

See hubVis package for more information

Data processing: Projection:

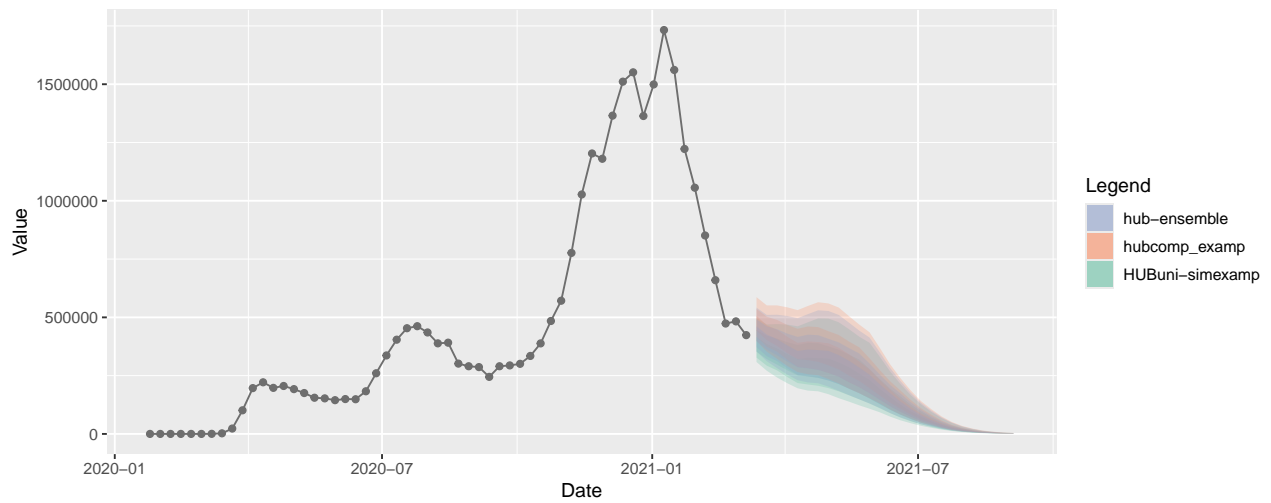
```
> # Aggregate the data (projection + ensemble)
> plot_df <- rbind(round1, round1_ens)
> # Add the target_data column for x-axis
> plot_df <- dplyr::mutate(plot_df, target_date = as.Date(origin_date) +
+                               (horizon * 7) - 1)
> # Remove empty column to avoid issue
> plot_df <- plot_df[!sapply(plot_df, function(k) all(is.na(k)))]
```

Target Data:

```
> target_data <- arrow::read_parquet("target-data/time-series.parquet")
> target_data <- dplyr::filter(target_data, location == "US",
+                               target == "inc case",
+                               date < min(plot_df$target_date))
```

Plot:

```
> plot_a_inccase <- dplyr::filter(plot_df, scenario_id == "A-2021-03-05",
+                               location == "US", target == "inc case")
> plot <- hubVis::plot_step_ahead_model_output(plot_a_inccase, target_data,
+                               interactive = FALSE)
> plot
```



```
> plot_inccase <- dplyr::filter(plot_df, location == "US", target == "inc case")
> plot <- hubVis::plot_step_ahead_model_output(plot_inccase, target_data,
+ facet = "scenario_id",
+ interactive = FALSE)
> plot
```

