Examples Nextflow Analyzer

Clémence Sebe and George Marchment

Here we present a few examples on how to use the Nextflow Analyzer.

Setup

First download the git of the project which can be found here : https://github.com/George-Marchment/AnalyseDonneesNextflow.

Open the folder '*AnalyseDonneesNextflow*' and use the command line : '*sudo python3 setup.py install*' to install the analyser, we can now start working with the tool.

Example 1 : Single Workflow mode (analysis of DSL1 workflow)

The analyzer is developed so that it can extract the information of the processes and the structure of a workflow written in DSL1, since a workflow DSL1 is written in a single file, we give the analyzer that single file. Here we are working on the samba project found here : https://github.com/ifremer-bioinformatics/samba.

First by start by cloning the project, my current directory looks like this :

We can now perform our first analysis, open a terminal and use the following command line to use the analyzer (obviously you'll need to change the addresses) :

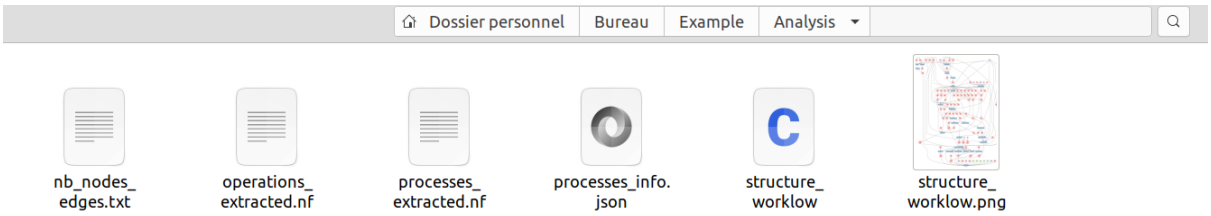
NFanalyzer --input '/home/george/Bureau/Example/samba/main.nf' --results_directory '/home/george/Bureau/Example' --name 'Analysis' --mode 'single' --dev 'T'

- The input address is where the file we want to analyze is found

- Results directory is where we want to save the data
- name is the name of the folder that will be create and where the data will be saved
- mode is the mode we are using, either single or multi mode. Here we are analyzing a single file so single mode
- dev is a boolean which can either be '*T*' or '*F*', when true the analyzer is in developer mode and keeps the developper files which are created (see below) otherwise it deletes them (dev mode is 'off' by default)

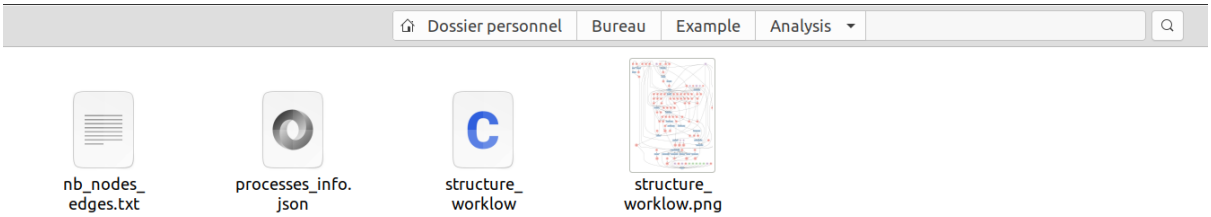
```
george@george-Surface-Laptop:~/Bureau/Example$ NFanalyzer --input '/home/george/Bureau/Example/samba/main.nf' --result
s_directory '/home/george/Bureau/Example' --name 'Analysis' --mode 'single' --dev 'T'

 _   _ _____ _____ _____ _   _ _____ _____ _____ _____ _____ _____ _____ _____
| \ | |  ___|  _  |  ___| | | |  ___|  _  |  ___| ___ \_   _|  ___|  _  |  _  |
|  \| | |__ | | | | |__ | | | | |__ | | | | |__ | |_/ / | | | |__ | | | | | | |
| |\  |  __|| | | |  __|| | | |  __|| | | |  __||    /  | | |  __|| | | | | | |
| | \ | |___\ \_/ / |___\ \_/ / |___\ \_/ / |___| |\ \  | | | |___\ \_/ / |_| |
\_|  \_\____/ \___/\____/ \___/\____/ \___/\____\_| \_| \_/ \____/ \___/ \___/


------------------------------------------------

Developped by Clemence Sebe and George Marchment
------------------------------------------------

Loading Biotools and EDAM libraries. Status : COMPLETED

Single Workflow analysis mode was selected

Analyzing the workflow : /home/george/Bureau/Example/samba/main.nf
Workflow written in DSL1
Extracted 29 processes
Extracted 108 operations
Structure reconstructed
With 29 processes, 104 operations and 212 edges
Results saved in : /home/george/Bureau/Example/Analysis
```

Here is the result of the analyzer, the analyzer found 29 processes, 108 operations, it managed to reconstruct the structure with 29 processes, 104 operations (meaning that there are some channel which are not connected to any processes or other operations hence don't have an effect on the structure) and 212 channels (edges).



Here are the results which are saved in the folder.

- nob_node_edges.txt corresponds to the data on the structure (le number on nodes etc..)
- operations_extarcted and processes_extarcted are the documents which were kept since developer mode was 'on', they correspond to a summary of the operations and processes extracted



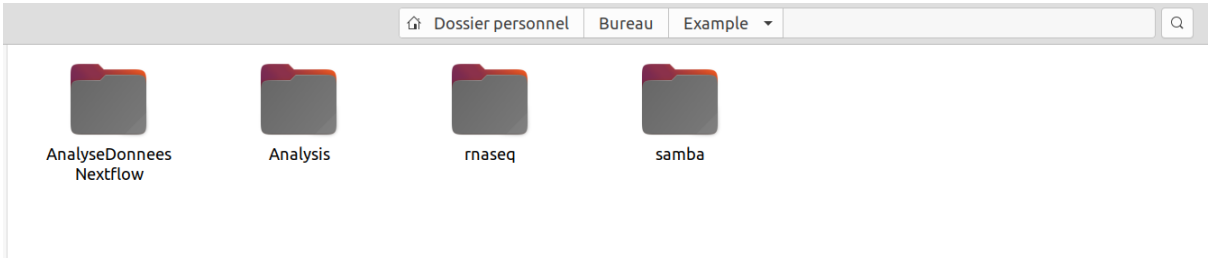
This is what the folder looks like when developer mode is ‘off’

- processes_info.json is the information saved on the processes from the analysis
- Finally structure_workflow is the structure saved from the analysis, it is in 2 formats : dot and png.

Example 2 : Single Workflow mode (analysis of DSL2 workflow)

The analyzer is developed so that it can extract the information of the processes of a workflow written in DSL2 (it cannot extract the structure of a workflow written in DSL2), since a workflow DSL2 is written in multiple files, we give the analyzer a folder containing all the nextflow files of a DSL2 workflow. Here we are working on the rnaseq project found here : https://github.com/nf-core/rnaseq

First by start by cloning the project, my current directory looks like this :



We can now perform the analysis, open a terminal and use the following command line to use the analyzer (obviously you’ll need to change the addresses) :

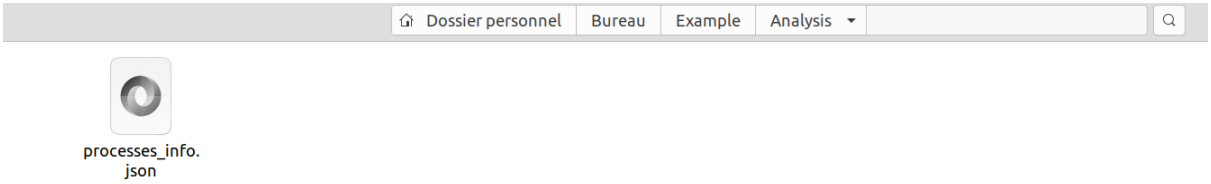
NFanalyzer --input '/home/george/Bureau/Example/rnaseq' --results_directory '/home/george/Bureau/Example' --name 'Analysis' --mode 'single'

When analyzing a workflow written in DSL2, we do not give as input the address to the main but the address of the folder containing all of the nextflow files which define the workflow.

Here is the result of the analyzer, there are many more lines than the first analysis, let's take a look at the last few line :

```
Analyzing /modules/nf-core/modules/hisat2/align/main.nf : 1 processes found
Analyzing /modules/nf-core/modules/hisat2/extractsplicesites/main.nf : 1 processes found
Analyzing /modules/nf-core/modules/hisat2/build/main.nf : 1 processes found
Analyzing /modules/nf-core/modules/qualimap/rnaseq/main.nf : 1 processes found
Analyzing /modules/nf-core/modules/rseqc/junctionsaturation/main.nf : 1 processes found
Analyzing /modules/nf-core/modules/rseqc/inferexperiment/main.nf : 1 processes found
Analyzing /modules/nf-core/modules/rseqc/junctionannotation/main.nf : 1 processes found
Analyzing /modules/nf-core/modules/rseqc/readdistribution/main.nf : 1 processes found
Analyzing /modules/nf-core/modules/rseqc/tin/main.nf : 1 processes found
Analyzing /modules/nf-core/modules/rseqc/innerdistance/main.nf : 1 processes found
Analyzing /modules/nf-core/modules/rseqc/readduplication/main.nf : 1 processes found
Analyzing /modules/nf-core/modules/rseqc/bamstat/main.nf : 1 processes found
Analyzing /modules/nf-core/modules/stringtie/stringtie/main.nf : 1 processes found
Analyzing /subworkflows/nf-core/fastqc_umitools_trimgalore.nf : 0 processes found
Analyzing /subworkflows/nf-core/rseqc.nf : 0 processes found
Analyzing /subworkflows/nf-core/bedgraph_to_bigwig.nf : 0 processes found
Analyzing /subworkflows/nf-core/dedup_umi_umitools.nf : 0 processes found
Analyzing /subworkflows/nf-core/align_hisat2.nf : 0 processes found
Analyzing /subworkflows/nf-core/mark_duplicates_picard.nf : 0 processes found
Analyzing /subworkflows/nf-core/bam_sort_samtools.nf : 0 processes found
Analyzing /subworkflows/nf-core/bam_stats_samtools.nf : 0 processes found
Analyzing /subworkflows/local/input_check.nf : 0 processes found
Analyzing /subworkflows/local/prepare_genome.nf : 0 processes found
Analyzing /subworkflows/local/align_star.nf : 0 processes found
Analyzing /subworkflows/local/quantify_salmon.nf : 0 processes found
Analyzing /subworkflows/local/quantify_rsem.nf : 0 processes found
Total number of processes in the workflow Analysis : 55
Results saved in : /home/george/Bureau/Example/Analysis
george@george-Surface-Laptop:~/Bureau/Example$
```

When analyzing a DSL2 workflow, the analyzer shows each nextflow file found and the number of processes found in that file. Finally it shows the total amount of processes found.

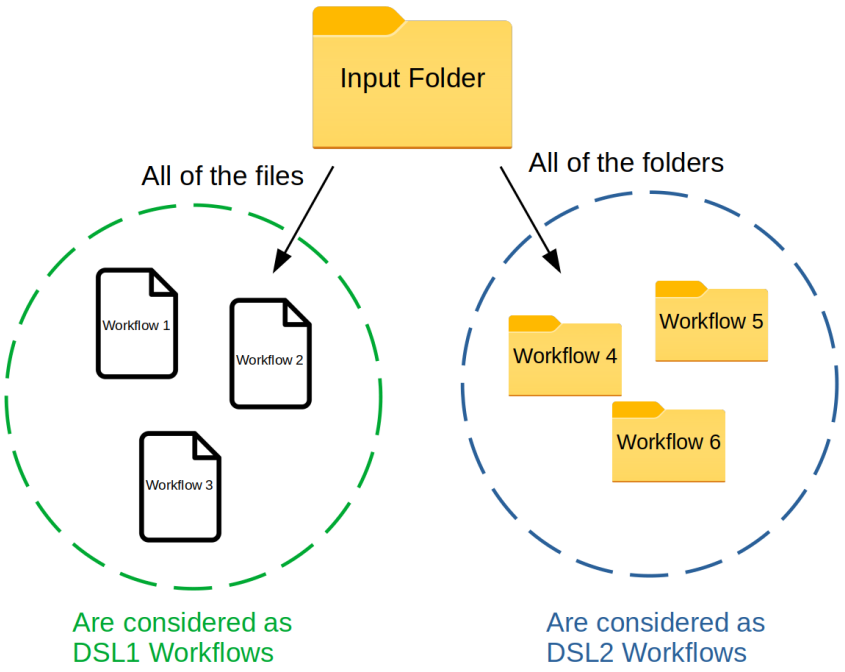


Here is the result of the analysis, since we are analyzing a DSL2 workflow, the structure is inaccessible, we still get the information on the processes.

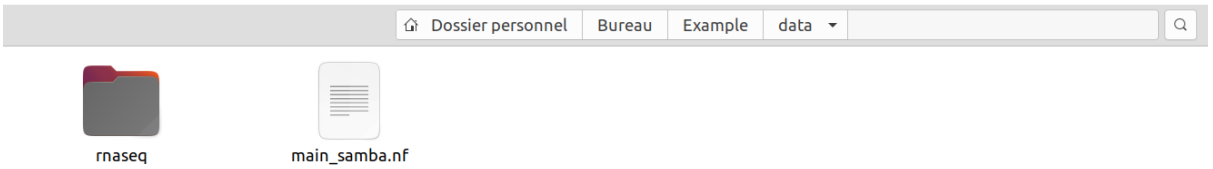
Example 3 : Multi Workflows mode (analysis of DSL1 + DSL2 workflow)

It's great analyzing and extracting data on a single workflow, when wanting to extract information on many workflows, it's inconvenient to manually do the analysis every time. Luckily the nextflow analyzer has a multiple workflow analysis mode, where it can perform the same analysis as a single workflow, multiple times automatically and save the data in a way which is convenient.

Before starting, we need to prepare our data, we are going to analyze the 2 workflows we have previously downloaded. Since DSL1 workflows are found in a single file, we need to give the analyzer that file, and since DSL2 workflows are found in multiple files, we need to give the analyzer the folder containing those files. Well that is how we are going to organize the data, for multi mode we are going to give the analyzer a folder containing multiple files which refer to the DSL1 workflows and multiple folders which refer to the DSL2 workflows.



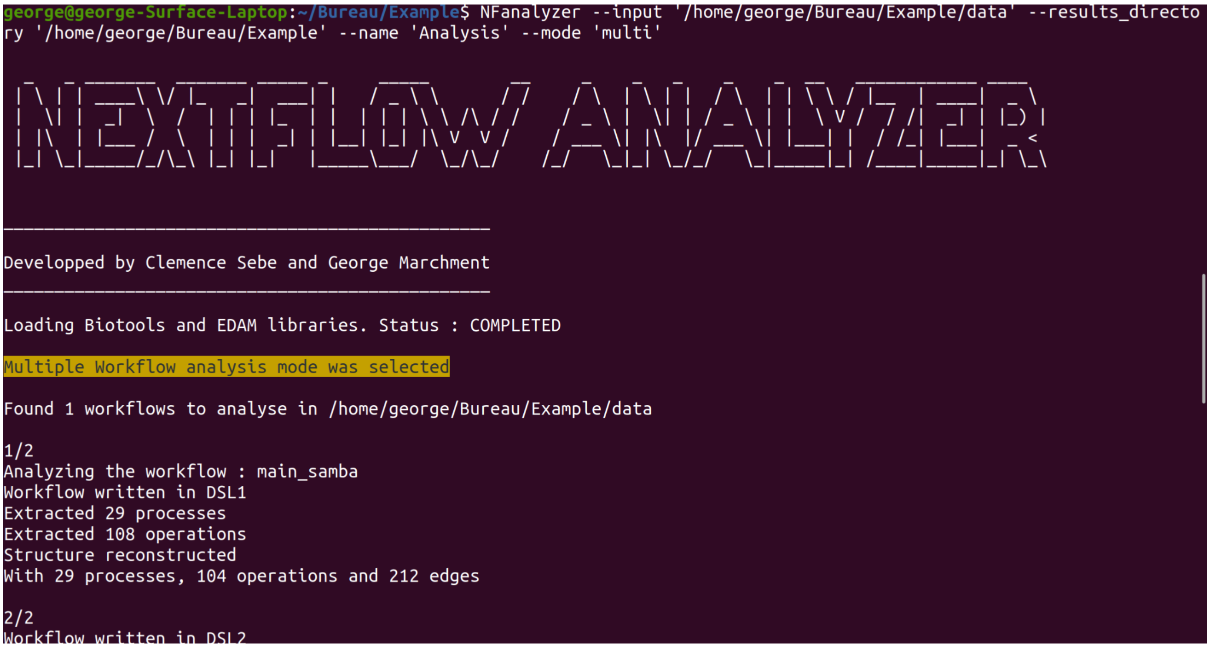
If you are working on the 2 workflows above, you should have a folder which resembles this :



rnaseq is the same folder as the original, main_samba.nf, is the main of the samba workflow.

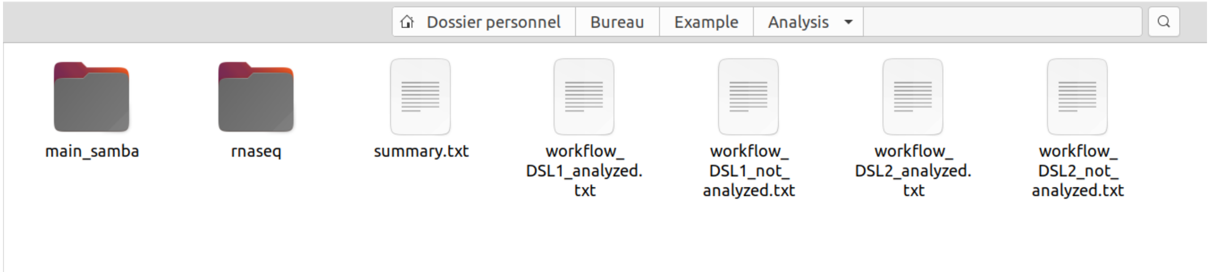
Now we are ready to start the multi analysis, open a terminal and use the following command line to use the analyzer (obviously you'll need to change the addresses) :

NFanalyzer --input '/home/george/Bureau/Example/data' --results_directory '/home/george/Bureau/Example' --name 'Analysis' --mode 'multi'



When analyzing the workflows, the analyzer shows its progress.

Let’s look at the results :



Here there is much more information :

- First in the folders main_samba and rnaseq, there is the same information as when we performed the single analysis of the workflows
- summary.txt is a file which contains the summary of the analysis (number of DSL1 workflows analyzed, number of DSL2 workflows failed to analyze etc..)
- In the remaining files, the addresses of the workflow which have been successfully analyzed or failed to analyze with the corresponding error (for why it failed).

This concludes this document, showing a few examples on how to use the Nextflow analyzer, if you have any questions please contact either :

- clemence.sebe@universite-paris-saclay.fr
- george.marchment@universite-paris-saclay.fr