

# Déploiement de la plateforme de traitement des données phénotypage haut débit 4P sur l'infrastructure France Grilles

V. Nègre, E. David, M. Weiss, P. Burger, R. Chapuis, B.  
Adam, A. Tireau, P. Moreau, A. Tong, G. Colombeau,  
S. Thomas, P. Neveu, J. Pansanel, F. Baret

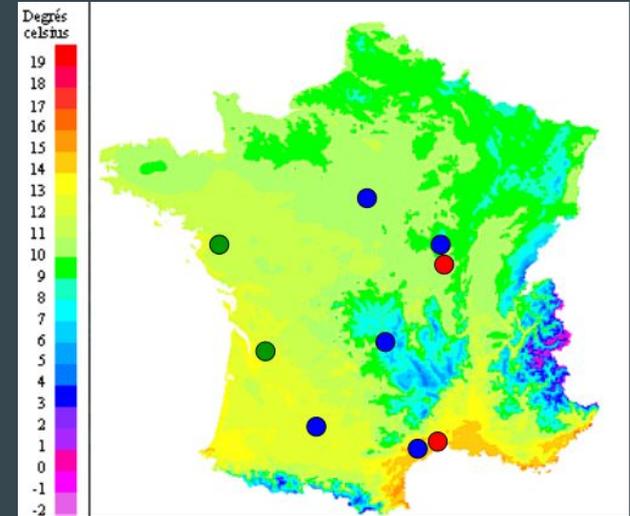


# Le projet



- Projet Investissement d'avenir (2012-2023, 24 M€)
- Infrastructure nationale de phénomique végétale
- Plateformes expérimentales de phénotypage haut-débit réparties sur 9 sites (champ, serre, omique)

## Plateformes PHENOME



- omique
- serre
- champ

# Exemples de dispositifs

Phenoarch



Pheno3C



Auzeville



conditions  
contrôlées

conditions semi-contrôlées  
ou au champ



TRAITEMENT DES DONNÉES  
ACQUISES AU CHAMP

# Objectifs scientifiques

- Caractériser les comportements des variétés d'une même espèce dans un environnement donné (climat, sol, pathogènes, ravageurs) sur des espèces variées
- Des milliers de parcelles (variétés/modalités) à caractériser au cours du cycle de développement de la culture

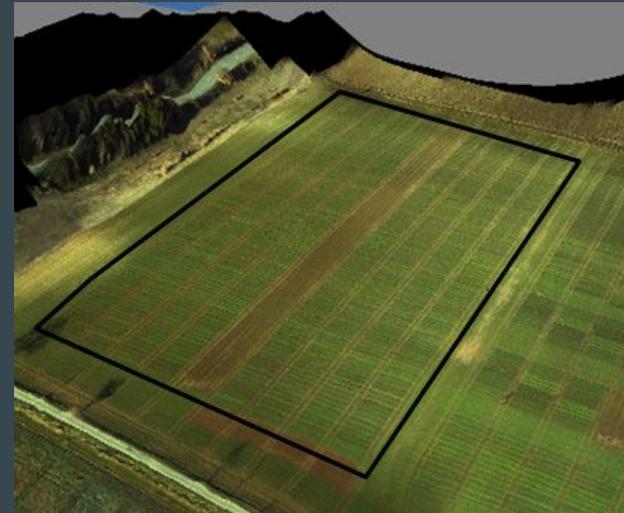
→ Suivi non destructif = télédétection



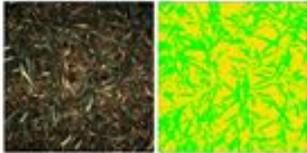
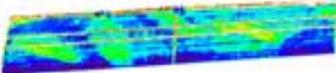
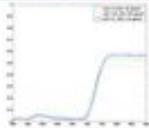
Des vecteurs



Des capteurs



# Que peut-on mesurer concrètement?

			Utilisations
	Camera haute résolution/RVB		<ul style="list-style-type: none"><li>• Identification adventices</li><li>• Enherbement</li><li>• Taux de couverture verte</li><li>• Développement foliaire</li><li>• Comptage de plante</li><li>• Dégâts de nuisibles</li><li>• Biomasse, risque de verse</li></ul>
	LIDAR Mesure de distance et profondeur		<ul style="list-style-type: none"><li>• Structure</li><li>• Port du couvert</li><li>• Interception lumineuse</li><li>• Expérimentation</li></ul>
	Caméra IR thermique		<ul style="list-style-type: none"><li>• Température de surface</li><li>• Stress hydrique</li><li>• Pilotage de l'irrigation</li></ul>
	Camera multispectrale		<ul style="list-style-type: none"><li>• Développement</li><li>• Statut azoté/chlorophylle</li><li>• Stress</li><li>• Pilotage de l'azote</li></ul>
	Spectromètre		<ul style="list-style-type: none"><li>• Chlorophylle/statut azoté</li><li>• Stress (hydrique, ...)</li><li>• Développement</li><li>• Composés de la plante</li></ul>

# Contraintes de développement

- Utilisable par toutes les plateformes d'acquisition PHENOME
  - Compatible avec les vecteurs/capteurs utilisés
  - Compatible avec la plateforme des partenaires (ARVALIS/HIPHEN): partage des modules de traitement
  - Utilisation en version standalone/cloud
- Traçable
  - Gestion et archivage des données/capteurs/vecteurs <-> système **PHIS** (Phenotyping Hybrid Information System)/ontologies
  - Gestion et versionnage des chaînes de traitement de données
- Flexible
  - Générique: intégration de chaînes de traitement développées dans des langages différents (C++, python, matlab), utilisation d'un logiciel commercial sous licence
  - Évolutive: ajout de nouveaux traitements/type de données
  - Utilisation par un public non informaticien (scientifique/technicien)

# Partenaires



Ephesia consult

France Grilles

INRA

hi-phen

ARVALIS

Développement logiciel Ressources calcul et stockage

Développement plateforme

Modules  
traitement  
des données

Modules  
traitement  
des données

Spec/Test/Utilisation

UMR AGIR  
UE PHACC  
UE DIASCOPE

# Fonctionnalités de la plateforme



PLANT  
PHENOTYPING  
PROCESSING  
PLATFORM

Home

Raw Datasets

Processed Datasets

Macros

Modules

Monitoring

Downloads

Data type

Toulouse

Marie Weiss

Amount of data

425.6 GB

Upload raw data

Processed ratio

31 %

Process new data

Launched processes

58

Run a process

Published macros

9

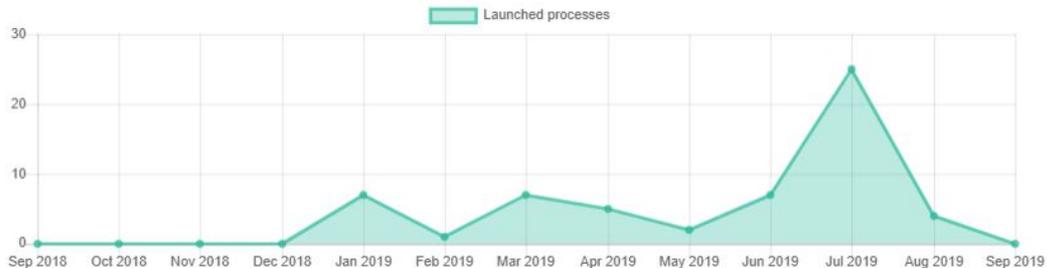
Create a macro

Disk space

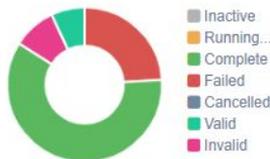
42%

Free disk space

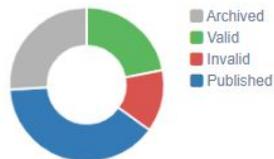
Launched processes over the last 12 months



Processes status



Macros status



Datatypes



## Processed Datasets

+ Process dataset

Process	Creation date	Measuring session	Experiment	Vector	Datatypes	Status	PHIS	Size
All ▾	DD/MM/YYYY	DD/MM/YYYY	All ▾	All ▾	All ▾	All ▾	All ▾	
	20/08/2019 09:00	31/07/2019 09:47	Clouds_Raul	Drone	<a href="#">Extracted Plots</a> <a href="#">Mean Reflectance</a> <a href="#">Orthomosaics</a> <a href="#">Photogram Results</a>	Complete »		23.2 GB (23,216,223 KB)
	13/08/2019 13:41	31/07/2019 09:47	Clouds_Raul	Drone	<a href="#">Extracted Plots</a> <a href="#">Mean Reflectance</a> <a href="#">Orthomosaics</a> <a href="#">Photogram Results</a>	Complete »		23.4 GB (23,380,179 KB)
	12/08/2019 14:45	16/04/2019 14:45	19TE01	Drone	<a href="#">Dense Cloud</a> <a href="#">Photogram Results</a> <a href="#">Soil Height Map</a>	Failed »		9.5 MB (9,542 KB)
	12/08/2019 11:46	31/07/2019 09:47	Clouds_Raul	Drone	<a href="#">Extracted Plots</a> <a href="#">Mean Reflectance</a> <a href="#">Orthomosaics</a> <a href="#">Photogram Results</a>	Failed »		63.2 MB (63,177 KB)

Modules / SoilHeight\_UAVRGB\_SFM

## SoilHeight\_UAVRGB\_SFM

Modify Duplicate Publish Delete

Description

Compute Ground level Height map from UAV flight when no vegetation is on the ground

Compatible species: All (sunflower, wheat, soybean, maize, apple tree, wheat pea fababean, Sunflower, )

Type: Calculation module

Author: Marie Weiss ([marie.weiss@inra.fr](mailto:marie.weiss@inra.fr))

Compatible vectors: Drone

Technical guide: [CanopyHeight\\_SFM\\_0.1.pdf](#)

Source repository : [git@bitbucket.org:phenohdprocessing/soilheight\\_uavrgb\\_sfm.git](https://github.com/phenohdprocessing/soilheight_uavrgb_sfm.git)

Docker image: soilheight\_uavrgb\_sfm:1.0

Status: ✔ Published from 05/06/2019 16:24

Integration status: ✔ Integrated from 01/09/2019 08:12

Macros using this module

[Calcul hauteur sol à partir de dense point cloud \(3DPC\)](#)

[Soil Height from RGB UAV](#)

[Soil Height from RGB UAV from SfM algorithm](#)

Inputs

Legend: Mandatory input Optional input

Prefix	Label	Type	Default value
	<span style="background-color: red; color: white; padding: 2px;">Photoscan dense cloud</span>	file	
	<span style="background-color: red; color: white; padding: 2px;">Window Size</span>	float	12
	<span style="background-color: red; color: white; padding: 2px;">Maximum standard deviation of soil height</span>	float	0.13
	<span style="background-color: red; color: white; padding: 2px;">Maximum number of clusters</span>	float	4
	<span style="background-color: red; color: white; padding: 2px;">Output folder</span>	folder	module_public_SoilHeight_UAVRGB_SFM

Outputs

Label	Type	Conditioning parameter	Value
<span style="background-color: blue; color: white; padding: 2px;">Soil Height Map</span>	file		\$(outputFolder)/SoilHeightMap.mat

Chemin vers dépôt docker  
intégration automatique

Choix parmi une liste de type de données  
Possibilité de donner des valeurs par  
défaut/manuelles ou issues d'autres modules

Soil Height from RGB UAV from SfM algorithm

Duplicate
Process dataset
Publish
Archive
Delete

Details Workflow

The configuration of your macro is complete. You can now publish and use it.

For more details, please refer to the online guide: [create a macro](#).



Module: [Phenoscript - Process](#)

Description:

Le module process permet d'exécuter de manière automatique un projet de photogrammétrie, Agisoft Photoscan dans notre cas.

Input dataset:

RGB Images

Add a data:

Dense Cloud (L1 - Drone)

Inputs	Source	Value
Camera profile file *	Phenoscript - Camera Profile -> Camera profile file	
Drone height *	RGB Images -> Drone height	
Export dense cloud	Manual	true
Export orthomosaics	Default value	false
GCP coordinates *	RGB Images -> GCP coordinates	
Photoscan settings	Manual	photoscan_settings_ms_dense_cloud.json Select file... <span>Browse ...</span>
Registered drone images *	Phenoscript - Filtration -> Filtered drone images	

Processors

0 %

0.0 / 0.0 CPUs

Memory

0 %

0.0 kB / 0.0 kB

4P Disk

42 %

1.8 TB / 4.3 TB

Services

- iRODS
- Cromwell
- Docker Swarm
- VM - Photoscan 1
- VM - Node 1
- VM - Node 2

PHIS Instances

- PHIS Clermont-Ferrand
- PHIS Montpellier
- PHIS Toulouse
- PHIS Dijon
- PHIS Ouzouer-le-Marché

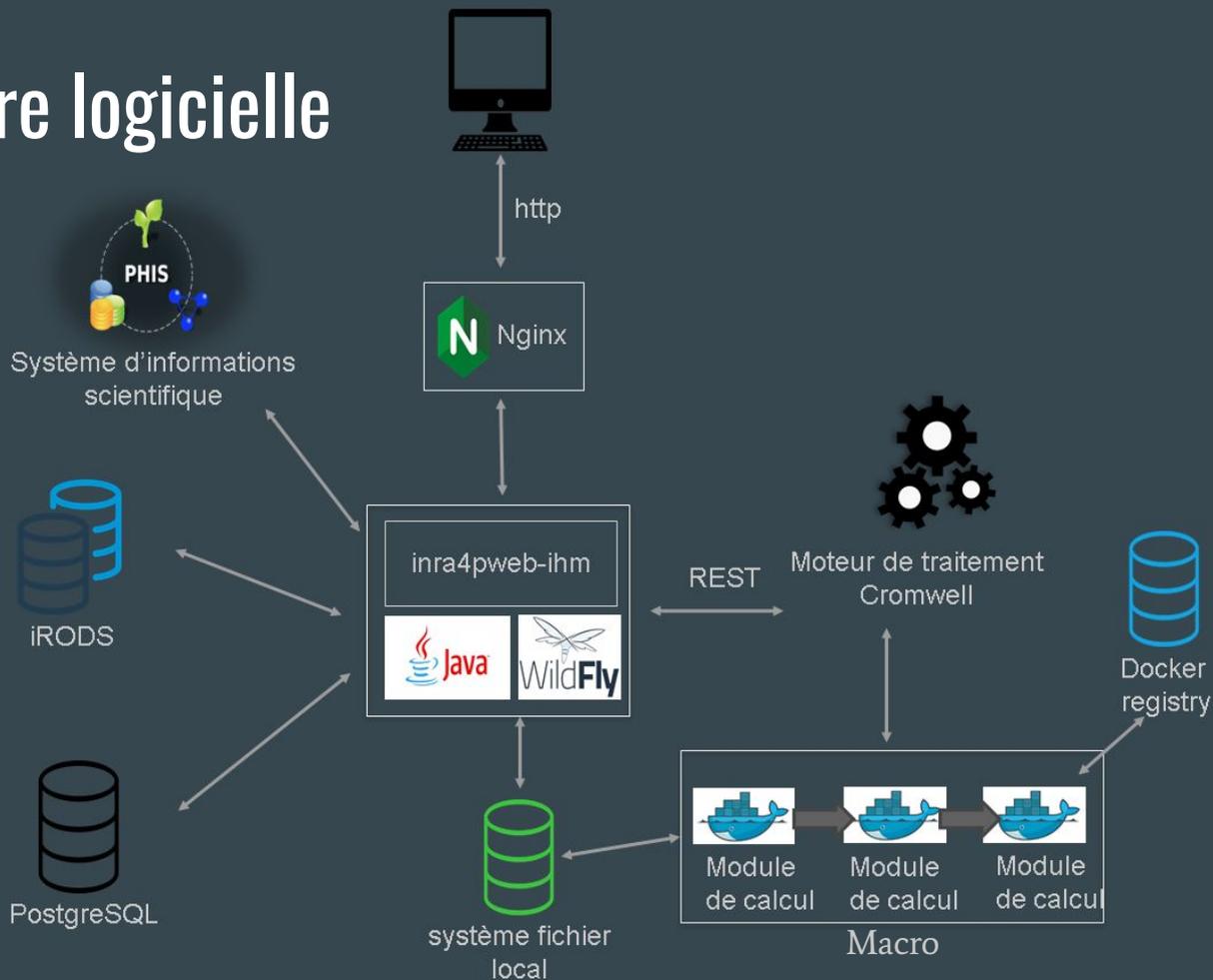
## Monitoring history

Processes

Storage

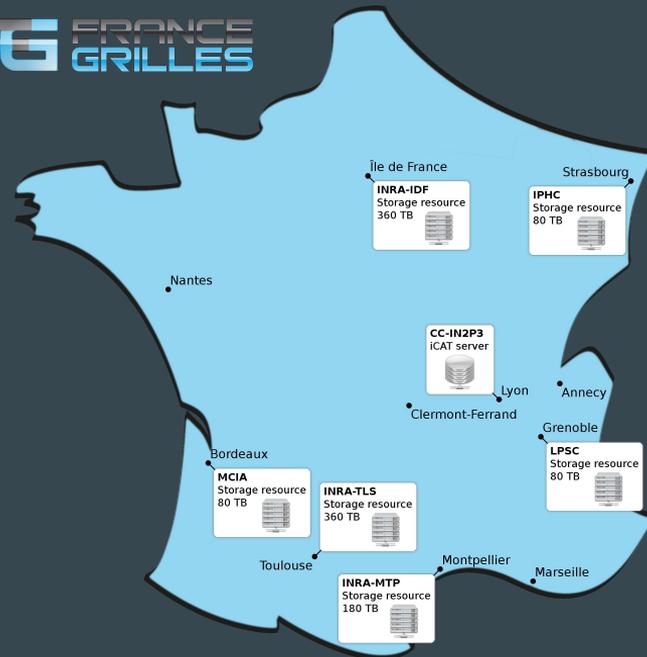
Macro	Measuring session	Installation	Start date	Duration	Status / Progress	Actions
<input type="text" value="Name"/>	<input type="text" value="DD/MM/YYYY - DD/MM/YYYY"/>	<input type="text" value="All"/>	<input type="text" value="DD/MM/YYYY - DD/MM/YYYY"/>		<input type="text" value="All"/>	
<a href="#">Pré Traitement Drone Multispectral v1.1</a>	31/07/2019 09:47	Toulouse	20/08/2019 09:00	261 h 56 mn 35 s	Complete <span style="border: 1px solid green; padding: 2px;">8/8</span>	
<a href="#">Pré Traitement Drone Multispectral v1.1</a>	31/07/2019 09:47	Toulouse	13/08/2019 13:41	438 h 34 mn 55 s	Complete <span style="border: 1px solid green; padding: 2px;">8/8</span>	
<a href="#">Extraction HDF5 Phénomobile</a>	04/06/2019 11:47	Montpellier	13/06/2019 01:35	2049 h 18 mn 19 s	Cancelled <span style="border: 1px solid gray; padding: 2px;">1/3</span>	

# Architecture logicielle



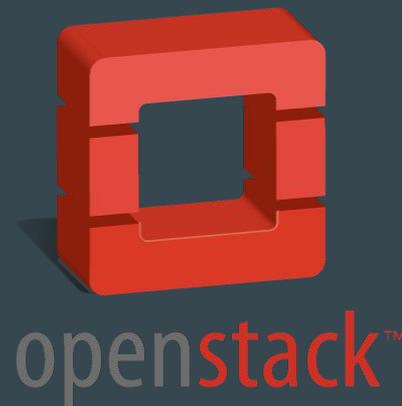
# Infrastructure de stockage

- plateforme mutualisée **FranceGrilles - iRODS**
- **stockage distribué**
- **réplication automatique** sur 3 sites géographiques (DC Toulouse; DC IDF; CINES Montpellier)



# Infrastructure de calcul

- service **FranceGrilles-Cloud**
- 6 machines virtuelles
- CentOS 7, jusqu'à 16VCPU, 64Go de RAM



# Création d'une module

- Un module de calcul peut être écrit dans n'importe quel langage (Java, C, C++, Python, etc.) mais doit suivre les conditions suivantes:
  - être embarqué dans une **image Docker** incluant tout l'environnement d'exécution
  - **exécutable en ligne de commande** sans intervention de l'utilisateur
  - **paramètres** (déclarés dans 4P) doivent être **fournis en arguments**
  -

```
CLI Example: docker run -v <host>:<mount_point> -w <mount_point> <module_name>:<tag> [arguments]
```

- **Publié sur un dépôt git** (Gitlab, Bitbucket, Github, etc.)
- Intégration continue avec Jenkins et Apache Groovy



# Création d'une module

```
1  #!/usr/bin/env groovy
2
3  node {
4  >   stage('Checkout') { ...
16   }
17
18  >   stage('Docker build') { ...
21   }
22
23  >   stage('Docker check') { ...
25   }
26
27  >   stage('Parsing version') { ...
46   }
47
48  >   stage('Docker push') { ...
67   }
68  }
69
```

Fichier Groovy

# Exécution d'un workflow

- moteur d'exécution **CromWell** développé par le Broad institute
- Lorsque l'utilisateur crée une macro, l'application 4P génère un fichier WDL (Workflow Description Language) décrivant les modules et leur enchaînement ainsi qu'un fichier JSON décrivant les paramètres d'entrées.

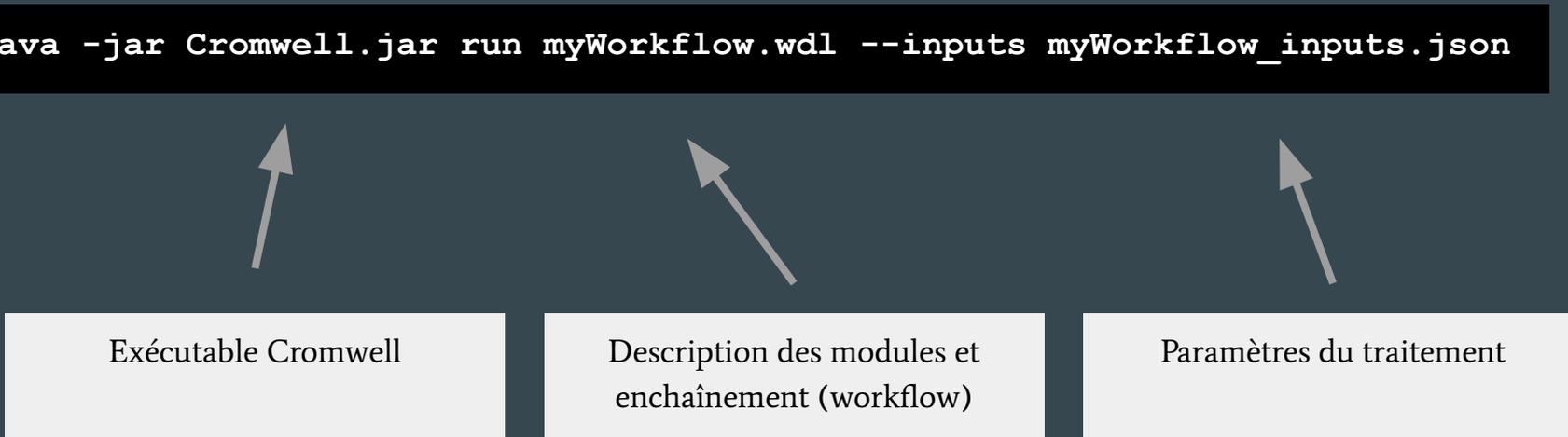


<https://software.broadinstitute.org/wdl/>

# Exécution d'un workflow

- Ce fichier est transmis à **CromWell** qui exécute le traitement

```
java -jar Cromwell.jar run myWorkflow.wdl --inputs myWorkflow_inputs.json
```



Exécutable Cromwell

Description des modules et  
enchaînement (workflow)

Paramètres du traitement

# Exécution d'un workflow

```
workflow macro_138 {
  String workingPath ...
  call logTaskStart as logTaskStart_importData {
    input: description = "import data from iRODS" }

  # Begin parallelization
  scatter (mono_parameter n inputPhenomobileFiles) {
    call logTaskStart_module_public_PhenomobileExtraction {...}
    call module_public_PhenomobileExtraction { ...}
  } # End of parallelization

  task logTaskStart {
    String description ...
    command <<<
      echo "{\"date\" : \"`date
'+%s%3N' `\", \"level\" : \"INFO\", \"message\" : \"log.process.task.start
\", \"parameters\" : [\"${description}\"]}"
    >>>
    output {
      String out = "out"
    }
  }
}
```

Description du workflow  
(macro)

Déclaration des modules

exemple de fichier WDL

# Soumission d'une anomalie

Signaler une anomalie sur le traitement

Vous êtes sur le point de signaler une anomalie sur le traitement actuel. Veuillez préciser la nature du problème rencontré. Les logs du traitement seront automatiquement joints à la demande.

Titre \* :

Description \* :

Sévérité :  
Mineure

Pièces jointes :  
Sélectionner un ou plusieurs fichiers... Parcourir...

Annuler Envoyer

API REST  
gitlab

Open Opened il y a 2 mois by Application 4p

## plantage traitement complet multispectral

plantage dans le module de photogrammetrie semble-t-il.

Reported by: Philippe Burger (philippe.burger@inra.fr)  
Data group: 2018\_Sunrise\_18ZM01, 2018\_Sunrise\_18ZM02 28/06/2019 11:26 L1/L2-DRONE  
Macro: Traitement Drone Complet Multispectral v1.1

logs.txt

0 0

Write Aperçu

Write a comment or drag your files here...

Markdown and quick actions are supported

4P - interface soumission d'une anomalie sur un traitement

ouverture d'un ticket sur  
forgemia.inra.fr (gitlab)



# Monitoring

- Basé sur le logiciel Prometheus
- Sur chaque VM une tâche tourne et envoie le résultat à la machine manager



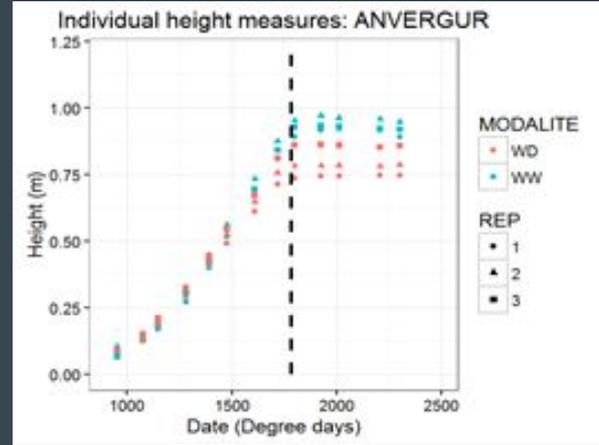
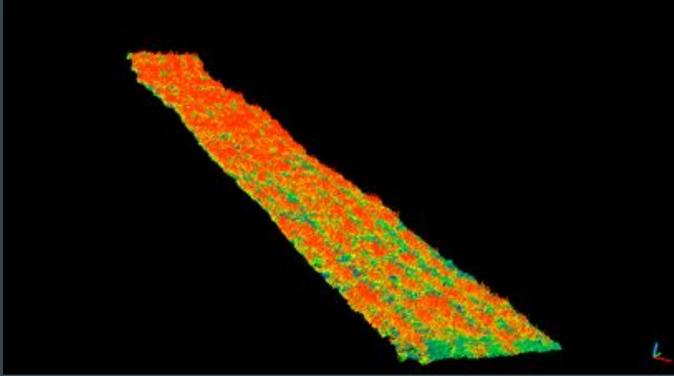
# Intégration avec le SI PHIS

- appel à des services webs REST basé sur l'API Swagger
- Web services implémentent la Breeding API (standard pour phénotypage)
- Récupérer les droits des utilisateurs, type de capteur (vecteur), type de données (session d'acquisition)

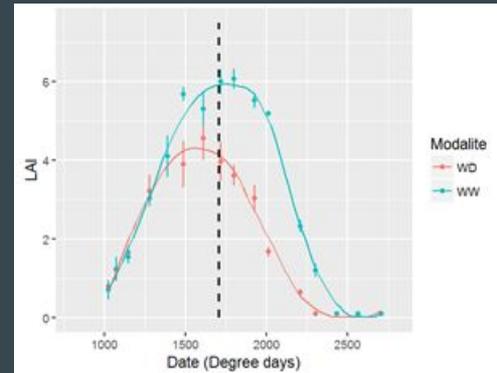
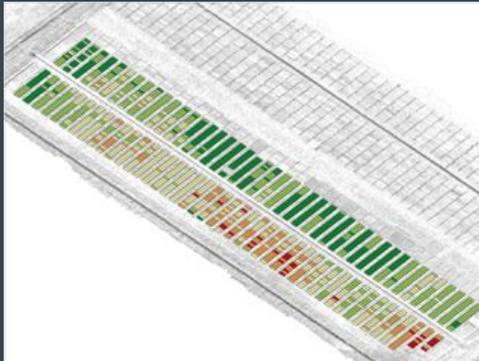


<http://www.phis.inra.fr/>

# Exemples concrets



**LIDAR**  
Suivi hauteur de couvert  
(S.Madec, S. Thomas)



**Caméra multispectrale**  
Suivi d'indice foliaire  
Caractérisation spatiale  
& temporelle

# Perspectives

- développer de nouvelles macros (données phéno-mobile, ré-encodage modules matlab, passage en python)
- renforcer intégration avec le SI scientifique PHIS (récupération des variables, publication des traitements)
- optimiser l'exécution des workflows (instanciation des VMs ou containers à la volée)

# Conclusion

- Fonctionnalités ont été implémentées
- Bonne communication entre les partenaires du projet notamment avec le prestataire informatique qui a su intégrer de multiples technologies
- Support fourni par France Grilles a permis de déployer rapidement la plateforme

# Remerciements



Questions?