
Programme de Formation

ImageJ/Fiji : concevoir et programmer des macros pour automatiser des workflows d'analyse d'images

Organisation

Durée : 21 heures

Mode d'organisation : Présentiel

Contenu pédagogique



Public visé

Cette formation s'adresse aux chercheurs, enseignants-chercheurs, ingénieurs, techniciens, doctorants et personnels R&D souhaitant automatiser des traitements ou des analyses d'images au moyen du langage macro ImageJ/Fiji.

Elle concerne les professionnels issus de domaines variés tels que les sciences de la vie, les sciences des matériaux, les géosciences, les sciences de l'environnement, la physique, la chimie, ainsi que tout autre domaine produisant des données d'imagerie.

Cette formation s'adresse en priorité aux utilisateurs disposant déjà d'une pratique courante d'ImageJ/Fiji et souhaitant aller au-delà de l'utilisation des fonctionnalités existantes en développant leurs propres outils d'automatisation, afin de construire des workflows d'analyse reproductibles et adaptés à leurs problématiques scientifiques.



Objectifs pédagogiques

- Comprendre les concepts fondamentaux de la programmation
- Développer des briques fonctionnelles en langage macro
- Concevoir un programme modulaire et réutilisable
- Automatiser un Workflow d'analyse d'images
- Évaluer la qualité d'un développement



Description

Philosophie pédagogique

La formation est construite selon une progression pédagogique itérative visant à développer progressivement l'autonomie des participants dans la conception et le développement de macros d'automatisation.

Activer → Découvrir → Formaliser → Appliquer → Réinvestir → Prendre du recul

Chaque séquence pédagogique suit une organisation identique, permettant d'acquérir non seulement les **concepts du langage macro ImageJ/Fiji**, mais également une **méthode de développement réutilisable** pour résoudre des problématiques scientifiques variées.

Contrairement à une approche consistant à présenter successivement les différentes instructions du langage, la formation repose sur une **pédagogie par projets**. Chaque macro développée répond à un



objectif concret, facilement identifiable par les participants, principalement orienté vers des problématiques de représentation graphique. Ce choix pédagogique permet de dissocier l'apprentissage de la programmation des difficultés liées au traitement ou à l'analyse d'images, afin que les participants puissent concentrer leur réflexion sur la **démarche algorithmique**.

Chaque projet débute par l'**observation de la macro finalisée**. Les participants identifient ensuite collectivement les **différentes étapes nécessaires à sa réalisation**. Cette analyse conduit à la **construction de l'algorithme**, qui constitue la trame de la macro. La formation distingue volontairement deux niveaux de réflexion : **concevoir l'algorithme** puis le **traduire en langage macro**. Les participants apprennent ainsi à **raisonner avant de coder**, en développant d'abord une solution conceptuelle qu'ils implémentent ensuite progressivement. Chaque étape est développée sous la forme d'une **macro autonome**, testée et validée avant d'être encapsulée dans une fonction puis intégrée au programme principal. Les concepts de programmation (variables, fonctions, boucles, conditionnels, tableaux, chaînes de caractères...) sont ainsi introduits au moment où ils deviennent nécessaires à la résolution d'un problème concret.

L'objectif n'est donc pas d'apprendre un langage de programmation de manière exhaustive, mais d'**acquérir une méthode de développement** permettant aux participants de :

Comprendre les concepts fondamentaux de la programmation :

- Comprendre les principes de l'algorithmique et leur traduction en langage macro ImageJ/Fiji.
- Identifier les principaux types d'instructions et leurs domaines d'utilisation.
- Comprendre le rôle des variables, chaînes de caractères, tableaux, fonctions et structures de contrôle.

Développer des briques fonctionnelles en langage macro :

- Manipuler les principales instructions du langage macro ImageJ/Fiji.
- Écrire des fonctions avec ou sans arguments et valeurs de retour.
- Tester et valider chaque fonctionnalité développée.

Concevoir un programme modulaire et réutilisable :

- Décomposer une problématique en tâches élémentaires.
- Organiser un programme sous forme de fonctions réutilisables.
- Assembler les différentes briques dans une macro documentée.
- Construire un toolset regroupant plusieurs outils complémentaires.

Automatiser un workflow d'analyse d'images :

- Traduire une problématique scientifique en étapes automatisables.
- Développer une macro répondant à un besoin d'analyse ou de représentation.
- Adapter cette démarche à de nouvelles problématiques.

Évaluer la qualité d'un développement :

- Tester une macro sur plusieurs jeux de données.
- Identifier les cas particuliers et les situations générant des erreurs.
- Améliorer le programme par une gestion adaptée des exceptions.
- Porter un regard critique sur les solutions développées afin d'en améliorer la lisibilité, la robustesse et la réutilisabilité du code.

Cette progression est répétée tout au long de la formation afin de développer simultanément la **maîtrise du langage macro**, les **bonnes pratiques de développement** et la confiance des participants dans leur **capacité à concevoir** leurs propres outils d'automatisation. La troisième journée constitue l'aboutissement de cette démarche en appliquant la méthode acquise à des problématiques réelles proposées par les participants.

Progression pédagogique d'une séquence :

Étape	Finalité pédagogique	Activités du formateur	Activités des participants
Activer	Consolider les acquis	Anime un réveil pédagogique, rappelle les notions mobilisées par le projet du jour	Répondent, reformulent, établissent les liens avec les séquences précédentes
Découvrir	Analyser une	Présente la macro finalisée et	Observent le résultat attendu,

	problématique	anime le brainstorming permettant d'identifier les étapes de l'algorithme	proposent les étapes, décomposent le problème
Formaliser	Structurer l'algorithme	Formalise l'algorithme, introduit les concepts du langage devenus nécessaires	Reformulent l'algorithme, identifient les briques fonctionnelles
Appliquer	Développer une solution	Accompagne le développement, valide les tests et introduit les bonnes pratiques	Programment, testent, corrigent, encapsulent les fonctions
Réinvestir	Construire un programme complet	Guide l'intégration des différentes briques fonctionnelles	Assemblent les fonctions, construisent la macro principale, documentent le code
Prendre du recul	Développer l'esprit critique	Anime le débriefing technique et méthodologique	Analysent la robustesse du programme, identifient les limites, proposent des améliorations



Prérequis

Niveau d'études : aucun niveau de diplôme spécifique n'est exigé.

Connaissances : disposer d'une pratique courante d'ImageJ/Fiji et être amené à traiter ou analyser des images scientifiques dans son activité professionnelle. Une première expérience de la construction de workflows d'analyse est recommandée. Aucune connaissance préalable en programmation n'est requise.

Équipement : ordinateur portable personnel (Windows, macOS ou Linux) disposant des droits permettant l'installation de logiciels.

Logiciels : ImageJ/Fiji (la procédure d'installation est fournie avant la formation).



Modalités pédagogiques

La formation est assurée en **co-animation permanente** par deux formateurs partageant **plus de vingt ans d'expérience commune** dans l'enseignement du traitement et de l'analyse d'images. Cette collaboration de longue date permet une animation particulièrement fluide : les interventions sont naturellement complétées, illustrées ou reformulées par le second formateur dès que cela facilite la compréhension d'un concept ou répond à une question soulevée par le groupe.

Lors des phases de développement, l'organisation repose sur une **répartition dynamique des rôles**. Un formateur pilote la **progression collective** en explicitant la démarche de conception, tandis que le second circule parmi les participants pour **répondre aux difficultés individuelles, proposer des pistes de réflexion ou accompagner le débogage**, sans interrompre le rythme de la séance. Les rôles s'inversent naturellement au fil de la formation, en fonction des thématiques abordées et des expertises de chacun.

Cette modalité s'inspire des principes du **pair programming**, une pratique largement utilisée en développement logiciel dans laquelle deux développeurs travaillent simultanément sur un même programme : l'un conduit la réalisation tandis que l'autre observe, questionne, propose des alternatives et contribue à la qualité du raisonnement. Transposée au contexte pédagogique, cette approche permet aux participants d'assister à la **construction d'une réflexion algorithmique**, d'observer les **échanges entre deux développeurs expérimentés** et de comprendre qu'une même problématique **peut souvent être résolue de plusieurs façons**. Au-delà de l'apprentissage du langage macro, ils développent ainsi leur capacité à analyser un problème, comparer différentes stratégies d'implémentation et justifier leurs choix techniques.



Moyens et supports pédagogiques

Ressources pédagogiques :

Les participants disposent d'un **livret pédagogique** regroupant les supports de cours, les fiches d'activités, les macros, fonctions et bibliothèques développées au cours de la formation, les principales instructions du langage macro ImageJ/Fiji ainsi que les bonnes pratiques de conception et de débogage. Ce document constitue une ressource de référence facilitant le réinvestissement des acquis et le développement de nouvelles macros.

Équipements et préparation de la formation :

Les travaux pratiques sont réalisés sur les **ordinateurs personnels des participants**, préalablement configurés.

Trois semaines avant la formation, un premier courriel précise les modalités pratiques et les prérequis techniques (installation de Fiji/ImageJ, téléchargement des jeux de données, vérification de la connexion Internet). Les participants sont invités à transmettre **une ou plusieurs problématiques d'automatisation, accompagnées si possible des images correspondantes**, susceptibles d'être exploitées lors de la dernière journée de formation. Un second courriel, envoyé une semaine avant la session, diffuse le jeu de données consolidé et les dernières informations pratiques.

Cette préparation permet de sélectionner des **cas d'usage représentatifs** des besoins du groupe et favorise un **transfert immédiat des apprentissages vers les situations professionnelles** des participants.



Modalités d'évaluation et de suivi

L'évaluation repose sur un **suivi individualisé et continu** tout au long des activités de programmation. Les échanges entre les participants et les formateurs permettent de vérifier progressivement la compréhension des concepts, la capacité à analyser une problématique, à construire un algorithme, puis à le traduire en une macro fonctionnelle, lisible et correctement documentée.

Les différents projets développés au cours de la formation constituent les principaux temps d'**évaluation formative**. Ils permettent d'apprécier la capacité des participants à décomposer un problème complexe en fonctions élémentaires, à développer et tester chacune d'elles de manière indépendante, à intégrer ces briques dans un programme cohérent et à porter un regard critique sur la robustesse, la lisibilité et la maintenabilité du code produit.



Informations sur l'admission

L'admission à cette formation ne fait l'objet d'aucun examen, test ou sélection préalable ; l'inscription est validée après réception du dossier complet et confirmation par l'organisme de formation.



Informations sur l'accessibilité

Notre organisme s'engage à garantir l'accessibilité de ses formations à distance et en présentiel aux personnes en situation de handicap. Un référent handicap est mobilisable afin d'analyser les besoins spécifiques et de mettre en place, lorsque cela est possible, les adaptations pédagogiques, techniques ou organisationnelles nécessaires.