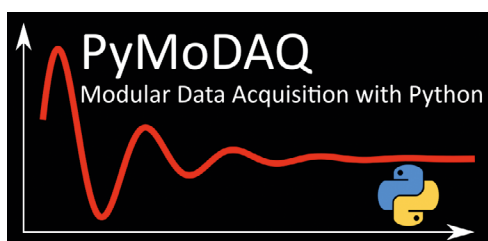


Le logiciel PyMoDAQ : acquisition, analyse et diffusion de données

Sébastien J. WEBER*

CEMES-CNRS, Université de Toulouse, 29 rue Jeanne Marvig, 31055 Toulouse, France

* sebastien.weber@cemes.fr



La donnée, depuis son acquisition dans le laboratoire jusqu'à sa publication en passant par son analyse, joue un rôle central dans la vie des scientifiques. Cet article explicite la notion de données expérimentales FAIR et présente PyMoDAQ un logiciel open-source collaboratif aidant le scientifique dans chacune de ces étapes.

<https://doi.org/10.1051/photon/20412925>

Article publié en accès libre sous les conditions définies par la licence Creative Commons Attribution License CC-BY (<https://creativecommons.org/licenses/by/4.0>), qui autorise sans restrictions l'utilisation, la diffusion, et la reproduction sur quelque support que ce soit, sous réserve de citation correcte de la publication originale.

L'acquisition de données concerne l'ensemble des disciplines expérimentales (imagerie, spectroscopies, métrologie...) possédant chacune des instruments dédiés et des techniques de mesures adaptées. Il existe une multitude de logiciels permettant l'acquisition de ces données. Ces logiciels sont soit développés pour répondre à un besoin expérimental spécifique, soit sont fournis par le fabricant de l'instrument utilisé. Tout va bien jusqu'à ce que le dispositif expérimental évolue et que d'autres instruments y soient ajoutés afin d'obtenir plus d'informations. Le plus souvent, le logiciel doit donc être complété voire réécrit si sa structure initiale ne permet pas d'évolution. A l'ère de la science ouverte, le concept de données FAIR (voir encart) joue un rôle central dans le développement de logiciels d'acquisition de données. Cet article présente les problématiques à résoudre pour avoir un

système d'acquisition de données applicable à tout dispositif expérimental et rendre les données enregistrées aussi FAIR que possible. Nous montrerons en quoi la suite logicielle open-source PyMoDAQ [1] « Modular Data Acquisition with Python » est une solution face à ces défis.

Du générique vers le spécifique, un concept pour l'acquisition des données

Que ce soit sur de grands instruments (synchrotron, laser à électrons libres, ...) ou sur une expérience de laboratoire (situation sur laquelle on se restreindra dans la discussion), le nombre d'instruments utilisés pour caractériser le processus physique d'intérêt évolue constamment. Prenons comme exemple,

une expérience classique de spectroscopie femtoseconde dans des nanomatériaux, voir figure 1. Cette expérience vise à caractériser la dynamique temporelle d'absorption de lumière de films minces métalliques [2] à une échelle de temps femtoseconde ou picoseconde. Les instruments de base nécessaires à intégrer au logiciel d'acquisition sont ici la platine de translation qui va retarder l'impulsion sonde par rapport à l'impulsion pompe et l'amplificateur à détection synchrone qui va collecter le signal de sonde transmis à la fréquence de modulation du modulateur acousto-optique (AOM). Le logiciel associé doit donc enregistrer ce signal, scalaire, en fonction d'un paramètre variable ici le délai temporel pompe-sonde *via* la position de la platine. Mais pour compléter cette mesure, on va vite avoir besoin de rajouter une caméra pour visualiser la superposition des faisceaux pompe et sonde, une mesure de ●●●

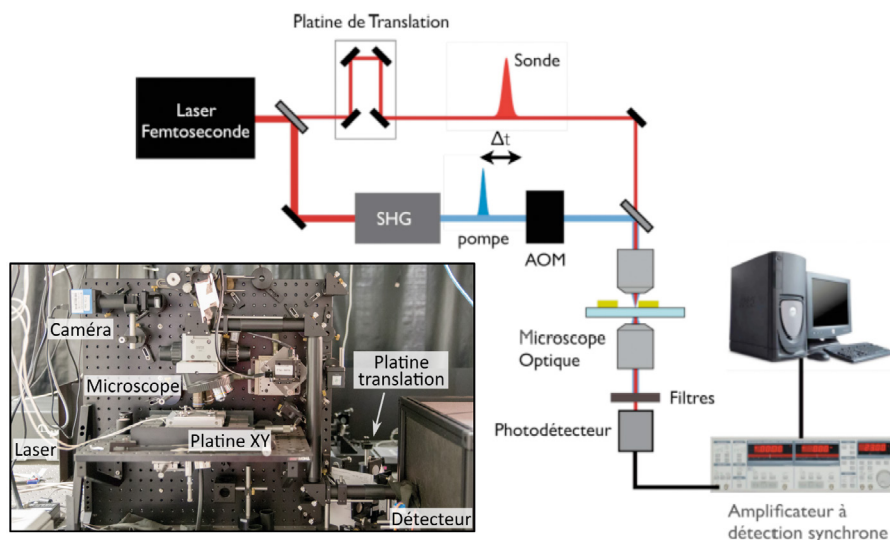


Figure 1. Dispositif expérimental d'une expérience pompe-sonde d'absorption transitoire femtoseconde. Une source laser femtoseconde est séparée en deux impulsions dont le retard est contrôlé par une platine de translation. La première impulsion, dite pompe, met le matériau dans un état excité. La seconde impulsion sonde le système à un retard donné en mesurant l'amplitude de la transmission à travers le matériau. L'expérience est répétée pour différents retards.

puissance du laser, le contrôle de la position de l'échantillon sous le faisceau focalisé etc. On s'aperçoit vite qu'un logiciel spécifique écrit pour l'expérience initiale va devoir grandement se complexifier et de nombreuses problématiques vont alors se poser comme l'ajout de nouveaux instruments, leur remplacement en cas de panne, la gestion de l'affichage des données... L'affichage devait initialement gérer des scalaires en fonction du délai (donc graphique 1D) mais si l'expérimentateur décide d'enregistrer le spectre en fonction de la position XY de l'échantillon dans le microscope, l'affichage devra gérer des données 3D. L'affichage doit donc s'adapter à la dimensionnalité des données et du scan...

Le développement de logiciels «maisons» et spécifiques à une expérience souffre d'un autre facteur déterminant: l'obsolescence. Elle est liée à plusieurs facteurs que sont : la licence du langage utilisé, la compatibilité avec le/les système d'exploitation, le remplacement ou l'ajout d'un instrument et surtout la fuite des cerveaux. La réalité est que beaucoup de ces logiciels « maisons » sont écrits par de jeunes chercheurs qui, particulièrement dans le milieu académique, sont amenés à partir après quelques années et emportent avec eux le savoir et la maîtrise du logiciel. Enfin, il y a la notion de facilité

d'utilisation. En effet pour qu'un logiciel soit utilisé, il doit avoir une documentation claire et exhaustive, une interface graphique simple mais complète et une gestion des données performante. Pour obtenir un dispositif d'acquisition de données et d'orchestration expérimentale pérenne, il est donc vital de construire ou de choisir un logiciel modulaire et athématique.

Versatilité

La versatilité ou modularité est vitale dès qu'on utilise plus d'un instrument dans un dispositif expérimental. Il s'agit donc de bénéficier d'une structure programmatique qui permette la définition du nombre et du type d'instrument sans limite intrinsèque ainsi que d'une bibliothèque d'instruments compatibles avec cette structure et dans laquelle on puisse puiser pour ajouter ou remplacer un instrument. Dans le cadre d'expériences utilisant

un microscope optique on peut citer le logiciel Micro-Manager: « [...] *μManager works with microscopes from all four major manufacturers (Leica, Nikon, Olympus and Zeiss), most scientific-grade cameras and many peripherals used in microscope imaging* ». C'est un projet open-source qui contient une grande variété d'instruments dans sa bibliothèque.

Logiciel agnostique

Dans l'exemple précédent, micro-manager répond bien au critère de versatilité mais sous réserve que l'on souhaite faire une expérience autour d'un microscope. Les concepteurs ont choisi de s'adresser spécifiquement à la communauté des microscopistes avec un réel succès mais une limitation à cette thématique. Le fait que les logiciels d'acquisition soient développés autour d'une thématique ou application donnée s'explique par deux raisons : la première est qu'il est naturel d'aller directement vers la résolution de sa problématique expérimentale. Le temps qu'on y passe est déjà suffisamment important pour ne pas en investir plus dans une solution logicielle générique qui ne sera jamais utilisée dans son laboratoire. La seconde est liée aux interfaces utilisateurs. Si l'on souhaite offrir une telle interface aux utilisateurs du logiciel¹ pour leur confort et leur habitude d'utiliser ce type d'interface² alors il est relativement aisé de construire une interface dans un but (thématique ou technique expérimentale) précis mais qui aura alors tendance à être peu adaptable.

Pour résoudre à la fois le problème de modularité instrumentale, d'interface graphique et de spécificité, PyMoDAQ a pris le parti de créer deux interfaces graphiques génériques (modules de contrôle) pour

¹ Le parti pris inverse serait de n'offrir qu'une possibilité d'acquisition par l'écriture de scripts de code informatique

² Microsoft Windows est passé par là...

ses instruments : d'une part pour les détecteurs (généralisant des données et appelés *DAQ_Viewer*) et d'autre part pour le contrôle des paramètres que l'on souhaite varier dans une expérience (appelés *DAQ_Move*). Le nombre et type de ces modules de contrôle sont configurés pour chaque dispositif expérimental et ces derniers sont regroupés dans un tableau de bord « *Dashboard* », voir Figure 2. Parce que l'interface est identique pour tous les instruments d'un type donné, il est alors facile d'en charger autant qu'on veut et à tout moment d'en remplacer un. Cette solution permet le contrôle informatique complet du dispositif expérimental sans se soucier de la thématique scientifique ou de ce qu'on veut faire avec. La spécificité, si nécessaire, est obtenue en construisant des extensions au tableau de bord, qui vont être capable de prendre le contrôle programmatique des modules de contrôle. On peut ainsi facilement procéder à une expérience de type « scan » en chargeant l'extension *DAQ_Scan* qui va s'occuper de la sélection des instruments, de la façon de réaliser le scan (par exemple linéaire, aléatoire, etc.) et de sauvegarder les données d'une façon adaptée. Cette séparation permet le remplacement d'un instrument défectueux par un autre sans impacter l'extension. Il faut juste reconfigurer le tableau de bord. De nombreuses autres extensions existent et permettent de répondre à une très large variété de situations expérimentales. Cependant tout ceci n'est possible que si la bibliothèque d'instruments pilotables par PyMoDAQ est suffisamment complète. PyMoDAQ bénéficie déjà d'une bibliothèque conséquente d'instruments divers (mesures électriques, spectromètres, oscilloscopes, caméra, contrôleur de position, de rotation...) mais bénéficie surtout d'une riche documentation et de formations annuelles pour pouvoir ajouter ses propres instruments à la bibliothèque existante.

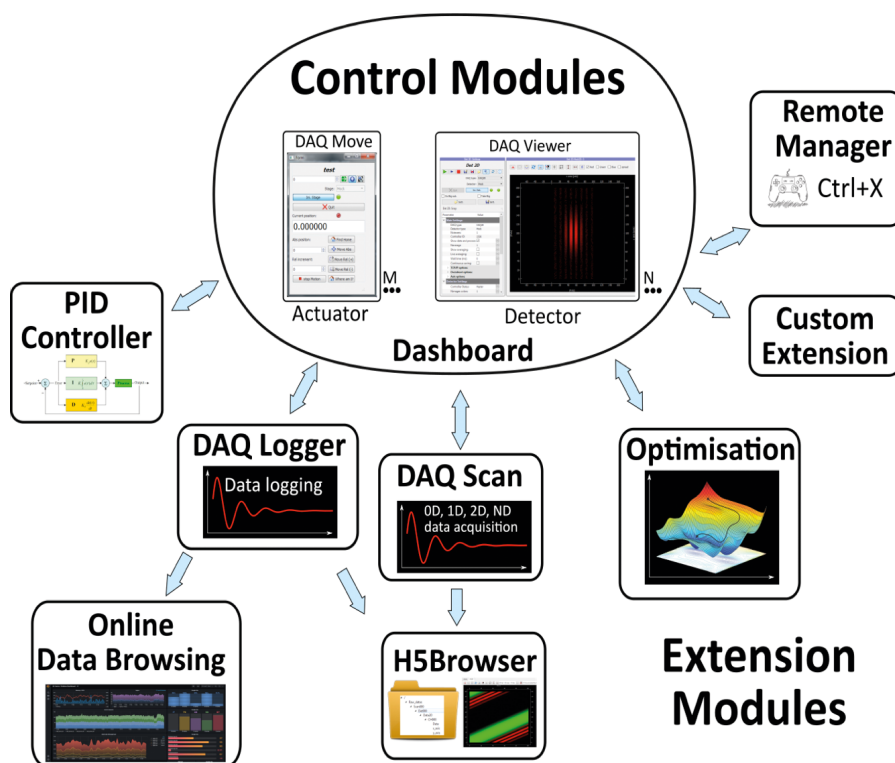
Obsolescence, open source et communauté

L'obsolescence des logiciels « maisons » est une perte nette de temps et de connaissance. La solution adoptée par PyMoDAQ tient dans la notion de projet open-source et développement collaboratif. Le choix de l'open-source, tant au niveau du langage informatique sélectionné (ici Python) et sa très grande communauté, qu'au niveau de la licence même de PyMoDAQ (licence libre dite permissive MIT) encourage les utilisateurs à s'approprier le logiciel. Chacun peut librement le télécharger, le tester, le commenter, le modifier, le compléter et enfin l'améliorer. La création d'une communauté des utilisateurs et des développeurs fait que la connaissance est partagée par plusieurs et donc préservée dans le temps.

Les données dans PyMoDAQ

La donnée est un concept central mais un peu flou dans le développement d'un logiciel d'acquisition. En effet, la donnée peut être simplement les valeurs numériques récupérées à partir d'un instrument de type détecteur. Mais si on considère les valeurs numériques issues d'un spectromètre optique, quel est leur intérêt scientifique si ces valeurs ne sont pas rapportées à un axe d'énergie ou de longueur d'onde ? Aucun ! Il faut donc y adjoindre un ensemble d'informations supplémentaires permettant de complètement caractériser la donnée numérique : ce sont les meta-données, l'ensemble des données numériques et des métadonnées constitue la « donnée » telle qu'on va la discuter dans la suite. On a tendance, et c'est bien normal, de ne penser qu'aux types de données qu'on produit dans son laboratoire sans forcément aller vers plus de généralité. Si on reprend l'exemple d'une expérience de microscopie, la donnée numérique est généralement une image à laquelle il faut associer

Figure 2. Diagramme du logiciel PyMoDAQ – Chaque instrument de l'expérience est représenté et contrôlé par un module de contrôle groupé dans un tableau de bord. Les tâches spécifiques sont alors réalisées par une extension orchestrant les modules de contrôle et leurs actions : acquisition ou modification d'un des paramètres variables.



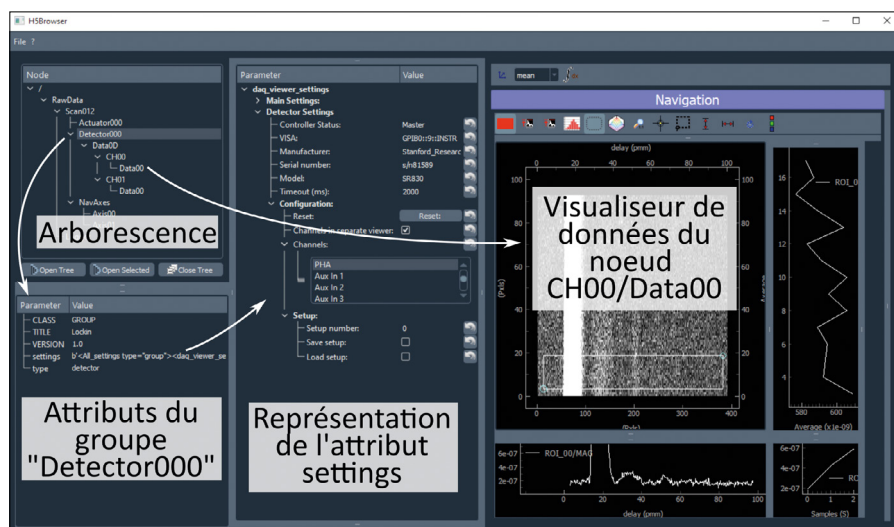


Figure 3. Exploration de donnée PyMoDAQ par l'extension H5Browser. La partie à gauche montre l'arborescence du contenu d'un fichier HDF5 obtenu par le scan de l'absorption transitoire femtoseconde d'un film d'or en fonction du délai pompe-sonde. Les attributs sont représentés en bas à gauche. Les attributs de type «settings» sont représentés au milieu, tels qu'ils étaient dans le détecteur utilisé. Les données sont représentées dans un «Data Viewer» spécifique. L'axe horizontal représente le délai pompe-sonde (déplacement de la platine en millimètre) et l'axe vertical l'acquisition de plusieurs scans. Les encarts en bas et à droite représente une extraction et moyennage partiel des données sélectionnées par le rectangle blanc (région d'intérêt).

des axes calibrés représentant la taille réelle de l'échantillon décrit par l'image. Mais si on considère la donnée comme cette image en fonction de la longueur d'onde de la lumière émise alors on obtient une donnée hyper-spectrale, un «data cube» de dimension trois. Le logiciel doit donc être capable de gérer de manière transparente aussi bien des données scalaires que des «data cube» complexes.

Les objets «données» de PyMoDAQ

Pour pouvoir décrire la très grande variété de données possible, PyMoDAQ a créé un écosystème dédié qui permet la description complète des données sous forme d'objets informatiques (valeurs numériques, axes éventuels, dimension, unités, barres d'erreurs, métadonnées diverses...), leur manipulation, leur sauvegarde (et à l'inverse leur chargement) ainsi que leur représentation graphique. Cette description générique simplifie l'écriture du logiciel d'acquisition mais peut aussi être utilisée pour analyser et manipuler les données issues de PyMoDAQ.

Visualisation des données

La visualisation des données durant leur acquisition est essentielle pour s'assurer que le dispositif expérimental fonctionne correctement³. Elle

permet aussi, à travers l'utilisation de régions d'intérêt et de l'intégration du signal dans cette région, de simplifier la représentation de la donnée brute afin de révéler un comportement anormal ou exotique signe d'un problème ou d'un futur prix Nobel⁴! Il est donc vital de pouvoir simplement visualiser ceci en direct sans devoir attendre l'analyse *a posteriori* des données collectées. PyMoDAQ possède donc des visualiseurs graphiques adaptés à la dimensionnalité des données. Ils peuvent être utilisés durant l'acquisition ou durant le post-traitement des données, voir Figure 3. Ils possèdent un ensemble de fonctionnalités permettant d'extraire rapidement des informations

sur les données affichées, d'en changer l'aspect ou encore d'y afficher des déviations standards.

La sauvegarde des données

PyMoDAQ bénéficie d'un écosystème complet d'objets informatiques décrivant les données et facilitant leur manipulation. Mais au final, ce n'est pas ce type d'objet que nous gardons sur nos ordinateurs mais des fichiers. PyMoDAQ s'est tourné très tôt vers un format de fichier binaire, compressible et organisé sous forme d'arborescence avec des groupes contenant d'autres groupes ou des tableaux de données. Il s'agit du format HDF5. C'est un type de fichier avec une empreinte sur le disque minimale et des accès en lecture et écriture sans devoir charger tout le contenu sur la mémoire vive. Outre ces avantages, chaque nœud de l'arborescence peut contenir un nombre infini d'attributs. Ces attributs permettent de stocker toutes les métadonnées d'un objet de données PyMoDAQ. Il a donc été possible de créer une relation univoque entre l'objet donnée et son stockage dans le fichier HDF5. PyMoDAQ fournit toutes les fonctionnalités pour passer de manière transparente de l'un à l'autre. De plus, les fichiers de données peuvent être ouverts par une des extensions de PyMoDAQ, le H5Browser afin de parcourir les données contenues dans le fichier, les représenter dans un visualiseur adapté et de visualiser les métadonnées. La Figure 3 montre l'interface graphique du H5Browser explorant des données obtenues grâce au dispositif expérimental de la Figure 1. Cette relation entre objet de donnée et fichier HDF5 permet l'interopérabilité des données car en utilisant PyMoDAQ et un fichier HDF5, on obtient alors un objet «donnée» python pouvant être traduit pour être compréhensible

³ Qui n'a jamais lancé une acquisition en oubliant de débloquer la source laser?

⁴ Bon ok sûrement un problème...

dans d'autres logiciels, par exemple d'analyse, ou exporté dans un autre format de fichier.

Les métadonnées pour la description et la réutilisation des données

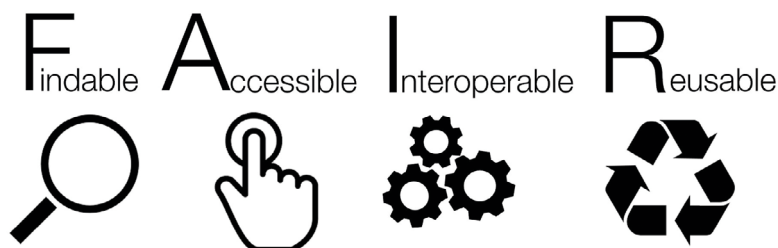
A ce stade, il manque encore un certain nombre de métadonnées afin de rendre les données réutilisables. Il faut en effet qu'elles contiennent les informations qui permettront de bien identifier ce qu'elles représentent et comment elles ont été produites, par exemple la description de la technique expérimentale utilisée. Ces deux points sont en général couverts par l'utilisation d'un cahier de laboratoire⁵ (papier ou électronique) qu'il faut pouvoir inclure totalement ou partiellement dans les métadonnées. Les cahiers de laboratoire électroniques, tel que eLabFTW, permettent la publication de ces informations et l'obtention d'un identifiant unique. L'inclusion de cet identifiant (et la possibilité d'obtenir le document associé) comme métadonnée est une solution pour répondre à cette problématique. Il faut aussi qu'on soit capable de les reproduire expérimentalement. Pour ce dernier point, il est donc vital d'enregistrer la configuration du ou des instruments mis en jeu lors de la sauvegarde des données. Ce dernier point est fait sous forme d'un attribut dans le fichier HDF5 contenant la représentation de tous les paramètres accessibles des instruments tels qu'ils le sont dans les modules de contrôle graphique du tableau de bord.

Trouvable et accessible

Pour ce qui est des concepts FAIR restant, c'est à dire trouvable et accessible, il faut pour ça s'appuyer sur des solutions externes de type entrepôts de données. Ce sont des services qui permettent de gérer les données collectées qui peuvent être installés localement ou à l'échelle d'une institution. Parmi les solutions existantes,

⁵ Plus connu sous son petit nom de cahier de « manip »

Le concept de donnée FAIR englobe les notions de production et de publication de données de manière à les rendre « Facile à trouver », « Accessible », « Interopérable » et « Réutilisable ». Ce principe est le prérequis d'une science ouverte, où les données, en particulier celles issues des institutions publiques doivent pouvoir être utiles à la société et donc en premier lieu être faciles à trouver et compréhensibles. L'interopérabilité vise à pouvoir facilement visualiser et utiliser les données dans divers logiciels et donc sa structure doit être bien documentée.



on peut citer DataVerse et NOMAD. Pour rendre les données trouvables ces services doivent analyser les données et en extraire les métadonnées nécessaires pour décrire l'expérience qui les a générées et éventuellement la technique expérimentale. NOMAD qui s'intéresse particulièrement aux sciences des matériaux est capable de classer des données comme issues de diffraction aux rayons X, spectroscopie électronique de pertes d'énergies etc... NOMAD peut alors fournir un référencement unique pour ces données qui pourront être citées et partagées. A noter qu'il est possible de définir un standard de donnée sur NOMAD et/ou un standard associé à une technique expérimentale ou thématique scientifique. Un projet est en cours pour rendre les données PyMoDAQ compatibles avec l'écosystème NOMAD. La structure des données de PyMoDAQ sera ainsi

« comprise » par NOMAD mais c'est aux communautés scientifiques qu'il revient de définir les métadonnées nécessaires pour une technique expérimentale ou une thématique scientifique donnée pour enfin rendre les données complètement FAIR.

Conclusion

La production de données FAIR joue un rôle de plus en plus important dans le contexte de la science ouverte. Il est donc nécessaire d'utiliser des systèmes d'acquisition de données simplifiant leur création. La possibilité de s'appuyer sur des solutions logicielles collaboratives et modulaires, telles que PyMoDAQ, offre un réel gain de temps et de qualité depuis la conception d'une expérience jusqu'à la publication de résultats. Essayer PyMoDAQ, c'est l'adopter! ●

RÉFÉRENCES

- [1] S. J. Weber, Review of Scientific Instruments **92**, 045104 (2021)
- [2] R. Marty *et al.*, Nano Lett. **11**, 3301 (2011)