Extraction d'informations sur les workflows scientifiques à partir de la littérature

Clémence Sebe*, Aurélie Névéol*, Sarah Cohen-Boulakia*, Alban Gaignard**

*Université Paris-Saclay, CNRS, LISN
prenom.nom@lisn.fr
**Université de Nantes, CNRS, INSERM Institut du Thorax
alban.gaignard@univ-nantes.fr

Résumé. Les workflows scientifiques offrent aux bioinformaticiens un cadre pour représenter, échanger et assurer la reproductibilité de leurs pipelines d'analyses. Ils sont décrits dans la littérature (texte) et/ou stockés dans des dépôts de workflows (code). Un enjeu majeur pour tendre vers une meilleure réutilisation des workflows par des tiers est de reconstruire le lien entre la documentation (texte) et l'implémentation (code) du workflow. A partir du texte intégral d'articles décrivant des workflows en anglais, nous proposons une méthode de modélisation et d'extraction d'informations des composants des workflows. Nous présentons un corpus de 24 articles annotés à l'aide d'un schéma comportant 16 entités et 10 relations. Nous utilisons ce corpus pour entraîner et évaluer des modèles statistiques d'extraction d'information sur les workflows. Nous montrons la faisabilité de la tâche comme première étape vers l'intégration d'information concernant les workflows issus de la littérature et des dépôts de workflows.

1 Introduction

Dans les sciences fortement génératrices de données, telles que la *bioinformatique*, les résultats scientifiques sont produits à l'aide de chaînes de traitement complexes pouvant prendre en entrée de très grandes quantités de données expérimentales. Ces chaînes peuvent être constituées de nombreuses étapes, faire appel à des outils bioinformatiques, et demander des temps de calcul considérables. Elles peuvent être implémentées à l'aide de scripts (Bash, Python...) qui pilotent l'execution des outils, et constituent le liant entre les données, les traitements, les outils et l'environnement d'exécution. Cependant, le développement et l'utilisation de tels scripts engendrent de multiples difficultés d'implémentation dans la conception et l'éxécution de la chaîne de traitement ainsi que des difficultés de maintenance et réutilisation par des tiers.

En réponse à ces problèmes, des efforts considérables ont été déployés ces vingt dernières années pour proposer aux bioinformaticiens des approches pour les guider vers une meilleure automatisation de leurs chaînes de traitement : les *systèmes de workflows scientifiques* (Cohen-Boulakia et al. (2017)). Deux systèmes sont aujourd'hui particulièrement utilisés en bioinformatique : Nextflow (Di Tommaso et al. (2017)) et Snakemake (Köster et Rahmann (2012)).

On assiste actuellement à deux phénomènes : (1) augmentation des articles scientifiques décrivant des workflows bioinformatiques qui sont sous une forme "descriptive", (étapes dé-

crites sans que l'on puisse les executer) et (2) augmentation des workflows bioinformatiques disponibles sous des dépots comme Github (3 000+ workflows Nextflow et Snakemake au 01/01/2022) qui sont sous une forme "programmatique" (implémentation disponible mais le manque de documentation les rendent difficilement réutilisables).

Un enjeu est donc le developpement d'un outil simple d'accès et automatique, permettant d'extraire des informations sur des workflows décrits dans la littérature pour non seulement documenter systématiquement les workflows présentés dans la littérature mais aussi accompagner leur recensement dans les dépôts de workflows.

C'est dans ce cadre que s'inscrit la présente publication qui introduit une méthodologie d'extraction d'informations concernant des workflows issus de la littérature. Les contributions de ce premier travail sont les suivantes : (1) nous proposons une représentation des composants d'un workflow à l'aide d'un schéma comportant 16 entités et 10 relations ; (2) nous proposons un corpus d'articles en anglais décrivant des workflows annotés à l'aide de cette représentation ; (3) nous montrons l'utilité de ce corpus pour l'extraction automatique d'informations concernant les workflows à l'aide de méthodes statistiques. Le code de ce projet est disponible sur Github (`https://github.com/ClemenceS/WorkflowExtractionNLP`).

2 Identification d'un corpus décrivant des workflows

Nous avons tout d'abord travaillé à l'élaboration d'un corpus décrivant des workflows bioinformatiques. Pour ce faire, nous avons interrogé deux bases de données : PubMed et PubMed-Central (NLM (2021)). Nous y avons effectué des requêtes strictes pour comprendre les articles présents. La requete suivante nous a permis de collecter un premier corpus : Recherche des articles dans PubMed Central comprenant le terme "nextflow" ou "snakemake" dans le résumé et pour lesquels il existe un lien vers Github dans le corps de l'article : *(nextflow[Abstract] OR snakemake[Abstract]) AND github[All Fields]*.

Elle a permis d'extraire quarante-huit articles décrivant des workflows sous le système de gestion Nextflow et quarante-neuf articles sous SnakeMake, soit un ensemble de quatre-vingt-dix-sept articles comprenant, chacun, en moyenne 2 200 mots (2 000 mots pour Nextflow et 2 400 mots pour Snakemake) (08/10/2022). Le nombre d'articles de ce type est en constante augmentation car ces deux systèmes de workflows sont très utilisés. Ces articles ont été extraits à l'aide de la librairie Python entrezpy (Buchmann et Holmes (2019), v2.1.3) et sont stockés dans un fichier XML. En analysant les articles, nous avons remarqué que les workflows bioinformatiques sont en général décrits dans une sous-partie des articles intitulée "Méthode" ou "Implémentation". Nous avons développé un script Python permettant de parser le fichier XML obtenu et d'extraire, pour chaque article, seulement son titre et la partie correspondante à la description du workflow et avons sauvegardé ces informations dans des fichiers texte.

3 Modélisation de la composition d'un workflow

Afin de proposer une description systématique de la composition d'un workflow bioinformatique, nous nous sommes appuyés sur des échanges avec des experts utilisateurs de workflows ainsi que sur la lecture de quelques articles du corpus. Globalement, les workflows sont formés d'étapes d'analyse de données et chaque étape contient un script qui peut faire appel

ou non à un outil bioinformatique. Pour qu'un workflow puisse s'exécuter il est important de bien garder la trace de l'environment d'exécution d'un workflow. Nous avons distingué deux grandes catégories de données : informations générales sur les workflows tel son environnement d'éxecution (système, langages de programmation utilisés dans les scripts, etc.) et informations sur le contenu du workflow lui-même, comme les outils bioinformatiques utilisés et les librairies. Nous trouvons également dans les articles des informations plus spécifiques sur la version et/ou la référence bibliographique décrivant le fonctionnement d'un outil ou d'une librairie utilisée, des descriptions et paramètres spécifiés pour certains éléments.

Nous avons également modélisé les relations entre ces types de données. La figure 1 présente la modélisation proposée, qui distingue des entités dites *globales* (les élements composant un workflow) et des entités dites *spécifiques* (les caractéristiques associées aux entités globales). Les flèches bleues représentent les relations possibles entre ces entités : toute entité *globale* peut être reliée à une entité *spécifique* de type "version" afin de caractériser la version du workflow ou de l'outil (Tool) décrit.

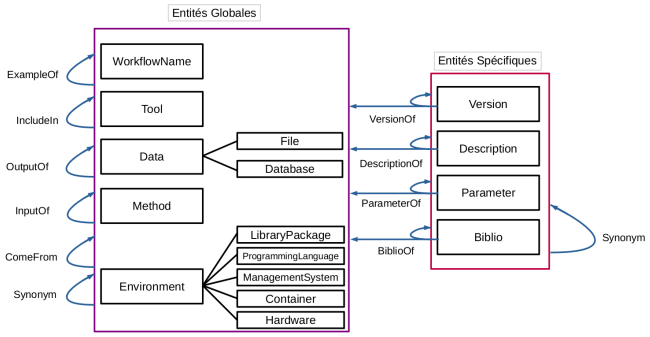

FIG. 1 – *Modélisation de la composition d'un workflow bioinformatique (entités et relations).*

4 Création d'un corpus annoté

Afin de valider ce système de représentation des informations et de créer une ressource permettant d'évaluer l'extraction de ce type d'information à partir d'articles de la littérature, nous avons élaboré un corpus annoté selon la méthode collaborative décrite par Fort (2016). D'abord, les entités et relations représentées sur la figure 1 ont été formalisées dans un schéma d'annotation. Nous avons ensuite conçu un guide d'annotation contenant une définition des entités et relations ainsi que des exemples d'occurrences en corpus illustrant les annotations cibles souhaitées. Nous introduisons les outils utilisés pour créer le corpus annoté (Matériel et méthodes) puis nous décrivons le corpus obtenu (Résultats).

4.1 Matériel et méthodes

Les annotations ont été réalisées avec le logiciel BRAT (Brat Rapid Annotation Tool, Stenetorp et al. (2012), v1.3 p1), dont les qualités sont attestées dans l'étude comparative réalisée

par Neves et Ševa (2019). En particulier, BRAT est un outil ergonomique qui permet l'utilisation de pré-annotations. La figure 2 présente l'annotation en entités et relations réalisée à l'aide de notre schéma sur un extrait d'article décrivant un workflow (PMID 35171290).

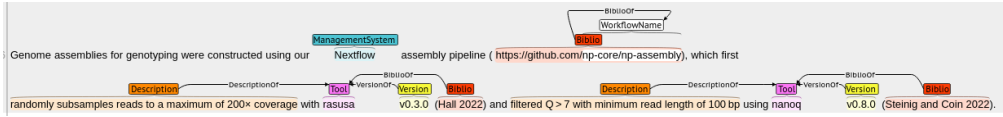

FIG. 2 – *Extrait du corpus annoté à l'aide du logiciel BRAT.*

Trois annotateurs ont participé à la campagne d'annotation : deux avaient une expertise en bioinformatique et une en traitement automatique de la langue biomédicale et développement de corpus annotés. Les annotations ont été réalisées en plusieurs étapes. Les annotateurs ont d'abord annoté des textes communs, puis des textes différents. L'accord inter-annotateur (IAA) a été calculé avec la F-mesure [1] à l'aide de BRAT-Eval (Verspoor et al. (2013), v0.0.2), un outil qui permet de comparer deux jeux d'annotations et d'analyser les divergences. L'évaluation de l'accord peut être strict (accord entre deux annotations si et seulement si deux portions de texte identiques sont annotées avec la même étiquette) ou relaché (accord entre deux annotations si deux portions de texte identiques ou avec un recouvrement sont annotées avec la même étiquette). Le score est compris entre 0 (aucun accord) et 1 (accord parfait).

4.2 Résultats

Étapes d'annotation. Lors de la première phase d'annotation, trois articles ont été annotés indépendamment. Le temps d'annotation était compris entre 1h30 et 3h. L'IAA mesuré variait entre 0.42 (modéré) et 0.88 (élevé) pour les entités et 0.16 (faible) et 0.79 (élevé) pour les relations selon les paires d'annotateurs. L'accord observé entre les relations dépend de l'accord observé entre les entités ; il est donc normal qu'il soit moins élevé sur les relations. Après une première discussion entre annotateurs en vue de créer un consensus, trois nouveaux articles ont été annotés. Les IAA ont été recalculés et les divergences discutées. Le temps d'annotation fut inchangé néanmoins les IAA étaient plus homogènes : de 0.44 à 0.63 pour les entités, de 0.35 à 0.47 pour les relations. Ces accords ne montrent pas l'augmentation globale espérée ce qui s'explique par le faible nombre d'articles annotés et le fait que deux des nouveaux articles comportaient des spécificités non vues et interprétées différemment par les annotateurs.

Nous avons calculé l'accord moyen des trois annotateurs avec le consensus ; il est de 0.70 ce qui est assez élevé et presage de bons résultats lors de l'utilisation d'un modèle d'extraction d'entités. Les scores par type d'entité montrent que, comme décrit dans Fort et al. (2012), certaines entités sont plus faciles à annoter (Biblio, 0.89) que d'autres (Description 0.57).

Lors des réunions de consensus, il a été estimé que le guide était bien compris par chacun. Nous avons décidé que la suite des annotations pourrait être réalisée sur des articles différents à l'aide de méthodes de pré-annotation permettant d'améliorer la consistance et la qualité des annotations (Névéol et al., 2011).

1. Nous renvoyons le lecteur à Artstein et Poesio (2008) pour une revue détaillée des mesures d'accord inter-annotateur et de leur contexte d'utilisation. Dans le cas de l'annotation en entités nommées, Grouin et al. (2011) montrent que la F-mesure et le κ peuvent être considérées comme équivalentes.

Outil de pré-annotation. Afin d'accélérer l'annotation, nous avons entrainé un modèle de reconnaissance d'entités à l'aide de l'outil NLStruct (Wajsbürt (2021)) sur les articles déjà annotés entièrement manuellement. Le modèle a ensuite été appliqué sur de nouveaux articles. Pour chacun, l'annotation automatique en entités obtenue a été corrigée par l'un des annotateurs et complétée par une annotation manuelle des relations. Le temps d'annotation manuelle a ainsi été divisé par deux.

4.3 Statistiques descriptives du corpus annoté

Nous obtenons un corpus de 24 articles annotés avec un total de 3993 entités et 1507 relations. Les tableaux 1 et 2 présentent la distribution de chaque type d'entité et de relation (respectivement) sur l'ensemble du corpus.

Entités	Occurences	Entités	Occurences
Tool	497	Environment	41
Version	135	Container	71
Description	521	ManagementSystem	96
Parameter	116	LibraryPackage	75
Biblio	572	ProgrammingLanguage	57
Data	616	Hardware	144
File	269	Method	262
Database	133	WorkflowName	388

TAB. 1 – *Nombre d'entités de chaque type annotées dans le corpus.*

Relations	Occurences	Relations	Occurences
VersionOf	130	InputOf	174
ParameterOf	38	OutputOf	94
DescriptionOf	480	ComeFrom	9
BiblioOf	428	ExampleOf	7
Synonym	55	IncludeIn	92

TAB. 2 – *Nombre de relations de chaque type annotées dans le corpus.*

5 Extraction d'entités nommées dans les workflows

5.1 Matériel et Méthodes

Les entités nommées ont été extraites à l'aide de la librairie Python NLStruct (Wajsbürt (2021), v0.0.5), qui implémente un modèle neuronal de reconnaissance d'entités nommées biLSTM-CRF à l'aide de trois composants : un encodeur au niveau du texte, des mots et un module de détection des frontières d'entités. Cette librairie présente également le double avantage de prendre en charge la détection d'entités imbriquées et d'accepter en entrée des fichiers au format Brat. Nous avons réalisé nos expériences avec quatre modèles de langue issus de la

librairie Huggingface (Wolf et al. (2019)) sensibles ou non à la casse et entraînés sur des corpus de langue générique en anglais (BertUncased, BerCased) ou de la littérature scientifique en anglais (SciBertUncased et SciBertCased).

Nous avons aussi testé la librairie Python OpenNre (Han et al. (2019)) pour l'extraction des relations. Le tableau 2 montre qu'il existe une grande disproportion entre les occurences des différentes relations qui a conduit dans ces expériences préliminaires à de mauvaises perfomances en prédisant la quasi-totalité des relations en *DescriptionOf*.

5.2 Résultats

Pour entrainer nos différents modèles d'extraction d'entités nommées, nous avons choisi dix-neuf articles. Nous les avons répartis en deux jeux de données : 70% dans le jeu d'entraînement (soit treize articles) et 30% des articles dans le jeu validation (six articles). Nous avons relancé les différents modèles cités ci-dessus cinq fois. A chaque fois, nous avons pioché aléatoirement treize articles pour former le jeu d'entraînement et les six articles restants formaient le jeu de validation. Pour calculer les différents scores des modèles ci-dessus, nous avons utilisé les cinq articles restants (différent des dix-neuf articles utilisés pour les jeux d'entraînement et de validation). Le tableau 3 indique la moyenne des scores obtenus sur ces cinq itérations. Les scores ont été obtenus à l'aide de l'outil BRAT-Eval en comparant les nouveaux cinq articles annotés manuellement avec les annotations de l'outil NLStruct.

	Precision	Recall	F1
BertUncased	*0.66406*	*0.60264*	*0.63184*
	0.7434	**0.67394**	**0.70698**
BertCased	*0.66182*	*0.59822*	*0.62822*
	0.73728	**0.66704**	**0.7002**
SciBertUncased	*0.6685*	*0.62366*	*0.64518*
	0.75106	**0.6995**	**0.72422**
SciBertCased	*0.64816*	*0.5915*	*0.61834*
	0.7356	**0.67044**	**0.70132**

TAB. 3 – *Moyenne des scores obtenus.*

Le meilleur modèle est SciBertUncased, qui offre une F-mesure globale de 0,72 pour l'extraction d'entités (tableau 3). Cette performance est tout à fait encourageante, au regard de la taille restreinte du corpus d'entrainement utilisé (13 articles - environs 14 000 mots). Les expériences confirment également l'intuition qu'un modèle entrainé sur un corpus d'articles scientifique issus des domaines biomédical et informatique (SciBERT) est plus adapté pour notre tâche qu'un modèle entrainé sur un corpus web (BERT).

6 Discussion

Nous introduisons dans cet article la première version d'une méthode d'extraction d'informations relatives aux workflows bioinformatiques décrits dans la littérature en anglais. La

solution est complète au sens où elle comprend un ensemble d'étapes depuis la sélection du corpus d'intérêt, la modélisation des entités et associations d'intérêt, la constitution d'un guide d'annotation, la production d'un corpus annoté, l'évaluation de l'accord inter annotateurs et va jusqu'à la proposition d'une solution automatique d'extraction d'information dont les premiers résultats sont très encourageants.

L'extraction de workflows depuis les publications est un problème recurrent et un besoin identifié depuis de nombreuses années en bioinformatique. Certaines approches ont été proposées (Allard et al. (2019)) sur les *business process* qui peuvent être vus comme des workflows mais dont l'identification dans les textes demeure très éloignée de celle des workflows bioinformatiques. L'extraction d'informations sur les workflows depuis les publications comporte celle des outils (logiciels) utilisés pour implanter les étapes du workflow. Wei et al. (2020) réalise cette étape et extrait de noms de logiciels dans les résumés et titres de 1 120 articles indéxés dans PubMed. Bien que notre corpus de travail comporte moins d'articles, notre étude reste d'intérêt au sens où nous travaillons sur le texte complet d'articles et que nous proposons de considérer d'avantage de composants de workflows.

Ce travail préliminaire ouvre plusieurs axes. D'abord, le nombre d'articles annotés doit être augmenté pour avoir une base d'apprentissage plus conséquente. La phase de pré-annotation doit permettre un passage à l'echelle pour tendre vers plusieurs centaines d'articles annotés.

L'extraction des relations constitue un autre enjeu notamment pour pouvoir faire face à la grande disproportion entre les occurences des différentes relations. A plus long terme un enjeu important consiste en l'extraction de l'ordre attendu entre les différentes étapes du workflows.

Références

Allard, T., P. Alvino, L. Shing, A. Wollaber, et J. Yuen (2019). A dataset to facilitate automated workflow analysis. *PloS one 14*(2), e0211486.

Artstein, R. et M. Poesio (2008). Inter-coder agreement for computational linguistics. *Computational linguistics 34*(4), 555–596.

Buchmann, J. et E. Holmes (2019). Entrezpy : A python library to dynamically interact with the ncbi entrez databases. *Bioinformatics (Oxford, England) 35*, 4511 – 4514.

Cohen-Boulakia, S., K. Belhajjame, O. Collin, J. Chopard, C. Froidevaux, A. Gaignard, K. Hinsen, P. Larmande, Y. Le Bras, F. Lemoine, F. Mareuil, H. Ménager, C. Pradal, et C. Blanchet (2017). Scientific workflows for computational reproducibility in the life sciences : Status, challenges and opportunities. *Fut Gen Comput Systems 75*, 284–298.

Di Tommaso, P., M. Chatzou, E. W. Floden, P. Barja, E. Palumbo, et C. Notredame (2017). Nextflow enables reproducible computational workflows. *Nature Biotech 35*, 316–319.

Fort, K. (2016). *Collaborative Annotation for Reliable Natural Language Processing : Technical and Sociological Aspects*. Wiley-ISTE.

Fort, K., A. Nazarenko, et S. Rosset (2012). Modeling the complexity of manual annotation tasks : a grid of analysis. In *Proceedings of COLING 2012*, Mumbai, India, pp. 895–910. The COLING 2012 Organizing Committee.

Grouin, C., S. Rosset, P. Zweigenbaum, K. Fort, O. Galibert, et L. Quintard (2011). Proposal for an extension of traditional named entities : From guidelines to evaluation, an overview.

In *Proc of Linguistic Annotation Workshop (LAW-V)*, Portland, OR, pp. 92–100.

Han, X., T. Gao, Y. Yao, D. Ye, Z. Liu, et M. Sun (2019). OpenNRE : An open and extensible toolkit for neural relation extraction. In *Proc. EMNLP-IJCNLP*, pp. 169–174.

Köster, J. et S. Rahmann (2012). Snakemake - a scalable bioinformatics workflow engine. *Bioinformatics (Oxford, England) 28*, 2520–2.

Neves, M. et J. Ševa (2019). An extensive review of tools for manual annotation of documents. *Briefings in bioinformatics 22*, 146–163.

NLM (2021). Medline, pubmed, and pmc (pubmed central) : How are they different ? Last Reviewed : October 13, 2021.

Névéol, A., R. Islamaj Doğan, et Z. Lu (2011). Semi-automatic semantic annotation of pubmed queries : A study on quality, efficiency, satisfaction. *Journal of Biomedical Informatics 44*(2), 310–318.

Stenetorp, P., S. Pyysalo, G. Topic, T. Ohta, S. Ananiadou, et J. Tsujii (2012). brat : a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 102–107.

Verspoor, K., A. Jimeno-Yepes, L. Cavedon, T. McIntosh, A. Herten-Crabb, Z. Thomas, et J.-P. Plazzer (2013). Annotating the biomedical literature for the human variome. *Database : the journal of biological databases and curation 2013*, bat019.

Wajsbürt, P. (2021). *Extraction and normalization of simple and structured entities in medical documents*. Theses, Sorbonne Université.

Wei, Q., Y. Zhang, M. Amith, R. Lin, J. Lapeyrolerie, C. Tao, et H. Xu (2020). Recognizing software names in biomedical literature using machine learning. *Health informatics journal 26*(1), 21–33.

Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et J. Brew (2019). Huggingface's transformers : State-of-the-art natural language processing. *CoRR abs/1910.03771*, 38–45.

Summary

Scientific workflows provide bioinformaticians a mean to represent, exchange and ensure the reproducibility of their analysis pipelines. Workflows are described in literature (text) and/or stored in workflow repositories (code). A major challenge to ensure better workflow reuse is to rebuild the link between the documentation (text) and the workflow code.

Based on workflow descriptions found in the full text of articles in English, we propose a method for representing and extracting information about the components of workflows. We present a corpus of 24 articles annotated with a schema made of 16 entities and 10 relations. We use this corpus to train and evaluate statistical models for extracting information about workflows. The results obtained show the feasibility of the task and are a first step towards the integration of workflow information from the literature and workflow repositories.