Combiner plongements de graphes et clustering pour l'alignement de connaissances pharmacogénomiques

Pierre Monnin, Chedy Raïssi, Amedeo Napoli, Adrien Coulet

Combiner plongements de graphes et clustering pour l'alignement de connaissances pharmacogénomiques

Pierre Monnin[1,2], Chedy Raïssi[1,3], Amedeo Napoli[1], Adrien Coulet[1,4]

[1] Université de Lorraine, CNRS, Inria, LORIA, Nancy, France
{prenom.nom}@loria.fr

[2] Orange, Belfort, France
pierre.monnin@orange.com

[3] Ubisoft, Singapore
chedy.raissi@ubisoft.com

[4] Inria Paris & Centre de Recherche des Cordeliers, Inserm, Université Paris Cité, Paris, France
adrien.coulet@inria.fr

Résumé : La publication et l'édition concurrentes de graphes de connaissances, notamment biomédicaux, au sein du Web des données entraînent l'existence de recouvrements entre ces graphes. Leur appariement est donc une tâche essentielle si l'on veut considérer l'ensemble des connaissances disponibles pour un domaine, ce qui inclut l'identification de nœuds équivalents, plus spécifiques ou similaires au sein de graphes agrégés. Nous proposons d'identifier et de typer ces appariements en apprenant des plongements de nœuds avec les réseaux convolutifs de graphes, puis en partitionnant par clustering les nœuds en fonction de la similarité de leurs plongements. Nous avons expérimenté cette approche en alignant des connaissances pharmacogénomiques, l'application réelle qui a motivé ce travail. Nous examinons particulièrement l'apport des connaissances de domaine en mesurant l'amélioration des résultats d'alignement après application de règles d'inférence. Nous observons des distances entre plongements cohérentes avec les alignements connus, notamment des distances plus faibles pour les équivalences.

Mots-clés : Graphe de connaissances, appariement, réseaux convolutifs de graphes, ontologie, clustering, pharmacogénomique.

1 Introduction

La communauté du Web sémantique a établi des outils et des standards qui facilitent la construction de graphes de connaissances (Hogan *et al.*, 2021). L'agrégation de ce type de graphes permet de grouper des éléments de connaissances de provenances diverses, dans l'espoir de disposer d'un point d'accès unique à l'ensemble des connaissances disponibles pour un domaine. Pour permettre cette agrégation, une tâche cruciale consiste à aligner différents éléments constituant les graphes de connaissances, c'est-à-dire à identifier des correspondances entre nœuds, arcs ou sous-graphes (Euzenat & Shvaiko, 2013). Le résultat attendu consiste à savoir si deux éléments comparés sont équivalents, si l'un est plus spécifique que l'autre, ou s'ils sont liés par une relation de similarité plus lâche.

Nous décrivons dans cet article une méthode pour identifier différents types d'alignements entre les nœuds d'un graphe qui agrège des éléments de connaissances de provenances diverses. Le développement de cette méthode est directement motivé par notre effort de construction d'un graphe qui agrège les connaissances de l'état de l'art du domaine de la pharmacogénomique (souvent noté PGx). Ce domaine étudie l'influence des facteurs génomiques sur la variabilité interindividuelle de réponse aux médicaments, ce qui est particulièrement important pour la mise en œuvre d'une médecine de précision (Caudle *et al.*, 2014). Une connaissance typique de ce domaine est par exemple illustrée par la Figure 1 qui représente une relation pharmacogénomique appelée `pgr_1`. Celle-ci établit que les patients traités par warfarine, un anti-coagulant, risquent de rencontrer des problèmes vasculaires à cause de variations génétiques du gène CYP2C9. Dans des travaux précédents, nous avons extrait 50 435 relations de ce type à partir de bases de données biomédicales de référence,

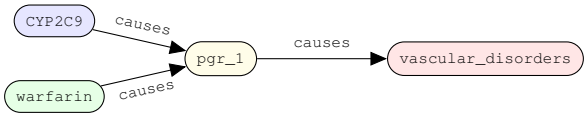


FIGURE 1 – *Représentation d'une relation pharmacogénomique entre le gène* `CYP2C9`*, l'anticoagulant* `warfarin` *et le phénotype* `vascular_disorders`*. Cette relation est réifiée au travers de l'individu* `pgr_1`*, qui lie ses composants par le prédicat* `causes`*.*

comme PharmGKB (Whirl-Carrillo *et al.*, 2012), et à partir de la littérature scientifique (Legrand *et al.*, 2020). Ces unités de connaissances de provenances diverses ont été agrégées dans un graphe appelé PGxLOD, disponible à `https://pgxlod.loria.fr/` (Monnin *et al.*, 2019b). La diversité de provenance des éléments agrégés et leurs redondances partielles ou complètes motivent notre besoin de les comparer et les aligner afin de consolider notre vue du domaine.

Pour cela, nous proposons d'aligner les nœuds d'un graphe de connaissances qui représentent des relations n-aires, comme celle de la Figure 1, par une approche qui combine plongements de graphes et clustering, suivant les étapes principales représentées en Figure 2 et détaillées en Section 2. Les plongements de graphes sont des vecteurs de faible dimension qui représentent les sous-structures d'un graphe (nœuds, arcs, ou sous-graphes) et qui visent à préserver autant que possible les propriétés du graphe (Cai *et al.*, 2018). Dans notre cas, nous proposons d'apprendre des plongements de nœuds, et ce avec des réseaux convolutifs de graphes (ou GCN pour l'anglais *Graph Convolutional Networks*) (Kipf & Welling, 2017; Schlichtkrull *et al.*, 2018) de telle sorte que les distances entre les plongements de nœuds similaires soient faibles. Nous avons choisi des plongements de graphes car leur nature continue fournit une flexibilité qui semble permettre de faire face aux représentations hétérogènes des nœuds à apparier (Guha, 2015). Pour suggérer des relations d'alignement à partir des plongements de nœuds, nous appliquons un algorithme de clustering sur l'espace des plongements et considérons les nœuds qui appartiennent à un même cluster comme suffisamment similaires pour être liés. Les clusters résultants sont évalués par comparaison avec des clusters "références", c'est-à-dire des clusters que nous cherchons à reproduire. Ces clusters références sont des ensembles de nœuds liés directement ou indirectement par des alignements préalablement connus. Notre approche est donc supervisée et nécessite la préexistence de tels alignements. Il est à noter que les graphes de connaissances du Web sémantique sont représentés avec des formalismes comme les Logiques de Descriptions (Baader *et al.*, 2003) auxquels sont associées des règles d'inférence. Dans ce travail, nous nous attachons également à évaluer l'impact de l'application de ces règles sur les alignements produits par notre approche. Nous détaillons les éléments méthodologiques en Section 2 et décrivons les expériences menées sur PGxLOD et les résultats obtenus en Section 3. Nous concluons l'article par quelques pistes de discussion en Section 4.

Cet article est une version synthétique et en français de l'article Monnin *et al.* (2022).

2 Alignement de nœuds par plongement et clustering

2.1 Description de l'approche d'alignement proposée

Notre approche prend en entrée un graphe de connaissances $\mathcal{K}$ agrégeant des éléments de différentes sources et un ensemble S de nœuds à aligner, S étant un sous-ensemble des nœuds de $\mathcal{K}$. La sélection initiale des nœuds est motivée par notre application, puisque nous cherchons seulement à aligner les nœuds qui représentent des relations pharmacogénomiques. Nous supprimons de $\mathcal{K}$ et S les littéraux et les arcs incidents aux littéraux. Ainsi, un nœud est soit une entité, soit une classe. Nous disposons de clusters "références", c'est-à-dire de sous-ensembles de nœuds de S qui sont déjà étiquetés comme similaires. Nous détaillons en Section 2.2 comment ces clusters références sont construits à partir d'alignements préexistants. La Section 2.3 détaille quant-à-elle notre façon de considérer les règles d'inférences.

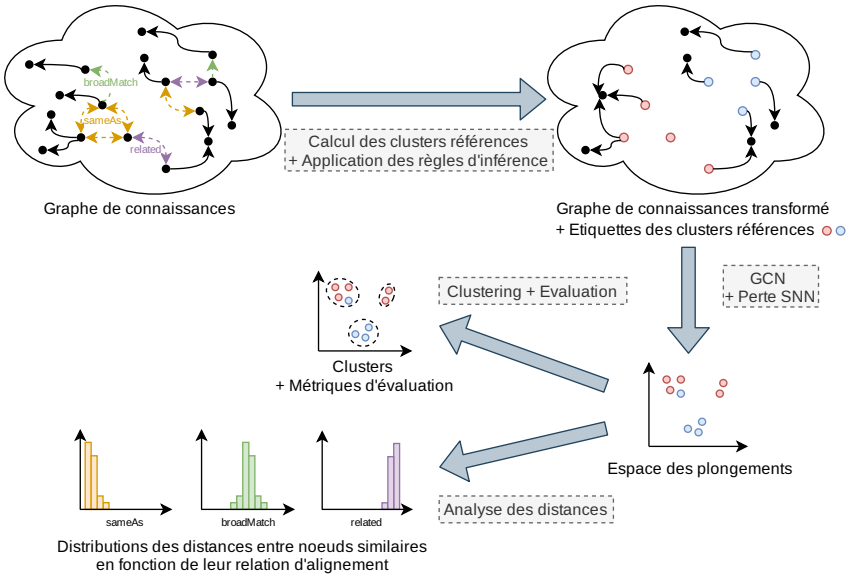


FIGURE 2 – *Principales étapes de notre approche. Les clusters références sont formés à partir d'alignements préexistants (*`owl:sameAs`*,* `skos:broadMatch`*,* `skos:related`*, etc.). Ces alignements sont masqués et des règles d'inférences associées aux connaissances de domaine sont appliquées. Les plongements de nœuds sont appris avec des réseaux convolutifs de graphes (GCN) et la fonction de perte Soft Nearest Neighbor (SNN). Un clustering est ensuite appliqué sur l'espace des plongements. Les clusters résultants sont évalués au regard des clusters références. Une analyse des distances est également menée.*

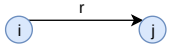
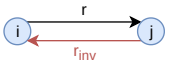
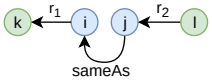
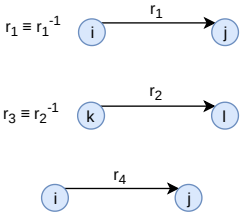
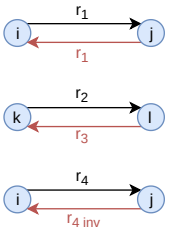
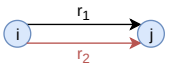
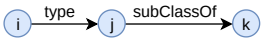
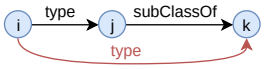
Nous proposons d'aligner les nœuds de S en suivant deux étapes : *(1)* Apprendre des plongements pour tous les nœuds de $\mathcal{K}$ de sorte à ce que la distance entre les plongements des nœuds de S au sein de mêmes clusters références soit faible (Section 2.4) ; *(2)* Appliquer un clustering sur les plongements des nœuds de S et considérer que les nœuds qui sont assignés au même cluster sont similaires (Section 2.5).

2.2 Obtention des clusters références

Dans notre cas, nous avons obtenu nos clusters références à partir d'alignements préexistants. Ces alignements sont issus d'une méthode à base de règles précédemment décrite dans Monnin *et al.* (2020) et emploient cinq types de relations d'alignement différentes. Par exemple, les nœuds peuvent être identiques (liens `owl:sameAs`), plus lâchement associés (liens `skos:related`), ou l'un d'entre eux peut être plus spécifique que l'autre (liens `skos:broadMatch`). Dans Monnin *et al.* (2022), nous avons expérimenté avec plusieurs ensembles de clusters références, obtenus en considérant diverses combinaisons de types d'alignements. Par souci de simplicité, nous présentons ici uniquement l'ensemble des clusters références obtenus en regroupant les nœuds liés directement ou indirectement par des relations d'alignement quelles qu'elles soient (dénoté $\mathcal{C}_0$ dans Monnin *et al.* (2022)).

Nous avons construit les clusters références en rassemblant les nœuds de S liés directement ou indirectement par ces alignements. Ces clusters correspondent donc aux composantes connexes calculées en ne sélectionnant dans $\mathcal{K}$ que les relations d'alignement entre nœuds de S. Ainsi, les relations d'alignement sont considérées comme symétriques (arcs non-orientés) et transitives (composantes connexes), ce qui est cohérent avec la sémantique d'une majorité des relations d'alignement. L'analyse des distances décrite en Section 3.2 et indiquée en Figure 2, examine le lien entre les types d'alignement (connus) et la distance entre les plongements des nœuds alignés (calculée).

TABLE 1 – *Présentation visuelle des transformations appliquées à $\mathcal{K}$ pour évaluer l'influence de diverses règles d'inférence sur notre tâche d'alignement.*

Graphe	Avant inférence	Après inférence
$\mathcal{G}_0$		
$\mathcal{G}_1$		
$\mathcal{G}_2$		
$\mathcal{G}_3$		
$\mathcal{G}_4$		
$\mathcal{G}_5$	Toutes les transformations de $\mathcal{G}_1$ à $\mathcal{G}_4$	

2.3 Règles d'inférence et graphes résultants

Nous souhaitons évaluer l'impact de l'application de différentes règles d'inférence (seules ou combinées) associées aux connaissances de domaine sur notre tâche d'alignement. La Table 1 liste les règles d'inférence considérées et nomme les graphes résultant de l'application de ces règles sur $\mathcal{K}$. $\mathcal{G}_0$ est le graphe de référence où aucune règle d'inférence n'est appliquée, mais des prédicats inverses abstraits sont systématiquement ajoutés. Cet ajout d'inverses abstraits est réalisé dans toutes les autres transformations, sauf $\mathcal{G}_2$ et $\mathcal{G}_5$. $\mathcal{G}_1$ résulte de la contraction des prédicats `owl:sameAs`. Notons que cette transformation n'est pas standard puisque, classiquement, les règles d'inférence propagent les propriétés aux nœuds équivalents plutôt que de les agréger. Cependant, dans notre cas d'étude, nous avons préféré cette transformation car elle est plus adaptée à l'architecture des GCN (au passage de messages notamment). Dans $\mathcal{G}_2$, nous n'ajoutons pas systématiquement un inverse abstrait mais considérons les définitions d'inverses et de symétrie des prédicats quand elles existent. $\mathcal{G}_3$ prend en considération la hiérarchie des prédicats. $\mathcal{G}_4$ complète le co-domaine des prédicats `rdf:type` en considérant la hiérarchie des classes de l'ontologie définie par le prédicat `rdfs:subClassOf`. Enfin, $\mathcal{G}_5$ est le graphe résultant de la combinaison des transformations de $\mathcal{G}_1$ à $\mathcal{G}_4$.

2.4 Apprentissage des plongements de nœuds

Pour l'apprentissage des plongements des nœuds de $\mathcal{K}$, nous utilisons les GCN et la fonction de perte appelée *Soft Nearest Neighbor* (Frosst *et al.*, 2019). Dans la suite, nous adoptons le modèle, les notations et les définitions de Schlichtkrull *et al.* (2018). Ainsi, $\mathcal{R}$ est l'ensemble des prédicats de $\mathcal{K}$. Etant donné un nœud i et un prédicat $r \in \mathcal{R}$, $\mathcal{N}_i^r$ est l'ensemble des nœuds atteignables depuis i par un arc étiqueté par r.

Les GCN peuvent être vus comme un système de passage de messages à plusieurs couches,

dans lequel le plongement $h_i^{(l+1)}$ du nœud i à la couche $(l+1)$ dépend du plongement de ses voisins à la couche (l), comme défini par l'Equation (1).

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right) \quad (1)$$

Le calcul de la convolution sur les nœuds voisins j de i est fait avec une matrice de poids $W_r^{(l)}$ spécifique à chaque prédicat $r \in \mathcal{R}$ et à chaque couche (l). La convolution est régularisée à l'aide de la constante $c_{i,r}$ propre à chaque paire (nœud, prédicat). Suivant Schlichtkrull *et al.* (2018), nous utilisons $c_{i,r} = |\mathcal{N}_i^r|$. La matrice $W_0^{(l)}$ permet un passage de messages entre couches pour un même nœud : le plongement de i à la couche $(l+1)$ dépend ainsi également de son plongement à la couche (l). σ est une fonction non linéaire telle que ReLU ou $\tanh$. Dans notre travail, nous utilisons un modèle GCN à 3 couches ayant comme fonctions d'activation $\tanh$ (couches 1 et 2) et une fonction linéaire (couche 3). Il convient de remarquer que le nombre de prédicats dans $\mathcal{K}$ peut entraîner un grand nombre de paramètres $W_r^{(l)}$ à optimiser. Pour garantir le passage à l'échelle de notre approche et réduire le nombre de paramètres à optimiser, nous utilisons la décomposition en bases décrite par Schlichtkrull *et al.* (2018). D'après l'Equation (1), les GCN calculent le plongement d'un nœud en considérant les plongements de ses voisins dans le graphe. Ainsi, les nœuds ayant des voisinages similaires auront des plongements similaires. Ce modèle est donc bien adapté à une approche d'alignement dirigée par la structure (Pang *et al.*, 2019; Wang *et al.*, 2018). Une telle approche correspond à notre objectif d'alignement de relations pharmacogénomiques car celles-ci sont entièrement définies par leur voisinage dans $\mathcal{K}$ (Figure 1).

Dans l'état de l'art, nous avons pu trouver des applications des GCN à la classification de nœuds ou à la prédiction de liens (Kipf & Welling, 2017; Schlichtkrull *et al.*, 2018; Wang *et al.*, 2018). Notre objectif étant différent, *i.e.,* regrouper des nœuds similaires, nous proposons d'utiliser une fonction de perte plus adaptée à cette application. Ainsi, nous entraînons intégralement les GCN depuis leur initialisation en minimisant la fonction de perte *Soft Nearest Neighbor* (ou SNN) qui, à notre connaissance, n'a jamais été utilisée avec des GCN. Cette fonction de perte a été décrite par Frosst *et al.* (2019) et est définie comme suit,

$$\mathcal{L}_{\text{SNN}} = -\frac{1}{|N|}\sum_{i\in N}\log\left(\frac{\sum\limits_{\substack{j\in N\\ j\neq i\\ Y_i=Y_j}} e^{-\frac{||h_i-h_j||^2}{T}}}{\sum\limits_{\substack{k\in N\\ k\neq i}} e^{-\frac{||h_i-h_k||^2}{T}}}\right) \quad (2)$$

avec : N un ensemble d'entraînement composé de nœuds assignés aux clusters références ($N \subseteq S$) ; Y les étiquettes des nœuds de N (ces étiquettes correspondent à l'assignation des nœuds de N aux clusters références) ; T la température ; h les plongements des nœuds (en sortie de la dernière couche du GCN). Minimiser la fonction de perte SNN revient à minimiser les distances intra-clusters (nœuds ayant la même étiquette) et à maximiser les distances inter-clusters (nœuds ayant des étiquettes différentes) pour les clusters références. La température T détermine dans quelle mesure les distances influencent la perte $\mathcal{L}_{\text{SNN}}$. En effet, les grandes distances entre plongements sont considérées quand T est grand, alors que seules les petites distances entre plongements sont considérées quand T est petit. Afin d'éviter de considérer T comme un hyperparamètre du modèle, nous suivons Frosst *et al.* (2019) et initialisons T à une valeur prédéfinie et apprenons $\frac{1}{T}$ comme un paramètre du modèle.

Cette étape permet ainsi d'apprendre le plongement de chaque nœud de $\mathcal{K}$ de telle sorte que la distance entre les plongements des nœuds similaires dans N soit faible. Nous rappelons qu'à ce stade, le type précis (et donc la sémantique) de l'alignement entre les nœuds n'est pas considéré car seules les étiquettes de clusters références sont connues du modèle.

TABLE 2 – *Résumé des résultats obtenus avec les clusters références considérant toutes les relations d'alignement et tous les graphes.*

Graphe	Performance
$\mathcal{G}_0$	Référence
$\mathcal{G}_1$	Amélioration
$\mathcal{G}_2$	Légère détérioration
$\mathcal{G}_3$	Amélioration
$\mathcal{G}_4$	Détérioration constante
$\mathcal{G}_5$	**Amélioration – Meilleurs résultats**

2.5 Clustering des plongements

Une fois les plongements de la dernière couche du GCN calculés, nous appliquons un clustering sur les plongements h_i pour tous les nœuds $i \in S$ non-utilisés lors de l'apprentissage. Les nœuds assignés à un même cluster sont considérés comme similaires et liés par une relation d'alignement. Les clusters générés sont évalués vis à vis des clusters références. Nous avons considéré et comparé trois méthodes de clustering différentes dans (Monnin *et al.*, 2022) mais, par souci de simplicité, nous ne rapportons ici que les résultats obtenus avec l'algorithme Single. Celui-ci consiste en un clustering hiérarchique fusionnant successivement les pairs de clusters dont la distance entre les observations les plus proches est minimale. Il prend comme paramètre le nombre de clusters à trouver.

3 Expérimentations

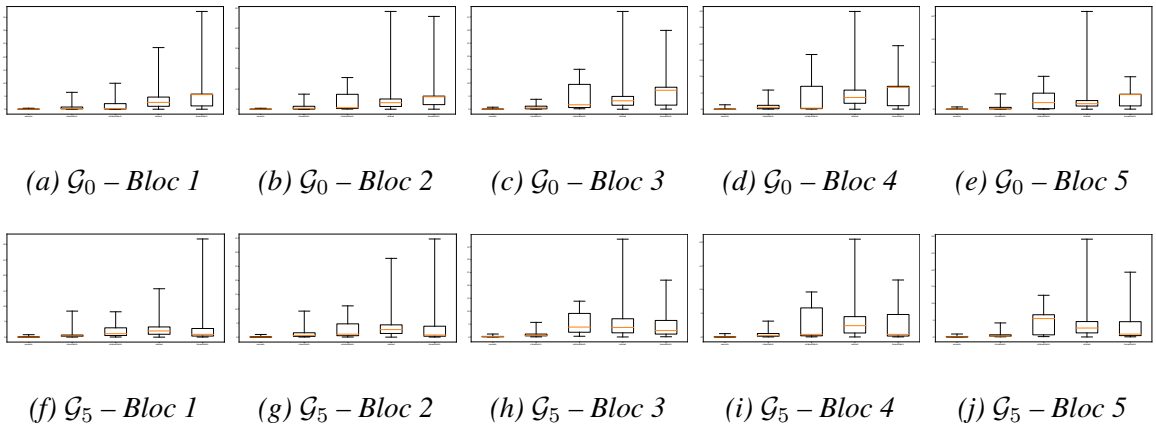
Nous avons réalisé nos expériences sur le graphe de connaissances PGxLOD, qui a motivé ce travail, avec une validation croisée à 5 blocs. Nos implémentations sont en Python et utilisent PyTorch et Deep Graph Library pour l'apprentissage des plongements, et scikit-learn pour le clustering. Notre code est disponible sur GitHub à `https://github.com/pmonnin/gcn-matching`.

3.1 Évaluation du clustering

Le clustering est seulement appliqué sur les plongements de nœuds appartenant aux ensembles de test car ce sont ces nœuds que nous voulons aligner. Nous évaluons donc les performances sur ces nœuds. Les résultats obtenus avec les clusters références considérant toutes les relations d'alignement et tous les graphes sont résumés en Table 2. On peut remarquer qu'appliquer toutes les règles d'inférence ($\mathcal{G}_5$) permet d'obtenir les meilleurs résultats. Cependant, la complétion du co-domaine des prédicats `rdf:type` en considérant la hiérarchie des classes de l'ontologie ($\mathcal{G}_4$) semble dégrader les performances.

3.2 Analyse des distances

Nous rappelons que, lors de l'apprentissage et du clustering, notre modèle ignore les différents types des relations d'alignement qui lient les nœuds similaires. En effet, la fonction de perte SNN ne considère que les étiquettes des clusters références, étiquettes qui n'indiquent pas les types d'alignements utilisés pour former les clusters. Cependant, inspirés par nos travaux préliminaires (Monnin *et al.*, 2019a), nous représentons en Figure 3 les distributions des distances entre nœuds similaires des ensembles de test en fonction du type de leur relation d'alignement. Ainsi, nous constatons que ces distributions de distances sont cohérentes avec la nature des relations d'alignement. Par exemple, les nœuds lâchement similaires ont



(a) $\mathcal{G}_0$ *– Bloc 1* *(b)* $\mathcal{G}_0$ *– Bloc 2* *(c)* $\mathcal{G}_0$ *– Bloc 3* *(d)* $\mathcal{G}_0$ *– Bloc 4* *(e)* $\mathcal{G}_0$ *– Bloc 5*

(f) $\mathcal{G}_5$ *– Bloc 1* *(g)* $\mathcal{G}_5$ *– Bloc 2* *(h)* $\mathcal{G}_5$ *– Bloc 3* *(i)* $\mathcal{G}_5$ *– Bloc 4* *(j)* $\mathcal{G}_5$ *– Bloc 5*

FIGURE 3 – *Distributions des distances entre nœuds similaires en fonction de leur relation d'alignement pour chaque ensemble de test sur les graphes* $\mathcal{G}_0$ *et* $\mathcal{G}_5$*. Sur chaque sous-figure, les relations sont, de gauche à droite,* `owl:sameAs`*,* `skos:closeMatch`*,* `skos:relatedMatch`*,* `skos:related`*, et* `skos:broadMatch`*.*

tendance à être plus éloignés que les nœuds équivalents (la médiane des distances est plus élevée).

4 Discussion et Conclusion

Pour nos clusters références construits à partir de l'ensemble des relations d'alignement, l'application de la plupart des règles d'inférence améliore les résultats, à l'exception de $\mathcal{G}_4$. Dans ce graphe, la relation d'instanciation est saturée. Ainsi, les classes "générales" (les plus hautes dans la hiérarchie) sont directement liées aux entités qui les instancient, au lieu d'être indirectement liées. Par conséquent, les plongements de ces entités sont directement influencés par les classes générales et spécifiques qui sont considérées via le même prédicat `rdf:type`, rendant possiblement difficile pour les GCN de pondérer différemment leur influence. Les classes spécifiques étant plus importantes que les classes générales pour discriminer les nœuds similaires des nœuds dissimilaires, une telle influence indiscriminée dans $\mathcal{G}_4$ pourrait expliquer la détérioration des performances d'alignement. On remarque toutefois que $\mathcal{G}_5$ (qui inclut les transformations de $\mathcal{G}_4$) permet d'obtenir les meilleurs résultats, ce qui souligne l'intérêt de considérer toutes les règles d'inférence ensemble. Cependant, de par la performance dégradée de $\mathcal{G}_4$ par rapport à $\mathcal{G}_0$, il pourrait être intéressant de tester en ne considérant que les transformations incluses dans $\mathcal{G}_1$, $\mathcal{G}_2$, et $\mathcal{G}_3$.

Par rapport à l'analyse des distances, la Figure 3 montre que les distances entre nœuds similaires sont différentes en fonction de la relation d'alignement qui les lie. Nous rappelons que notre modèle ignore l'existence de ces différentes relations d'alignement lors du calcul de la fonction de perte SNN. Toutefois, les distances reflètent la nature des relations d'alignement : les similarités fortes (liens `owl:sameAs` et `skos:closeMatch`) présentent des distances plus faibles que les similarités plus lâches (liens `skos:relatedMatch` and `skos:related`). Seule la relation `skos:broadMatch` semble être plus difficile à positionner par rapport aux autres relations. Cette observation pourrait s'expliquer par la nature de cette relation qui est l'unique relation d'alignement non-symétrique. De telles distributions cohérentes des distances semblent indiquer la "redécouverte" des relations d'alignement par les GCN et nous amène à penser que la distance entre plongements porte une certaine sémantique, c'est-à-dire que des distances faibles indiquent des types de relations associées à une similarité forte. Par conséquent, une perspective intéressante pourrait être de prédire le type de la relation d'alignement exacte entre les nœuds d'un même cluster, en fonction de la distance entre leurs plongements. De plus, de telles différences entre distributions de distances

semblent confirmer que le calcul par les GCN des plongements d'un nœud en fonction de son voisinage est bien adapté au calcul d'alignements basé sur la structure.

Nos résultats soulignent l'intérêt de considérer les connaissances de domaine représentées au sein des graphes de connaissances dans les approches de plongement, y compris en envisageant une intégration plus avancée de ces connaissances au sein des modèles de plongement. Des travaux futurs pourraient étudier la même tâche avec des règles d'inférence additionnelles (par exemple issues de la sémantique de OWL 2 RL), avec d'autres modèles de plongement, ou pour aligner des concepts entre ontologies. De plus, nous n'avons pas utilisé de mécanismes d'attention, qui pourrait également permettre l'intégration de connaissances de domaine comme pour les *Logic Attention Networks* (Wang *et al.*, 2019). Dans notre travail, les règles d'inférence associées aux connaissances de domaine sont utilisées pour transformer le graphe de connaissances avec une opération de pré-traitement. Néanmoins, ces règles pourraient directement être intégrées dans le modèle (par exemple en partageant les poids entre prédicats et super-prédicats). Les littéraux pourraient également être considérés (Wang *et al.*, 2018). Dans une perspective plus large, un futur travail important réside dans l'étude de la possibilité pour des relations associées à d'autres types de sémantiques (comme `part-of` par exemple) d'émerger spontanément dans l'espace des plongements.

Références

F. BAADER & OTHERS, Eds. (2003). *The Description Logic Handbook : Theory, Implementation, and Applications*. Cambridge University Press.

CAI H. *et al.* (2018). A comprehensive survey of graph embedding : Problems, techniques, and applications. *IEEE Trans. Knowl. Data Eng.*, **30**(9), 1616–1637.

CAUDLE K. E. *et al.* (2014). Incorporation of pharmacogenomics into routine clinical practice : the clinical pharmacogenetics implementation consortium (CPIC) guideline development process. *Current Drug Metabolism*, **15**(2), 209–217.

EUZENAT J. & SHVAIKO P. (2013). *Ontology Matching, Second Edition*. Springer.

FROSST N. *et al.* (2019). Analyzing and improving representations with the soft nearest neighbor loss. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*.

GUHA R. V. (2015). Towards a model theory for distributed representations. In *2015 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 22-25, 2015*.

HOGAN A. *et al.* (2021). *Knowledge Graphs*. Synthesis Lectures on Data, Semantics, and Knowledge. Morgan & Claypool Publishers.

KIPF T. N. & WELLING M. (2017). Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017*.

LEGRAND J. *et al.* (2020). PGxCorpus, a manually annotated corpus for pharmacogenomics. *Scientific Data*, **7**(1), 3.

MONNIN P. *et al.* (2019a). Knowledge reconciliation with graph convolutional networks : Preliminary results. In *Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG2019) Co-located with the 16th Extended Semantic Web Conference 2019 (ESWC 2019)*.

MONNIN P. *et al.* (2019b). PGxO and PGxLOD : a reconciliation of pharmacogenomic knowledge of various provenances, enabling further comparison. *BMC Bioinformatics*, **20-S**(4), 139 :1–139 :16.

MONNIN P. *et al.* (2020). Knowledge-based matching of n-ary tuples. In *Ontologies and Concepts in Mind and Machine - 25th International Conference on Conceptual Structures, ICCS 2020*.

MONNIN P. *et al.* (2022). Discovering alignment relations with Graph Convolutional Networks : A biomedical case study. *Semantic Web*, **13**(3), 379–398.

PANG N. *et al.* (2019). Iterative entity alignment with improved neural attribute embedding. In *Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG2019) Co-located with the 16th Extended Semantic Web Conference 2019 (ESWC 2019)*.

SCHLICHTKRULL M. S. *et al.* (2018). Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference, ESWC 2018*.

WANG P. *et al.* (2019). Logic attention based neighborhood aggregation for inductive knowledge graph embedding. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*.

WANG Z. *et al.* (2018). Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

WHIRL-CARRILLO M. *et al.* (2012). Pharmacogenomics knowledge for personalized medicine. *Clinical pharmacology and therapeutics*, **92**(4), 414.