frontiers in Cell and Developmental Biology

ORIGINAL RESEARCH
published: 11 February 2021
doi: 10.3389/fcell.2021.562908

FOntCell: Fusion of Ontologies of Cells

Javier Cabau-Laporta[1], *Alex M. Ascensión*[1], *Mikel Arrospide-Elgarresta*[1], *Daniela Gerovska*[1,2]* *and Marcos J. Araúzo-Bravo*[1,2,3,4,5,6]*

[1] *Computational Biology and Systems Biomedicine Group, Biodonostia Health Research Institute, San Sebastián, Spain,* [2] *Computational Biomedicine Data Analysis Platform, Biodonostia Health Research Institute, San Sebastián, Spain,* [3] *Basque Foundation for Science (IKERBASQUE), Bilbao, Spain,* [4] *Centro de Investigación Biomédica en Red (CIBER) of Frailty and Healthy Aging (CIBERfes), Madrid, Spain,* [5] *TransBioNet Thematic Network of Excellence for Transitional Bioinformatics, Barcelona Supercomputing Center, Barcelona, Spain,* [6] *Computational Biology and Bioinformatics, Department Cell and Developmental Biology Max Planck Institute for Molecular Biomedicine, Münster, Germany*

High-throughput cell-data technologies such as single-cell RNA-seq create a demand for algorithms for automatic cell classification and characterization. There exist several cell classification ontologies with complementary information. However, one needs to merge them to synergistically combine their information. The main difficulty in merging is to match the ontologies since they use different naming conventions. Therefore, we developed an algorithm that merges ontologies by integrating the name matching between class label names with the structure mapping between the ontology elements based on graph convolution. Since the structure mapping is a time consuming process, we designed two methods to perform the graph convolution: vectorial structure matching and constraint-based structure matching. To perform the vectorial structure matching, we designed a general method to calculate the similarities between vectors of different lengths for different metrics. Additionally, we adapted the slower Blondel method to work for structure matching. We implemented our algorithms into FOntCell, a software module in Python for efficient automatic parallel-computed merging/fusion of ontologies in the same or similar knowledge domains. FOntCell can unify dispersed knowledge from one domain into a unique ontology in OWL format and iteratively reuse it to continuously adapt ontologies with new data endlessly produced by data-driven classification methods, such as of the Human Cell Atlas. To navigate easily across the merged ontologies, it generates HTML files with tabulated and graphic summaries, and interactive circular Directed Acyclic Graphs. We used FOntCell to merge the CELDA, LifeMap and LungMAP Human Anatomy cell ontologies into a comprehensive cell ontology. We compared FOntCell with tools used for the alignment of mouse and human anatomy ontologies task proposed by the Ontology Alignment Evaluation Initiative (OAEI) and found that the F_β alignment accuracies of FOntCell are above the geometric mean of the other tools; more importantly, it outperforms significantly the best OAEI tools in cell ontology alignment in terms of F_β alignment accuracies.

Keywords: ontology alignment, ontology merging, automatic ontology merging, cell ontology, Human Cell Atlas (HCA), Ontology Alignment Evaluation Initiative (OAEI)

OPEN ACCESS

Edited by:
Carlos Vicario, Consejo Superior de Investigaciones Científicas (CSIC), Spain

Reviewed by:
Catia Pesquita, University of Lisbon, Portugal
Alexander Diehl, University at Buffalo, United States

***Correspondence:**
Daniela Gerovska
daniela.gerovska@biodonostia.org
Marcos J. Araúzo-Bravo
mararabra@yahoo.co.uk

Specialty section:
This article was submitted to Molecular Medicine, a section of the journal Frontiers in Cell and Developmental Biology

Received: *18 May 2020*
Accepted: *05 January 2021*
Published: *11 February 2021*

Citation:
Cabau-Laporta J, Ascensión AM, Arrospide-Elgarresta M, Gerovska D and Araúzo-Bravo MJ (2021) FOntCell: Fusion of Ontologies of Cells. Front. Cell Dev. Biol. 9:562908. doi: 10.3389/fcell.2021.562908

INTRODUCTION

Precision biomedicine technologies produce overgrowing quantities of information from of high throughput data from finer-grained biomedical samples reaching single-cell (Hwang et al., 2018) and subcellular (Grindberg et al., 2013) levels that allow to discover new cell types (Boldog et al., 2018; Gerovska and Araúzo-Bravo, 2019; Sas et al., 2020). This increasingly precise cell data render existing cell classification systems obsolete and create the demand for automatic comprehensive data-driven cell classification methods. Among the structures to classify knowledge domain items are the ontologies; they can be defined in several ways depending on the context of use (Busse et al., 2015). In information science, an ontology is defined as a seven-tuple, O:= {L, C, R, F, G, T, A}, where L:=LC $\bigcup$ LR is a lexicon of concepts LC and relations LR; C is a set of concepts; R is a set of binary relations on C; F and G are functions connecting symbols F: LC $\rightarrow$ C, G: LR $\rightarrow$ R; T is a taxonomy for the partial ordering of C, T(C_i, C_j), and A is a set of axioms with elements C and R (Busse et al., 2015). A critical question in the design of an ontology is the level of detail covered by the ontology. Thus, different ontologies of the same knowledge domain use different conceptualizations to obtain the desired level of granularity. In the case of cell ontologies, there are several cell type classifications in various formats; the most frequently used being the Web Ontology Language (OWL) (Smith et al., 2004) format, that encompass the vast majority of Open Biomedical Ontologies (OBO) Foundry (Smith et al., 2007) ontologies.

Cell classification relies on human data curation, however the growing number of discovered new cell types boosted by high throughput data generation such as single cell RNA-Seq and international research initiatives such as the Human Cell Atlas (HCA) (Rozenblatt-Rosen et al., 2017) creates a necessity to develop automatic computational methods that assist the creation of cell ontologies and the classification of these new cells as branches of existing cell ontologies (Osumi-Sutherland, 2017). New cell ontologies can be created by reusing and merging the information dispersed in multiple cell ontologies. Before merging two ontologies, it is necessary to find the correspondences between their concepts in a process named ontology alignment or matching. There exist numerous tools for the alignment and merging of ontologies (**Table 1**). The majority of them are semi-automatic since they require an initial input and some intermediate user inputs for performing the alignment; some tools focus only on the alignment.

In order to minimize human supervision of the ontology alignment and automate the ontology merging, we developed an algorithm and implemented it into FOntCell, a software package in Python for automatic merging of ontologies. We applied FOntCell to create a new more comprehensive and fine-grained ontology of the cellular development by merging cell ontologies giving rise to all cell types of the human body.

There are multiple ontologies with biomedical information (genomics, proteomics, and anatomy) (Lambrix et al., 2007). Two of the largest cell ontologies are CELDA (Seltmann et al., 2013) and LifeMap (Edgar et al., 2013). CELDA integrates information about gene expression, localization, development and anatomy of *in vivo* and *in vitro* human and mouse cells, as well as cell development. Therefore, we focused on the "development" annotation information of CELDA stored in the fields CL (Cell Ontology) (Bard et al., 2005), CLO (Cell Line Ontology) (Sarntivijai et al., 2014) and EFO (Experimental Factor Ontology) (Malone et al., 2010). Another important repository for cell information is LifeMap (Edgar et al., 2013); which includes cell type and gene expression annotations of cells in different stages of embryonic development. LifeMap provides contrasted data and enough cell types to be synergistically merged with CELDA, as each might have cell types missing in the other. A hurdle in the merging CELDA and LifeMap is their different labeling systems. Also there are more specific ontologies such as the Cell Ontology for Human Lung Maturation [LungMap Human Anatomy (LMHA)] that is a specific ontology of cells for lung development. Theseontologies use different labels for the same cell type and simple word matching cannot find equivalences. Therefore, it is necessary to align the ontologies (Lambrix and Tan, 2008), i.e., identify the classes of one ontology equivalent to the classes of the other ontology. We developed an algorithm that can find equivalence between two classes from two ontologies, taking into account not only the class labeling but also the internal structure of the ontologies.

MATERIALS AND METHODS

The main steps for ontology merging implemented in FOntCell are file ingestion, ontology parsing, ontology pre-processing, alignment, and merging. The internal relations of the ontologies that will be merged and the alignment parameters are specified in a configuration file (**Figure 1A**). The format of the configuration file is described in detail in the **Supplementary Material**. An instance of the configuration file for the merging of CELDA and LifeMap and their result with LMHA are also provided in the **Supplementary Material**. The equivalent classes are detected by a combination of name matching and graph-topology/structure similarity matching (**Figure 1B**). The merging works through expansion of the non-common relationships/edges branching from the equivalent classes. FOntCell searches for similar (to match them) and different (to append them during the merging) classes (**Figure 1C**).

Ontology Parsing

The parsing of the input ontologies generates two two-column matrices, *A2* and *B2*, each one with a number of rows equal to the number of class relations in the respective ontology. The first column contains the name of each class, and the second column, the name of one of its children. The matrices *A2* and *B2* are needed for the structure-mapping. FOntCell can merge any two ontologies in an .owl file in an OWL format, or in .ods files in parent-child relationship format compatible with the pyexcel-ods Python module. Additionally, FOntCell can read as input the matrices *A2* and *B2* in tabulated text files.

Ontology Pre-processing

FOntCell can merge ontologies that share some classes and knowledge domain; however, our main interest is to

TABLE 1 | Tools for the alignment and merging of ontologies and their features, adapted from Table 9 from Lambrix and Tan (2006).

Tool	Linguistic	Structure	Constraints	Auxiliary	Automatic	Merging	References
ArtGen	Name	Parents, children		WordNet	Semi or fully		Mitra and Wiederhold, 2002
ASCO	Name, label, description	Parents, children, siblings, path from root		WordNet	Fully		Le et al., 2004
Chimaera	Name	Parents, children			Semi	Merging	McGuinness et al., 2000
FCA-Merge	Name				Semi	Merging	Stumme and Maedche, 2001
FOAM	Name, label	Parents, children	Equivalence		Semi		Ehrig and Staab, 2004
GLUE	Name				Semi		Doan et al., 2004
HCONE	Name	Neighborhood		WordNet	Semi	Merging	Kotis and Vouros, 2004
IF-Map		Parents, children		Reference ontology	Semi		Kalfoglou and Schorlemmer, 2003
iMapper			Domain, range	WordNet	Semi		Su et al., 2004
Onto Mapper	Name	Parents, children			Semi		Prasad et al., 2002
Anchor-PROMPT	Name	Direct graphs			Semi	Merging	Noy and Musen, 2000
SAMBO	Name, synonym	Is-a a part-of, descendants & ancestors		WordNet UMLS	Semi	Merging	Lambrix and Tan, 2006
S-Match	Label				Fully		Giunchiglia et al., 2004
AML	Label, instances	Direct graph, logical repair algorithm		WordNet	Fully		Faria et al., 2013
LogMap	Label, name	Linguistic alignment, principle of locality		WordNet, UMLS-lexicon	Semi or fully		Jiménez-Ruiz and Cuenca Grau, 2011
AGM	Name, label	Graphs			Semi or fully		Lütke, 2019
ALIN	Label			Wordnet	Semi		da Silva et al., 2019
DOME	Label	doc2vec			Fully		Hertling and Paulheim, 2019
FCAMap-KG	Label, synonym	Part-of			Semi or fully		Zhao et al., 2018
Lily	Name, label	Direct graphs			Semi or fully		Wang and Xu, 2008
LogMapBio	Label, name	Linguistic alignment, principle of locality		WordNet, UMLS-lexicon, BioPortal	Semi or fully		Jiménez-Ruiz, 2019
LogMapLite	Label, name			WordNet, UMLS-lexicon	Semi or fully		Jiménez-Ruiz, 2019
POMAP++	Lame, label		Ontology attribute, linguistic match		Semi		Laadhar et al., 2017
FOntCell	Label, synonym	Direct graphs, attribute relation	Ontology attribute, linguistic match		Fully	Merging	This work

Tool: Algorithm name. Linguistic: type of data used by the linguistic based method. Structure: type of data used by the structure based method. Constraints: type of data used to perform a constraint-based alignment. Auxiliary: external tool used to improve the alignment. Automatic: automation level; fully, if only is required an initial input; semi, if some additional intermediate user inputs are required. Merging: if merging is done in addition to the alignment. Tools like ATOM (Raunich and Rahm, 2011) are omitted since they are ontology merging methods that require a mapping as input and they do not use alignment methods.

apply FOntCell to augment cell ontologies by merging known cell ontologies. Different ontologies use diverse description formats and data structures; additionally, some ontologies are ill-formed with redundant (duplicates) or missing relationships (disconnected branches). FOntCell robustly merges such ill-formed ontologies. However, we implemented an optional pre-processing functionality allowing to repair the input ontologies, select the data relation type (instances, children, parents) used as an input argument, and edit labels to modify the original ontology relationships by addition, deletion and/or merging of classes and relationships.

The pre-processing stage of FOntCell is implemented as an "Automatic ontology editor" that takes as an input an ontology edition .txt file (format given in the **Supplementary Material**) describing the pre-processing modifications of the pre-processed ontology. FOntCell uses such description to modify the class names and/or rewire the ontology. Among the implemented pre-processing functions of the "Automatic ontology editor" are: (a) Selection of classes by their name, i.e., select classes including

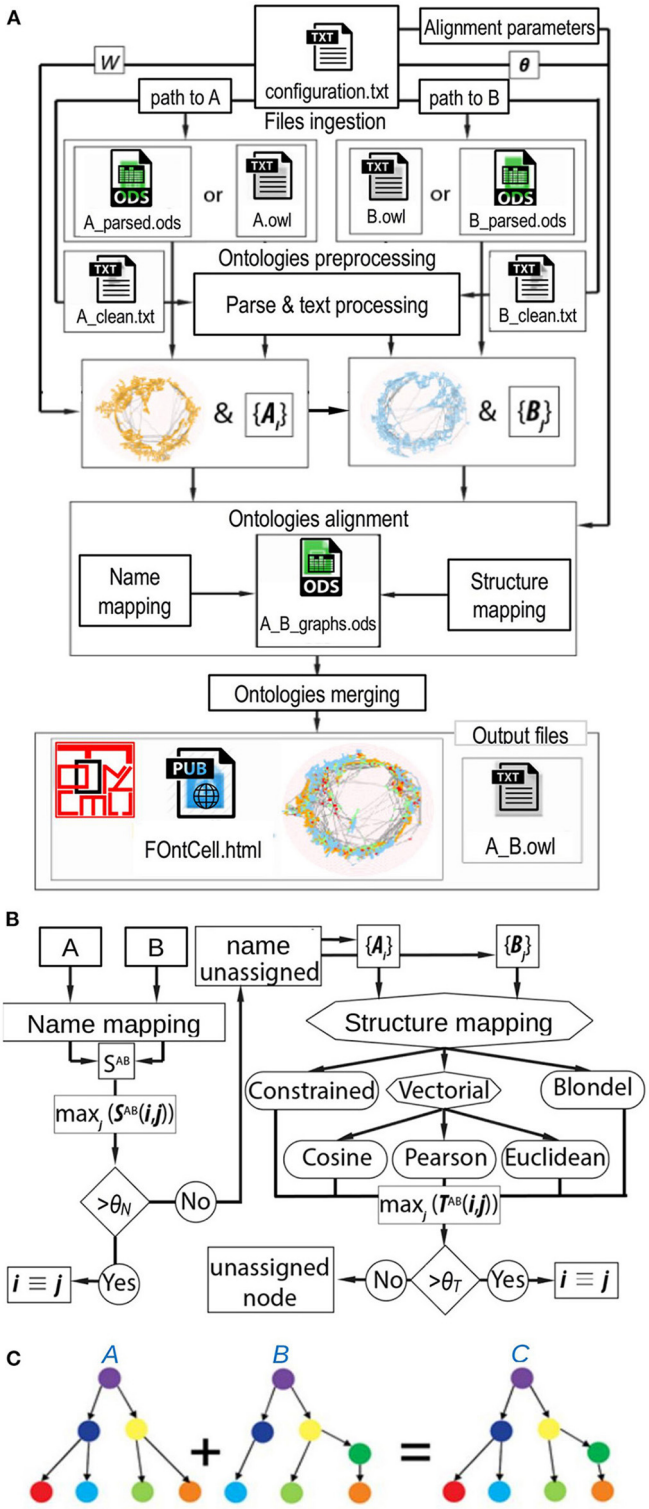

FIGURE 1 | FOntCell algorithm. **(A)** FOntCell software flux diagram with the main functionalities of files ingestion, ontologies preprocessing, ontology parsing, ontologies alignment, ontologies merging, and generation of output files. Together with the ontology files, the user feeds the alignment parameters: *W*, window length, and the similarities threshold vector $\theta = \{\theta_N, \theta_T, \theta_{LN}\}$. **(B)** Flux diagram of the FOntCell alignment algorithm combining the name mapping (left) and the structure mapping (right) using five alternative mapping methods. $\{A_i\}$ and $\{B_j\}$ denote the sets of subgraphs around nodes *i* and *j* of ontology A and B, respectively. The rhombi and octagons mark two or three alternative decisions, respectively. **(C)** Conceptual example of merging of ontologies. For merging two ontologies A and B into an ontology C, FOntCell aligns equivalent classes between A and B, and then merges the non-common relations that branch from the equivalent classes. Equivalent classes are marked with same colors in the two ontologies A and B.

TABLE 2 | Sizes of the processed ontologies before and after preprocessing.

	Before pre-processing		After pre-processing	
	#Classes	#Relations	#Classes	#Relations
CELDA	15,439	203,058	841	966
LifeMap	796	924	796	924
CELDA + LifeMap	1,408	1,855	–	–
LMHA	80	130	45	65
(CELDA + LifeMap) + LMHA	1,437	1,919	–	–

a certain word. (b) De-selection of classes to exclude classes from the resulting ontology. (c) Connection of two previously unrelated classes. (d) Merging of two classes with the resulting class "inheriting" the ID of one of the classes, and the relations and attributes of both classes. (e) Class label preconditioning to edit the class names and class synonyms, and eliminate general terms such as "the," "cell," "cells," "human," "mouse" or "-." The "Automatic ontology editor" was used to pre-process the ontologies in this work as follows.

CELDA Pre-processing

CELDA uses information from other ontologies and its original structure is disconnected; split into several trees, and contains information related to tissues, immortal cell lines, species, etc. Since we are only interested in the developmental cell type information, we need to parse to cell types. Thus, to generate a connected graph of cell development from CELDA, we used the "Automatic ontology editor" to: (1) Introduce new relationships between classes to eliminate discontinuities, some of which are due to the word "human" or "mouse" in the name of their classes. (2) Merge duplicated classes due to the same cell type appearing simultaneously from "human" and "mouse". (3) Eliminate the classes associated to immortalized cell lines selected as they contained "human" or "mouse" in the name, label or synonyms.

LifeMap Pre-processing

The pre-processing of LifeMap is required since LifeMap is an online database in non-OWL format, where its information about cell development is available at the LifeMap website repository (Edgar et al., 2013). We automatically searched the LifeMap website and obtained all the information related to cell name and synonyms, development hierarchy and cell localization and saved it to a two-column matrix file in .ods format. We used the "Automatic ontology editor" to perform string normalization by eliminating the symbols {"-," "/," ","} and the words {"cell," "cells," "human"} from cell names and synonyms.

LMHA Pre-processing

LMHA presents in its "natural" ontology a series of cell types from the lung, but without a direct relationship in the cell development of the tissue itself. We used the "Automatic ontology editor" to: (1) Remove classes that do not provide information about specific cell types such as the "immune cell" or "cell type" classes. (2) Provide some new relations and synonyms.

The sizes of the processed ontologies before and after preprocessing and shown in **Table 2**. The same pre-processed ontologies files were used in the rest of the work by all the merging ontology tools.

Calculation of the Name Mapping Matrix

Before performing the intra-ontology name matching, FOntCell processes the string of each label class of each ontology. Among other string processing tasks, FOntCell performs string normalization, removes mismatching words, splits words, selects substrings, selects only the class name, or optionally uses lists of synonyms representing variation of the class names. Next, it builds a name mapping matrix $S^{\mathrm{AB}}\left(a \times b\right)$, where a and b are the number of classes of ontologies A and B, respectively. Each element $S^{\mathrm{AB}}(i,j)$ is a measurement of the similarity between the labels of class i from ontology A and class j from ontology B. Additionally, the user can trigger a FOntCell option that takes into account synonym attributes for the calculation of the name mapping matrix S^{AB}.

In the simplest case of not activating the option of using synonyms, to measure the similarity between each class label of two ontologies A and B, FOntCell builds a name mapping matrix, S^{AB}, based on the Levenshtein metric (Levenshtein, 1966), which measures the minimum number of insertions, deletions and necessary replacements to make two strings equal. To obtain the similarity in the range [0, 1], we use the opposite of the scaled Levenshtein metric:

$$S^{AB}\left(i,j\right) = 1 - \frac{\mathrm{lev}\left(\mathrm{Label}_{\mathrm{i}}^{\mathrm{A}}, \mathrm{Label}_{\mathrm{j}}^{\mathrm{B}}\right)}{\max\left(\left|\mathrm{Label}_{\mathrm{i}}^{\mathrm{A}}\right|, \left|\mathrm{Label}_{\mathrm{j}}^{\mathrm{B}}\right|\right)} \tag{1}$$

where *lev* is the Levenshtein distance between two strings. For two strings a and b of lengths $|a|$ and $|b|$, respectively, the Levenshtein distance $lev(|a|,|b|)$ is:

$$lev\left(|a|,|b|\right) = \begin{cases} max\left(i,j\right) & if\ \min\left(i,j\right) = 0, \\ \min\begin{cases} lev\left(i-1,j\right)+1 \\ lev\left(i,j-1\right)+1 \\ lev\left(i-1,j-1\right)+1_{(a_i \neq b_j)} \end{cases} & otherwise. \end{cases} \tag{2}$$

where $1_{(ai \neq bj)}$ is the indicator function equal to 0 when $a_i = b_j$, and equal to 1 otherwise, and $lev(i,j)$ is the distance between the first i characters of a and the first j characters of b. $Label_i^A$ and $Label_j^B$ are the labels of the class i and j of the ontologies A and B, respectively, and $|\ |$ is the length of the string. Applying the pairwise Equation (1) for each stripped label class i of A, and the stripped label class j of B, FOntCell builds the name mapping matrix S^{AB} between A and B.

In the case of selecting the option to use the classes synonyms, the similarity between two classes i, j is calculated using the lists of synonyms, $\{i\}\in$A and $\{j\}\in$B that include also the principal label of the class. With these lists is calculated a name matching matrix $S^{\{i\}\{j\}}\left(|\{i\}| \times |\{j\}|\right)$ between each synonym of class list $\{i\}$ and each synonym of class list $\{j\}$ based on the Levenshtein

distance (Equation 2), $|\{i\}|$ and $|\{j\}|$ are the lengths of the lists $\{i\}$ and $\{j\}$, respectively. Finally, the highest score of $S^{\{i\}\{j\}}$ as the matching between the two classes is taken to be used in the final name mapping matrix $S^{\mathrm{AB}}(i,j) = \max S^{\{i\}\{j\}}$. FOntCell considers that two labels have a name matching, if their score given by Equation (1) is greater than a name score threshold θ_N (default 0.85).

Calculation of the Structure Mapping Matrix

Not all classes of an ontology are identifiable as classes of the other ontology by name mapping. For example, in the CELDA and LifeMap merging, when using only the name mapping, approximately 60% of the classes from CELDA are initially not assigned to LifeMap. One of the functionalities of FOntCell is to recognize matches between two ontologies and merge them into a unique class, i.e., two labels of two classes having very different name labeling but corresponding to the same concept. FOntCell discovers synonymous classes between two ontologies using structure mapping, i.e., two classes match if the subgraphs corresponding to their descendants have similar structures.

To relate the nodes of two ontologies, FOntCell extracts a local subgraph centered on each node, that we name generator nodes i from ontology A and generator nodes j from ontology B. The set of all subgraphs from the ontologies A and B are designated as {A} and {B}, respectively. The subgraphs extracted from the generators nodes i and j are denoted $\{i\}$ and $\{j\}$, respectively. The size of these subgraphs is given by the parameter W, which indicates the number of upstream and downstream relationships from the generator node that FOntCell takes to create the subgraphs (**Figure 2A**).

FOntCell measures the structure similarity mapping between two graphs using different methods to build a structure mapping matrix $T^{\mathrm{AB}}\left(a \times b\right)$, where a and b are the number of classes of ontologies A and B, respectively. Once a window length W (default 4) is selected, for each node i from A FOntCell constructs the surrounding subgraph of nodes $\{i\} \in A_w(i)$ and calculates its similarity with all subgraphs $\{j\} \in B_w(j)$, where $A_w(i)$ and $B_{\mathrm{w}}(j)$ are the subgraphs of length W centered in i and j, respectively. Each subgraph $A_w(i)$ is defined by a center node i and all the nodes inside a window length W upstream or downstream of i. Thus, FOntCell performs a structure convolutional matching, tailoring different metrics to calculate the similarity between subgraphs $A_w(i)$ and $B_{\mathrm{w}}(j)$.

The Blondel method (Blondel et al., 2004), initially developed to measure the similarity between graph vertices can be used to assess the structure matching between two networks, however it is quite computationally demanding (**Figure 3C**). To improve the speed of the structure mapping, we designed two new methods that calculate the structure matching of ontologies in a convolutional fashion: Vectorial structure matching and Constraint-based structure matching; additionally, we adapted the Blondel method to work for such new convolutional structure matching approach. An example of a convolution window sliding across a graph is depicted in **Figure 2B**.

FOntCell takes as generator nodes those without name assignment during the calculation of the name mapping matrix S^{AB}. The subgraphs generated from these nodes from both ontologies are evaluated using one of the metrics explained below.

Vectorial Similarity Based on Graph Convolution as a Structure Matching Metric

For each possible pair of nodes $i \in$ A and $j \in$ B, and window length W, FOntCell extracts the subgraphs $A_W(i)$ and $B_W(j)$ of length W centered in i and j, and adjacency matrices $\tilde{A}_W(i)$ and $\tilde{B}_W(j)$ with number of nodes a_{Wi} and b_{Wj}, respectively. For all possible pairs of nodes $k \in A_W(i)$ and $l \in B_W(j)$, FOntCell takes the corresponding rows $\tilde{ik}$ and $\tilde{jl}$ of the adjacency matrices $\tilde{A}_W(i)$ and $\tilde{B}_W(j)$, and calculates their similarity using one of the $M = \{1 - \text{cosine}, \text{Euclidean}, 1 - \text{Pearson}\}$ metrics. Since the lengths a_{Wi} and b_{Wj} of those rows are not necessarily equal, FOntCell calculates all nc possible convolution similarities, $p^c_{ik,jl}$ of the shorter row over the longer row, where $nc = \mathrm{abs}(a_{Wj} - b_{Wi}) + 1$ is the number of convolutions and $c \in [1, nc]$ (**Figure 2B**), and selects the maximum similarity: $p^{Max}_{ik,jl} = \max p^c_{ik,jl}$. Then, for each (i,j) pair, it assigns to the (i,j) position of the structure mapping matrix $T^{AB}(i,j)$ the maximum similarity $p^{Max}_{ik} = \max p^{Max}_{ik,jl}$ for jl across $\tilde{B}_W(j)$. For brevity, throughout the whole text, we name the vectorial structure matching using the $(1 - \text{cosine})$ and $(1 - \text{Pearson})$ metrics as cosine and Pearson structure matching, respectively.

Constraint-Based Similarity as a Structure Matching Metric

We developed the constraint-based structure matching based on three assumptions: (i) The matches obtained from the name matching are correct. (ii) The degree of structural similarity of two generator nodes is proportional to the number of matches between the subgraphs generated by them. (iii) Two generator nodes are more likely to be equivalent if their close relatives have name matches.

To calculate the similarity between two generator nodes, i and j, FOntCell first obtains the number of name matches between the two subgraphs $\{i\}$ and $\{j\}$. Next, it weighs them according to the proximity to the generator nodes i or j. The matches closer to the generator node score higher. To implement the method, for all possible pairs of nodes $i \in$ A and $j \in$ B, and for a window length W, FOntCell searches for all possible name matched pairs between the lists of nodes $\{k\} \in A_w(i)$ and the list $\{l\} \in B_w(j)$, where $A_w(i)$ and $B_w(j)$ are the subgraphs of length W centered in the generator nodes i and j, respectively, with the condition that at least one node of the list $\{k\} \in A_w(i)$ has a name match with a node of the list $\{l\} \in B_w(j)$. Then, for each node k in the list $\{k\}$, FOntCell calculates the shortest path s_{ki} to i, and produces a list $\{s_{ki}\}$ of shortest paths to estimate the proximities to the generator node i. FOntCell assigns to each shortest path s_{ki} a constraint $c_{ki} = W + 1 - S_{ki}$. Finally, it sums the list of constraints $\{c_{ki}\}$ to produce an accumulated constraint C_i and assigns it to the structure mapping matrix $T^{\mathrm{AB}}(i,j)$.

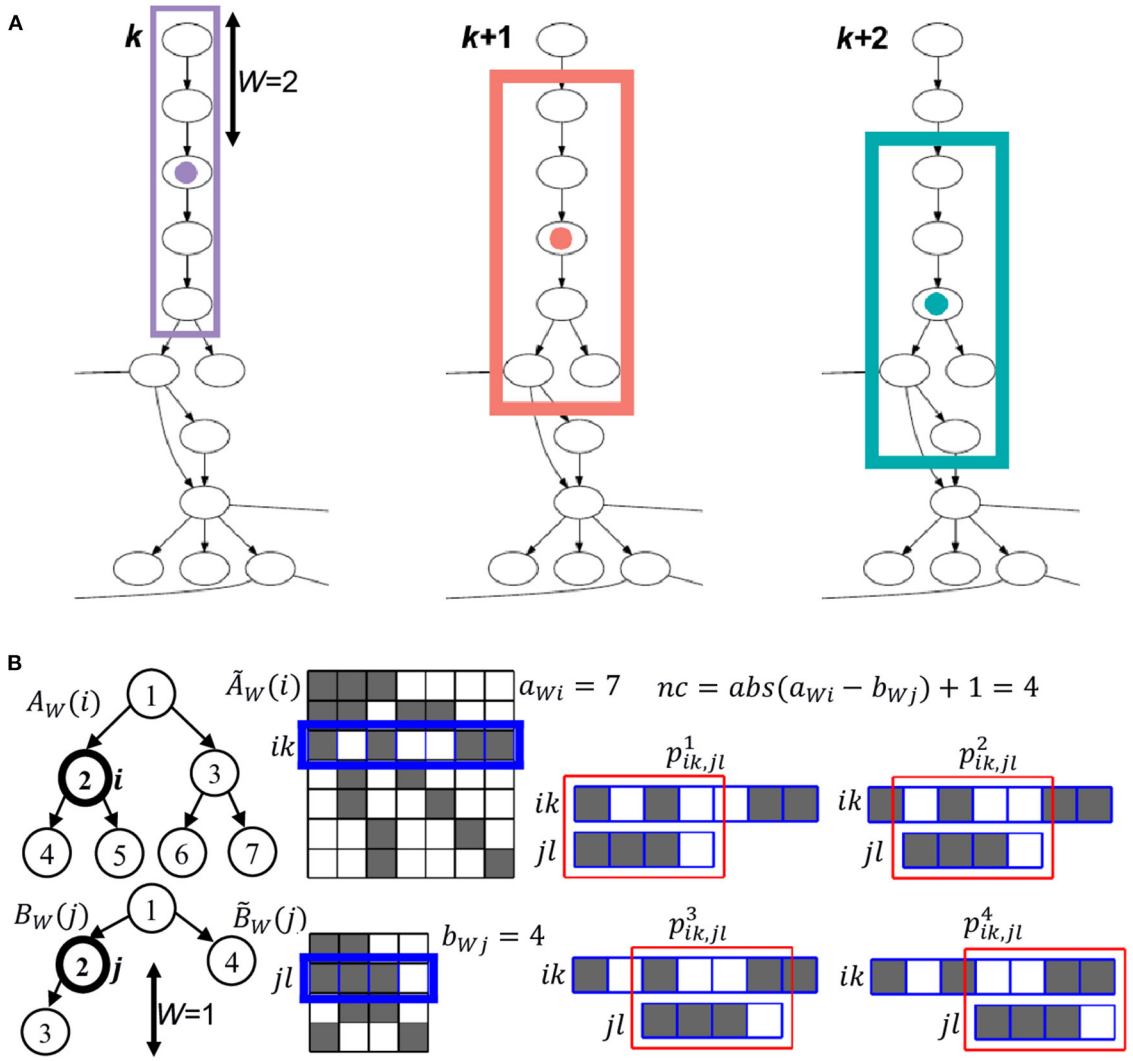

FIGURE 2 | Convolutional graph matching of FOntCell. **(A)** Example of three consecutive steps of the sliding window of length $W = 2$ used in the calculation of the structure convolutional matching. For each central (generator) node, marked with a colored circle, the nodes involved in the calculation of the structure convolutional matching are framed with a rectangle of the same color as its corresponding central node. **(B)** Example of graph convolution, for a sliding window of length $W = 1$, between two subgraphs $A_W(i)$ and $B_W(j)$ (left) with generator nodes i and j, adjacency matrices $\tilde{A}_W(i)$ and $\tilde{B}_W(j)$ (center), and number of nodes $a_{Wi} = 7$ and $b_{Wj} = 4$, respectively. The connected nodes are represented by dark cells in the adjacency matrices. For each row k of $\tilde{A}_W(i)$, and l of $\tilde{B}_W(j)$, a vectorial convolution is calculated. The step for the rows $k = 3$ of $\tilde{A}_W(i)$, and $l = 2$ of $\tilde{B}_W(j)$, is marked in blue as an example. The $nc = \text{abs}(a_{Wj} - b_{Wi}) + 1 = 4$ sliding windows of the shorter row jl of $\tilde{B}_W(j)$ over the longer row ik of $\tilde{A}_W(i)$ are marked in red (right), and the respective nc convolution similarities $p^c_{ik,jl}$ for each slide c are calculated using one of the metrics $M = \{1 - \text{cosine}, \text{Euclidean}, 1 - \text{Pearson}\}$.

Blondel Similarity as a Structure Matching Metric

To perform graph structure matching FOntCell adapts the original Blondel metric:

$$T^{AB}_{k+1} = \frac{\tilde{B} T^{AB}_k \tilde{A}^t + \tilde{B}^t T^{AB}_k \tilde{A}}{\left\| \tilde{B} T^{AB}_k \tilde{A}^t + \tilde{B}^t T^{AB}_k \tilde{A} \right\|}, \quad (3)$$

where t is the transpose operator. Equation (3) is calculated iteratively until an even number of steps k of convergence to a stable structure matching T^{AB} is reached. As with the other metrics, FOntCell takes sets of subgraphs generated from each generator node and calculates the similarity between these nodes using the Blondel metric. For each node i from A, FOntCell constructs the surrounding subgraph $\{i\} \in A$ and calculates its similarity with all subgraphs $\{j\} \in B$ using Equation (3) with the adjacency matrices of each subgraph. FOntCell performs a structure convolution, tailoring Equation (3) to the case of

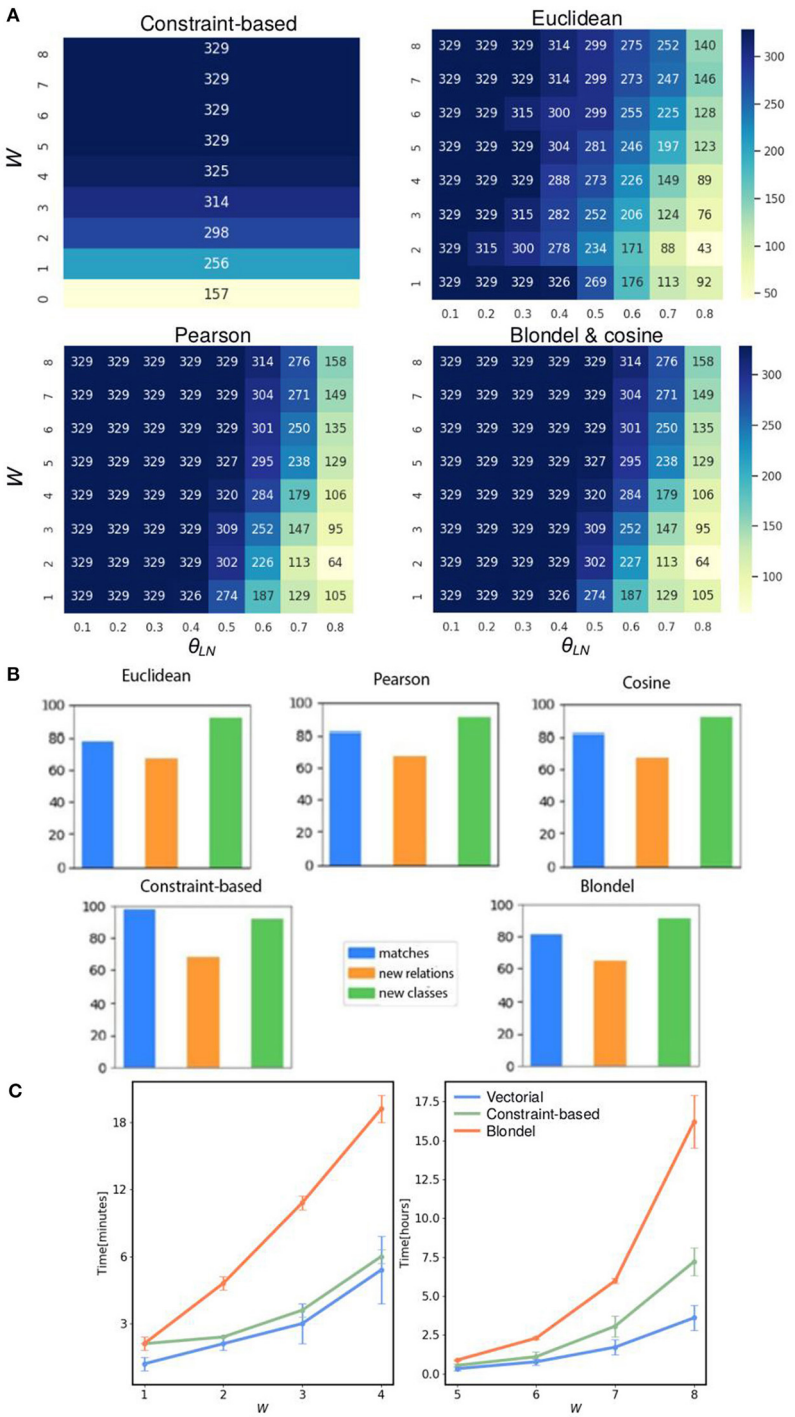

FIGURE 3 | FOntCell performance merging CELDA with LifeMap. **(A)** Heat maps of the matches obtained with two-parameter combinations, window length and name score threshold, using five structure matching methods: the three vectorial structure matching {cosine, Euclidean, Pearson}; constraint-based structure matching, and Blondel structure matching. The two optimized parameters are the window length W and the local sequence threshold θ_{LN} in the ranges [0.1, 0.8] and [1, 8], respectively, using steps of 0.1 for θ_{LN}, and 1 for W. Bluer color corresponds to higher number of synonyms. **(B)** Percentages of matches, new classes and new relations, obtained with the five structure matching methods with merging alignment parameters $W = 4$, $\theta_N = 0.85$, and $\theta_{LN} = 0.7$. **(C)** Run time for the five structure-matching methods for $\theta_N = 0.85$, and $\theta_{LN} = 0.7$, and window sizes W in the range [1, 8]. The vectorial structure matching {cosine, Euclidean, Pearson} have similar run time lines and are represented by a single line.

subgraphs {*i*} and {*j*}.

$$T_{k+1}^{\{i\}\{j\}} = \frac{\{\tilde{j}\}T_k^{\{i\}\{j\}}\{\tilde{i}\}^t + \{\tilde{j}\}^t T_k^{\{i\}\{j\}}\{\tilde{i}\}}{\left\|\{\tilde{j}\}T_k^{\{i\}\{j\}}\{\tilde{i}\}^t + \{\tilde{j}\}^t T_k^{\{i\}\{j\}}\{\tilde{i}\}\right\|} \quad (4)$$

where $\{\tilde{i}\}$ and $\{\tilde{j}\}$ are the adjacency matrices of the respective subgraphs {*i*} and {*j*}. Finally, the structure score on the position of *i* and *j* in the $T_{k+1}^{\{i\}\{j\}}$ matrix is assigned to $T^{\mathrm{AB}}(i,j)$.

Each of the above defined structure matching methods returns a structure score between two nodes (*i,j*) defined by $T^{\mathrm{AB}}(i,j)$ where $i \in$ A and $j \in$ B. This convolution improves the result of the structure mapping over the whole graphs since it reduces the influence of distant nodes and edges. Name mapping and structure mapping carry complementary information and FOntCell regains information from both.

Ontology Alignment

To match classes, FOntCell initially selects the best match for each node *i* from ontology A with a node *j* from ontology B, using the name mapping matrix S^{AB}. If $S^{AB}(i,j) > \theta_N$, FOntCell considers classes *i* and *j* as matched and classifies this assignment as a 'name match'. If $S^{\mathrm{AB}}(i,j) \leq \theta_N$, FOntCell takes the element $T^{\mathrm{AB}}(i,j)$ from one of the aforementioned structure matching methods selected by the user to calculate the structure mapping and considers the nodes *i* and *j* matched if $T^{\mathrm{AB}}(i,j) \geq \theta_T$, where θ_T is a structure mapping threshold selected by the user.

Ontology Local Name Matching

To improve the result achieved with the structure matching method, FOntCell performs a further local name comparison using the name mapping matrix S^{AB} to calculate the mean of the name match $S^{\{i\}\{j\}}$ of each subgraph pair {*i*} {*j*}, with the same window size *W* used to calculate $T^{\{i\}\{j\}}$. FOntCell takes the best name scores from $S^{\mathrm{AB}}(i,j)$, calculates the mean of these name matching scores for the pair {*i*} {*j*}, and then builds a new name matching matrix of {*i*}{*j*}: $S^{\{i\}\{j\}}$. If $S^{\{i\}\{j\}} > \theta_{LN}$, where θ_{LN} is a local name matching threshold (default value $\theta_{LN} = 0.7$), FOntCell considers nodes *i* and *j* as synonyms and classifies the corresponding classes as a structure match (**Figure 1B**). FOntCell creates a file with the relevant information about each node from A with five columns: (1st) native node label in A, (2nd) translated node label assigned from B, (3rd) name score, (4th) structure score, and (5th) type of assignment (Name/Structure). In case of no assignment, the type of assignment is marked as Non-matched.

Ontology Merging

Once the matched classes between two ontologies are detected, FOntCell translates the name/labels of all classes from ontology B to their equivalent names, if any, in ontology A. Next, FOntCell appends the translated classes from B to A and their corresponding offspring relationships. Then, it performs an ordered-set operation to eliminate all the possible class-relations repeats generated from the appendage. The resulting relation array represents the merging of the two ontologies. In addition, FOntCell creates an OWL format file with the result of the merging by reading the .owl file of ontology A and appending the new classes from B at the start of the ontology class site. The information of these new classes is stored in four columns: (1st) new ID, (2nd) class label, (3rd) class synonyms, and (4th) ascendant relationship. Finally, FOntCell creates an .html file with an interactive circular Directed Acyclic Graph (DAG) of the original and merged ontologies, and statistical information of the merging, i.e., percentage and number of added classes/relations and type of matches in textual and graphical form.

Alignment Performance Scores

To evaluate the performance of FOntCell during alignment and to compare it with other alignment methods, we used the Precision (Equation 5), Recall (Equation 6) and Accuracy (Equation 7) in terms of Type I and Type II errors:

$$\text{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \quad (5)$$

$$\text{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \quad (6)$$

$$F_\beta = \frac{\left(1+\beta^2\right)\cdot \text{Precision} \cdot \text{Recall}}{\left(\beta^2 \cdot \text{Precision}\right) + \text{Recall}} = \frac{\left(1+\beta^2\right)\cdot \mathrm{TP}}{\left(1+\beta^2\right)\cdot \mathrm{TP} + \beta^2 \cdot \mathrm{FN} + \mathrm{FP}} \quad (7)$$

where β is real positive number that accounts for how many times the precision is considered more important than the recall in the measurement of the accuracy, and TP, FP and FN are the numbers of True Positives, False Positives and False Negatives, respectively. We calculated three accuracies: F_1, harmonic mean of the precision and the recall; $F_{0.5}$ which gives double weight to the precision compared to the recall, attenuating the false negative influence; F_2, which gives double weight to the recall compared to the precision, giving more emphasis on false negatives.

To assess the alignment performance for the cases of CELDA+LifeMap and CELDA+LifeMap+ LMHA which are *de novo* alignments without reference ones, we built manually the references and used them to compare the performance of all the alignment tools.

RESULTS

Cosine Is the Best Structure Method for the CELDA + LifeMap Merging

To find the optimal parameters of FOntCell for the merging of CELDA with LifeMap, we performed a bidimensional scanning of the alignment parameters: local name threshold θ_{LN} and window length, *W*, in the range [0.1, 0.8] and [1, 8], respectively, using steps of 0.1 for θ_{LN}, and 1 for *W* for all structure mapping metrics: the three vectorial structure matching methods (Euclidean, Pearson, and cosine), the constraint-based structure matching, and the Blondel structure matching (**Figure 3A**). The constraint-based method does not involve local name matching, therefore θ_{LN} was not used. A name mapping threshold $\theta_N = 0.85$ produces an accuracy $F_1 > 0.9$ for almost all the metrics

(**Supplementary Table 1**), thus, we keep $\theta_N = 0.85$ for the rest of the analysis. $\theta_N > \theta_{LN}$ recovers some meaningful cases during the structure mapping and helps to overcome the graph isomorphism problem arising during subgraph comparisons. The name mapping threshold $\theta_N = 0.85$ assigns as similar class labels those labels that differ in orthographic variations, such as "s" endings, apostrophes, etc. Therefore, we set for the remaining analysis $\theta_N > \theta_{LN} = 0.7$ since we expected more name variability in nodes between subgraphs comparisons than in class-to-class comparison. It is important to reduce the θ_{LN} sensitivity since a more sensitive method finds more isomorph subgraphs. Smaller W produces smaller subgraphs, increasing the possibility to slip into isomorph subgraphs making the structure metric more sensitive to θ_{LN}, while for very large W, FOntCell merges unrelated subgraphs of the two ontologies. For the CELDA and LifeMap merging, the window size that minimizes the sensitivity to θ_{LN} is $W = 4$. The constraint-based method with $W = 4$ slips into subgraph isomorphism, i.e., it finds too many synonyms and has higher sensitivity to the change of W than other vectorial methods (**Figure 3A**). The Euclidean method is more restrictive than the other vectorial methods but more sensitive to θ_{LN} (**Figure 3A**). The Pearson and cosine methods produce almost the same number of matches for all combinations of alignment parameters (**Figure 3A**). The cosine method obtains exactly the same number of synonyms as the Blondel method for all pairs of parameters (**Figure 3A**).

To analyze the effect of each of the five structure mapping methods on the percentages of matches, new classes and new relations between them, and to find the best structure mapping method, we performed a FOntCell merging of CELDA and LifeMap for the optimized alignment parameters: $W = 4$, $\theta_N = 0.85$ and $\theta_{LN} = 0.7$, for each structure matching method. We found similar number of classes and relations added by the different structure matching methods, and similar number of matches (**Figure 3B**).

We studied the run time of the five structure mapping methods for the optimized alignment parameters $\theta_N = 0.85$ and $\theta_{LN} = 0.7$, and window sizes W in the range [1, 8], and we found that the vectorial methods (cosine, Euclidean and Pearson) are the fastest, and at least one order of magnitude faster that the Blondel method (**Figure 3C**). Since the vectorial methods are much faster than the Blondel method, and among them the cosine method obtains the same number of synonyms as the Blondel, we chose to use the cosine method in the remaining analysis.

The Merging of CELDA With LifeMap Expanded CELDA by 67%

The merging of CELDA and LifeMap with $\theta_{LN} = 0.7$ and $W = 4$ resulted in an ontology integrating all the 841 classes from CELDA with 567 classes from LifeMap. Thus, the merged ontology increased the cell ontology information of CELDA by 67% (**Figure 4A**) with accuracy $F_1 = 0.9$ (**Supplementary Table 1**). The generated by FOntCell interactive DAGs of CELDA, LifeMap and the resultant merged ontology are presented in **Figure 5**. Zooms of regions where FOntCell performed both name and structure mapping (**Figure 6**) illustrates some of the challenges arising during alignment of cell ontologies, and how the structural matching rescues information from one ontology to augment the other ontology and enhanced the final merged ontology: Two or more classes of CELDA can align with one class of LifeMap, a rather common phenomenon when activating the use of synonyms. In the zoomed regions CELDA (**Figure 6A**) and LifeMap (**Figure 6B**) have similar but not identical structures. CELDA starts with "hypoblast cell," with children "yolk cell" and "extraembryonic endoblast cell," with the latter further having as a child "secondary yolk sac", whereas in LifeMap the same developmental region starts with "hypoblast cell," followed by "extraembryonic endoderm cells," "yolk sac endoderm cells", and finally "allantois cell." The merging shows a consensus (**Figure 6C**) that starts with "hypoblast cell" (as in CELDA and LifeMap), that as child cell has "yolk cell" from (CELDA and LifeMap) and incorporates as an additional child the "extraembryonic endoblast cell" owing to the information provided by LifeMap. From these two children, resulting from the merging of similar but not identical cells follows the "secondary yolk" that additionally incorporates as a child the "allantois cell" owing to the structural information provided by LifeMap.

For a less restrictive pair of parameters $\theta_{LN} = 0.1$ and $W = 1$ using the cosine metric, 39.1% of classes from CELDA have a structure matching in LifeMap independently of the used structure matching method. For more restrictive parameters $\theta_{LN} = 0.7$ and $W = 7$, we obtained 32.2% of classes with structure mapping.

Aligning CELDA and LifeMap, FOntCell Has Precision of 99% With Name Mapping, and Mean Precision of 55% With the Five Structure Mapping Methods

We calculated the precision of the different mapping methods of FOntCell when merging CELDA and LifeMap with the optimal parameters $W = 4$, $\theta_{LN} = 0.7$ and $\theta_N = 0.85$ (**Figure 7A**). The results obtained through name mapping and the different structure mapping methods were validated checking failures, false positives (FP) and successes, true positives (TP) on the matching of the cell types and calculating the precision using Equation (5). The name matching shows 98.63% precision (**Figure 7A**) and has the highest number of matches (**Figure 7B**), 512, in comparison with the other matching methods of FOntCell. Among the structure mapping methods, highest precision of 62.10% is shown by the constraint-based method, followed by the cosine and the Pearson, 56.42%, Blondel, 50.27%, and the Euclidean, 48,99%, methods (**Figure 7C**). Evaluating the whole FOntCell mapping process, name and structure mapping methods taken together, we observe similar total precision with all the methods: ∼87% using the vectorial methods, 86.1% with the Blondel method, and 86% with the constraint-based method. Anyway, the vectorial methods produce higher total precision values due to the contribution of fewer matches than in the constraint-based and in the Blondel methods.

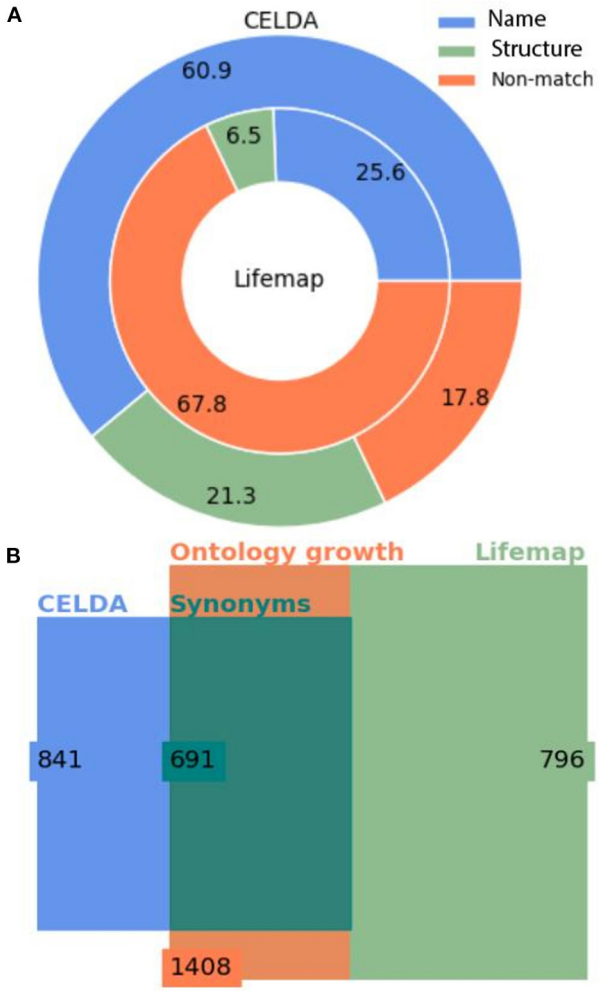

FIGURE 4 | Statistics of the merging of CELDA and LifeMap with the cosine structure matching metric. **(A)** Donut plot of the percentages of classes added by name mapping vs. the classes added by structure mapping to CELDA (outer circle) from LifeMap (inner circle). **(B)** Square Euler-Venn diagram with the number of classes before and after merging. The blue and light green rectangles frame the number of classes in CELDA and LifeMap, respectively, before the merging, the dark green rectangle frames the sum of name and structure equivalent classes, and the orange rectangle frames the total number of classes in the resultant CELDA and LifeMap merged ontology. Alignment parameters $W = 4$, $\theta_{LN} = 0.7$ and $\theta_N = 0.85$.

When considering only structure mapping precision, the Blondel method is the second worst one, slightly better than the Euclidean method (**Figure 7C**). However, when combined with name mapping, all the vectorial methods, including the Euclidean method, surpass the precision of the Blondel method (**Figure 7A**) due to the Blondel method producing more matches during the structure matching than the Euclidean (**Figure 7B**). This indicates that the synergies arising between name mapping and structure method combinations are stronger for the vectorial methods than for the Blondel method, at least in the case of CELDA and LifeMap merging.

Considering only the structure mapping precision, the Euclidean method has the lowest one, 48.99% (**Figure 7C**), while combined with name mapping it has precision of 87.44%, similar to the combined precision of the other vectorial methods, cosine and Pearson (**Figure 7C**), because of the low number of matches obtained during structure matching, which is actually the lowest (**Figure 7B**). This indicates that the synergies arising between name mapping and structure method combinations equilibrate for all the vectorial methods. The constraint-based method contributes the highest number of matches (**Figure 7B**), and although it has the highest precision of 62.1% among the structure mapping methods (**Figure 7C**), it has the lowest total precision among the combined methods (**Figure 7A**).

The Pearson and cosine methods show equal performance, both with the same number of matches, 179 (**Figure 7B**), and the same structure mapping precision of 56.42% (**Figure 7C**), which results a total precision of 87.69% when combined with the name mapping, a total precision of 87.69% when combined with the name mapping (**Figure 7A**). In conclusion, the cosine and Pearson methods in combination with name matching achieve the highest total precision and the smallest number of matches. Therefore, we chose the cosine method as default structure matching method of FOntCell. Anyway, we could have chosen the Pearson structure method with equally good results.

The Merging of CELDA and LifeMap Has an Accuracy F_1 of 0.91

We calculated the precision (Equation 5), the recall (Equation 6) and the family of accuracies F_β: F_1, $F_{0.5}$ and F_2 (Equation 7) with the optimal parameters: $W = 4$, $\theta_{LN} = 0.7$ and $\theta_N = 0.85$, for each of the structure metrics (cosine, Euclidean, Pearson, constraint-based, Blondel), and an additional metric that only measures the string similarity.

All metrics produce CELDA and LifeMap alignments with similar F_β accuracies. The Pearson and cosine obtain slightly higher F_1 and F_2 accuracies than the Euclidean for F_1 and F_2 due to the Euclidean slightly lower recall. The Blondel method exhibits similar to the vectorial methods behavior with a slight decrease in precision and a recall similar to cosine and Pearson. The constraint-based method has lower precision but higher recall (**Supplementary Table 1**).

For all FOntCell alignment methods, the precision ranges between 0.847 and 0.877, and the recall between 0.982 and 0.915 (**Supplementary Table 1**). In the name mapping case (StringEquiv), where structural alignment is not used, the precision is close to 1 but the recall nevertheless decreases considerably (**Supplementary Table 1**); thus the name mapping misses to align numerous classes and many of them are aligned by the structure alignment methods.

After merging the aligned the ontologies, the resulting ontologies with the cosine and Pearson methods have a greater number of matches and a greater growth in the number of classes (**Figure 3C**) compared to the Euclidean method. The parameters that influence more the process in terms of increasing the number of classes and the number of matches in the final ontology are W and θ_{LN}. For all values of θ_N, the number of structure

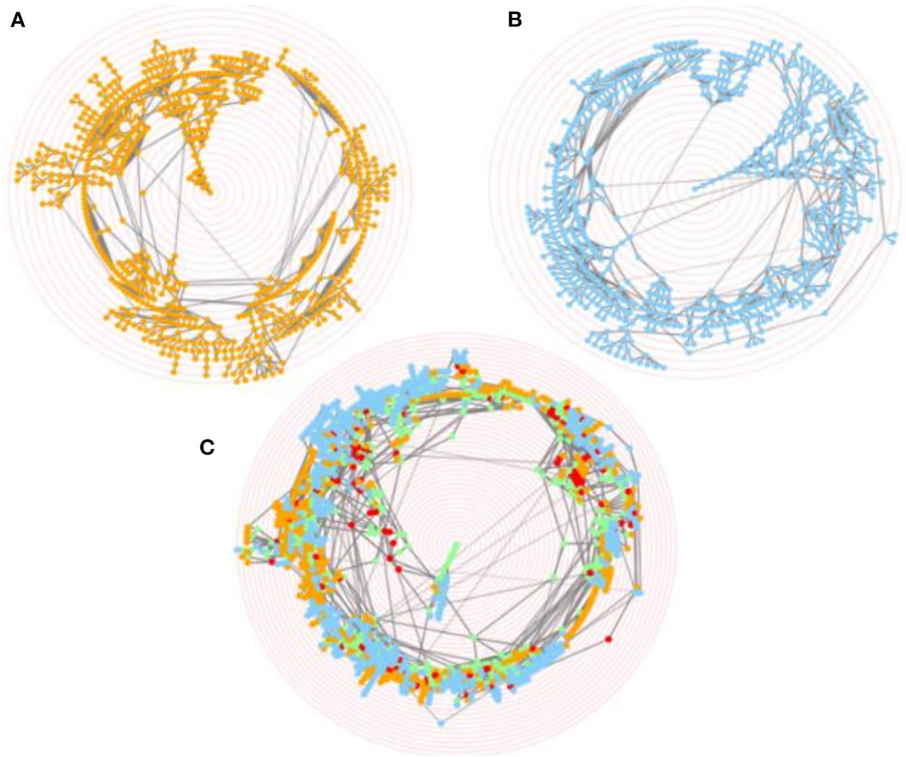

FIGURE 5 | Merging of CELDA and LifeMap ontologies. Screenshots of the interactive circular Directed Acyclic Graphs (DAGs) of **(A)** CELDA, **(B)** LifeMap and **(C)** the merged CELDA + LifeMap ontology, respectively. The orange and blue nodes are the non-matched contributions from ontology A and ontology B, respectively. The green and red nodes are the nodes with name and structure mapping, respectively. The ontology labels associated to the nodes appear when hovering over the nodes. The concentric red rings are zoom guides.

matches found have an inflection point when $W = 4$ and $\theta_{LN} = 0.7$ (**Figure 3A**). This is the midpoint where the method is not restrictive enough and not excessively permissive.

The CELDA + LifeMap merging with the optimal configuration parameters: $W = 4$, $\theta_{LN} = 0.7$, and the cosine vector method found 691 synonyms between the two ontologies and generated a final cell ontology with 1,408 classes, 841 from CELDA and 1,408 – 841 = 567 added classes from LifeMap (**Figure 4B**).

The F_β Alignment Accuracies of FOntCell Are Above the Geometric Mean When Comparing With Other Alignment Tools of the OAEI

To compare the alignment capability of FOntCell with other tools in a different problem, we selected the alignment of mouse and human anatomy ontologies task proposed by the Ontology Alignment Evaluation Initiative (OAEI) in 2019. We used the optimized parameters: $W = 4$, $\theta_{LN} = 0.7$ and $\theta_N = 0.85$, and performed the analysis for all the structural metrics implemented in FOntCell. For all metrics and the whole family of accuracies F_β, FOntCell performs above the geometric mean of the other tools (**Table 3**). The simple name matching, incorporated in FOntCell as a complementary alignment, is more precise but with lower recall, leading to lower F_1 and F_2 but higher $F_{0.5}$ accuracies. The different types of structural alignment of FOntCell find new classes undiscovered by the name mapping alignment.

FOntCell Outperforms Significantly the Best OAEI Tools in the CELDA and LifeMap Alignment in Terms of F_β Alignment Accuracies

We selected the best performing tools that we found during the alignment of mouse and human anatomy ontologies (**Table 3**): StringEquiv, AML and LogMap, and ran them with their default parameters to compare their performance with FOntCell in the case of the alignment of CELDA and LifeMap. They showed higher precision but a lower recall than FOntCell, especially in the case of AML and LogMap. The lower recall penalized their accuracy leading to lower F values (**Table 4**).

The F_1 accuracies of StringEquiv and the different methods of FOntCell alignments are similar, ∼0.9. For the accuracy that gives more weight to the precision, $F_{0.5}$, StringEquiv outperformed the other methods. For the accuracy that gives more weight to the recall, F_2, FOntCell that combines a name matching with a structure matching obtained a better result. Importantly, when merging is oriented at complementing two ontologies, the recall is a key value to be able to rescue many new cell types, and

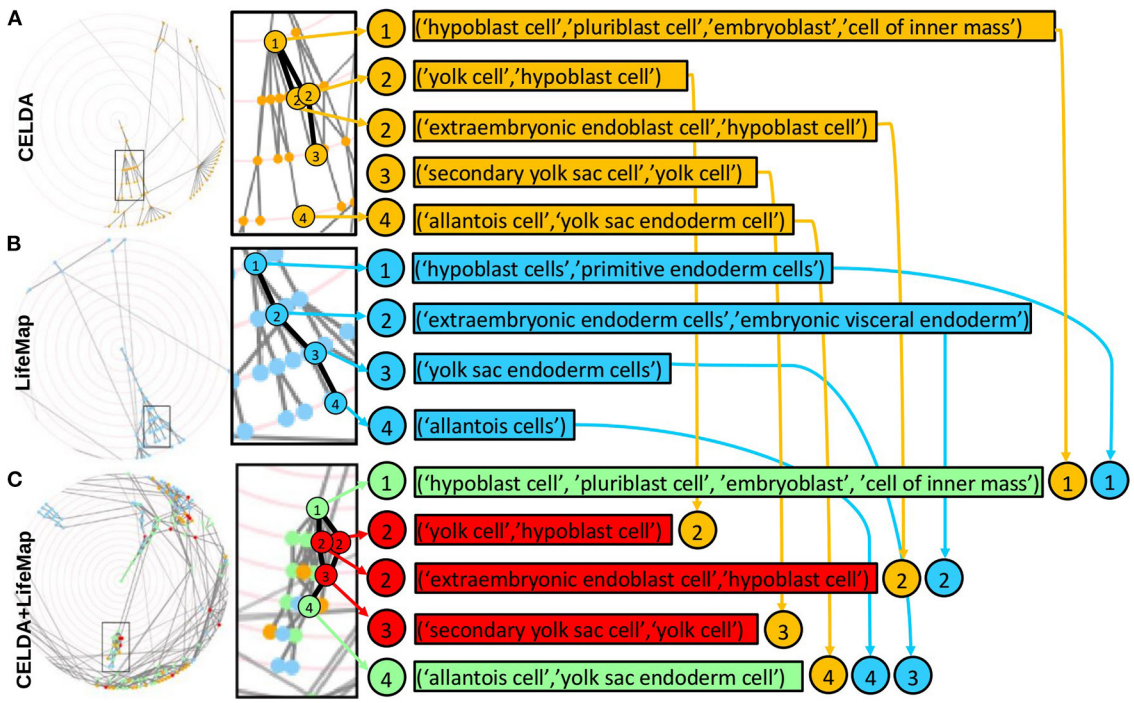

FIGURE 6 | Zooms of regions of CELDA, LifeMap and the merged ontology where FOntCell performs name and structure mapping. (Left) Screenshots of the interactive circular Directed Acyclic Graphs (DAGs) of **(A)** CELDA, **(B)** LifeMap and **(C)** the merged CELDA+LifeMap ontology. (Right) Zoomed regions with corresponding lists of cell types. The synonymous names in each list are separated by commas. The orange and blue nodes are the non-matched contributions from CELDA and LifeMap, respectively. The green and red nodes are the nodes with name and structure mapping, respectively. The numbers inside circles indicate the relative parent-child relationship in ascending order.

all metrics of FOntCell outperform the other methods in this case. Noteworthy, for the whole family of accuracies F_β, all FOntCell metrics outperform significantly the best OAEI tools in the CELDA and LifeMap alignment (**Table 4**).

The Merging of CELDA + LifeMap With LMHA Generates 65 New Relations and 39 New Classes

One of the applications of FOntCell is to merge an ontology from a broad, general description, with another ontology with very specific knowledge within the same knowledge domain. In order to illustrate this functionality, we merged the ontology resulting from CELDA + LifeMap merging with LMHA, a specific ontology of cells for lung development starting ∼36 weeks of human fetal gestation and continuing after birth with some variation in when the alveolar stage commences and when it is complete. The .owl file used in the merging was generated by Susan E Wert, Gail H. Deutsch, Helen Pan, and the National Heart, Lung and Blood Institute (NHLBI) Molecular Atlas of Lung Development Program Consortium Ontology Subcommittee (LungMAP) [U01HL122642] and downloaded from (www.lungmap.net) of the LungMAP Data Coordinating Center (1U01HL122638) of the NHLBI, on April 7, 2018. The merging of CELDA + LifeMap and LMHA produced 65 new relations and 39 new classes related to endothelial and lymphoid cells (**Figure 8**).

DISCUSSION

The discovery of new cell types such as those produced by the HCA consortium or their better characterization by single cell transcriptomics (Gerovska and Araúzo-Bravo, 2016) can render old cellular development ontologies obsolete. We developed FOntCell to address this problem with a novel algorithm that by merging ontologies adds new relationships and classes to a base ontology. Such algorithm allows us to construct from two ontologies a cell ontology that is as complete and up-to-date as possible. We implemented FOntCell as a new Python module that merges efficiently ontologies in the same or similar knowledge domains. It processes intra- and inter- ontology synonyms. To process intra-synonyms, its name similarity search engine is equipped with a name list processing functionality. To search for inter-ontology synonyms, FOntCell integrates the name similarity search engine with a structural similarity search based on graph convolution. Since the structural similarity assessment is a lengthy process that takes the highest percentage of the running time of the merge process, to perform the graph convolution we designed two methods to perform structural

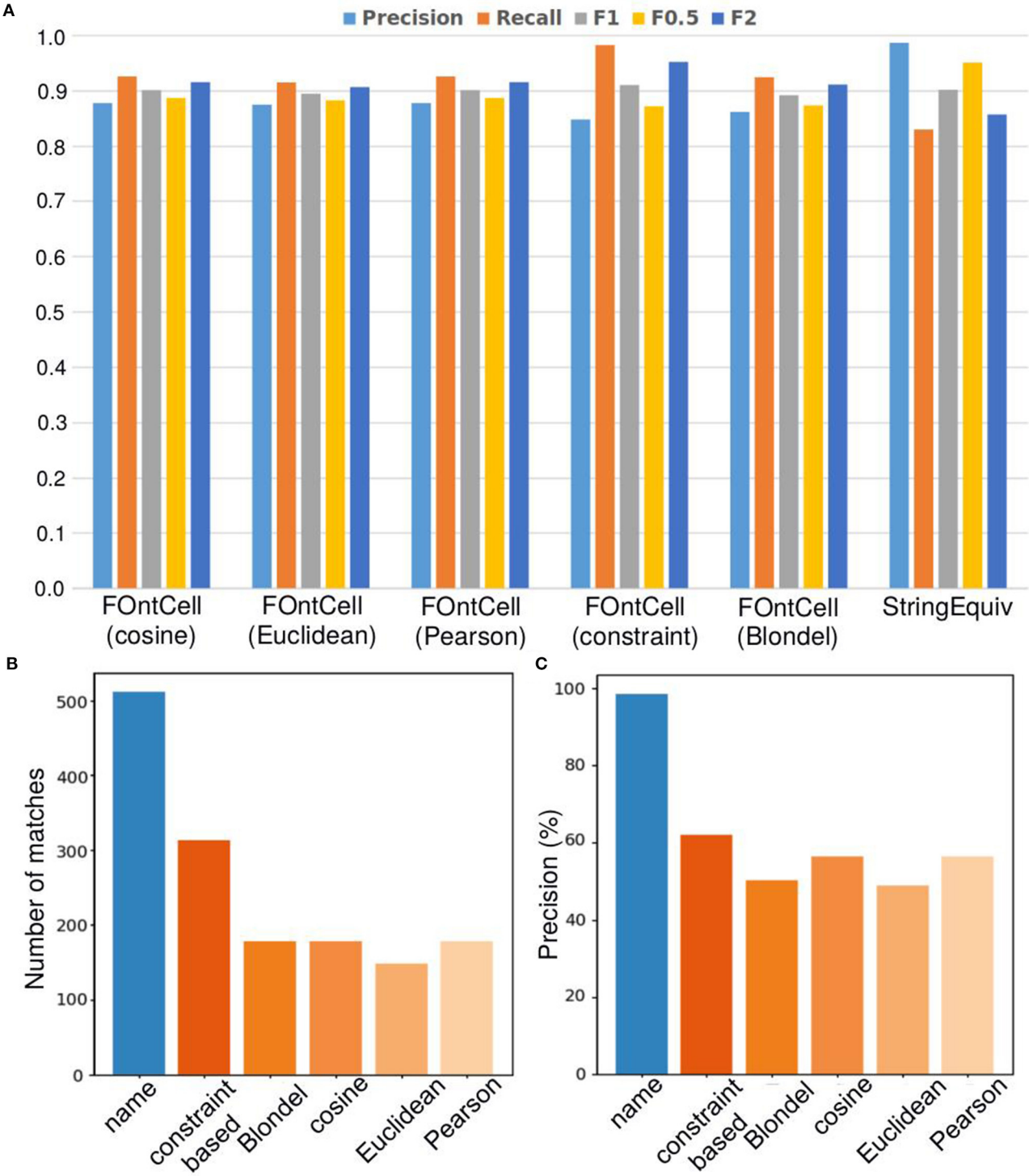

FIGURE 7 | Alignment performance of the different mapping methods of FOntCell when merging CELDA and LifeMap. **(A)** Precision, recall and F_β alignment accuracies of the different structure mapping methods combined with name mapping (FOntCell), and of name mapping applied separately (StringEquiv). **(B)** Number of matches during ontology matching with the different mapping methods. Name mapping is shown in blue and the structure mappings in different hues of orange. **(C)** Precision of the name matching and the different structure mapping methods. Name mapping is shown in blue and the structure mappings in different hues of orange. Alignment parameters $W = 4$, $\theta_{LN} = 0.7$ and $\theta_N = 0.85$.

TABLE 3 | Performance scores of FOntCell and other tools in the alignment of the anatomy ontologies of OAEI 2019.

	Precision	Recall	F_1	$F_{0.5}$	F_2
FOntCell (cosine)	0.861	0.720	0.784	0.829	0.744
FOntCell (Euclidean)	0.909	0.726	0.807	0.865	0.756
FOntCell (Pearson)	0.859	0.724	0.786	0.828	0.748
FOntCell (constraint)	0.846	0.718	0.777	0.817	0.740
FOntCell (Blondel)	0.860	0.724	0.786	0.829	0.748
StringEquiv	0.997	0.622	0.766	0.890	0.673
AML	0.950	0.936	0.943	0.947	0.939
LogMap	0.918	0.846	0.881	0.903	0.859
AGM	0.152	0.195	0.171	0.159	0.185
ALIN	0.974	0.698	0.813	0.903	0.740
DOME	0.996	0.615	0.760	0.886	0.666
FCAMap-KG	0.996	0.631	0.773	0.893	0.681
Lily	0.873	0.796	0.833	0.856	0.810
LogMapBio	0.872	0.925	0.898	0.882	0.914
LogMapLite	0.962	0.728	0.829	0.904	0.765
POMAP++	0.919	0.877	0.898	0.910	0.885
SANOM	0.888	0.844	0.865	0.879	0.852
GeoMean	**0.807**	**0.683**	**0.735**	**0.775**	**0.702**

In the case of FOntCell, the name of the used structure mapping method is inside parentheses. GeoMean are the geometric mean values of the performance scores of the other tools with which FOntCell is compared.

TABLE 4 | Performance scores of FOntCell and other tools in the alignment of CELDA and LifeMap.

	Precision	Recall	F_1	$F_{0.5}$	F_2
FOntCell (cosine)	0.877	0.925	0.900	0.886	0.915
FOntCell (Euclidean)	0.874	0.915	0.894	0.882	0.906
FOntCell (Pearson)	0.877	0.925	0.900	0.886	0.915
FOntCell (constraint)	0.847	0.982	0.910	0.871	0.952
FOntCell (Blondel)	0.861	0.924	0.891	0.873	0.911
StringEquiv	0.986	0.829	0.901	0.950	0.857
AML	0.971	0.269	0.422	0.639	0.315
LogMap	0.983	0.317	0.480	0.692	0.367
GeoMean	**0.980**	**0.413**	**0.567**	**0.749**	**0.463**

In the case of FOntCell, the name of the used structure mapping method is inside parentheses. GeoMean are the geometric mean values of the performance scores of the other tools with which FOntCell is compared.

convolution: vectorial topological similarity and constraint-based topological similarity. To calculate the vectorial topological similarities we designed a general method to calculate the similarities between vectors of different lengths for different metrics. Additionally, we adapted the Blondel method to work for such new topological convolution approach.

Different ontologies could benefit from different alignment parameters; e.g., for the CELDA + LifeMap merging, we found the vectorial methods produce similar results, with a slight advantage for the cosine method. All the functionalities of FOntCell allow the unification of dispersed knowledge in one domain into a unique ontology. FOntCell produces the results in commonly used ontology format files that can be re-used by FOntCell in an iterative way to adapt continuously the ontologies with the new data, endlessly produced by data-driven classification methods. To navigate across the merged ontologies, it generates HTML files with interactive circular DAGs.

FOntCell is a targeted tool for merging cell development ontologies. The objective behind this tool is the production of a cell-type ontology, which bases its relationships on development and serves as the basis for other works that require a holistic vision of cell development. FOntCell helps us collect the information within the different cell-type ontologies and contrasts them against each other without requiring standards or supervision, and grants us a final ontology that contains the cell types that are common and those that are not common between the two. FOntCell, being devised with this objective, does not obtain the same results when it tries to merge other types of ontologies which correspond to other internal hierarchies.

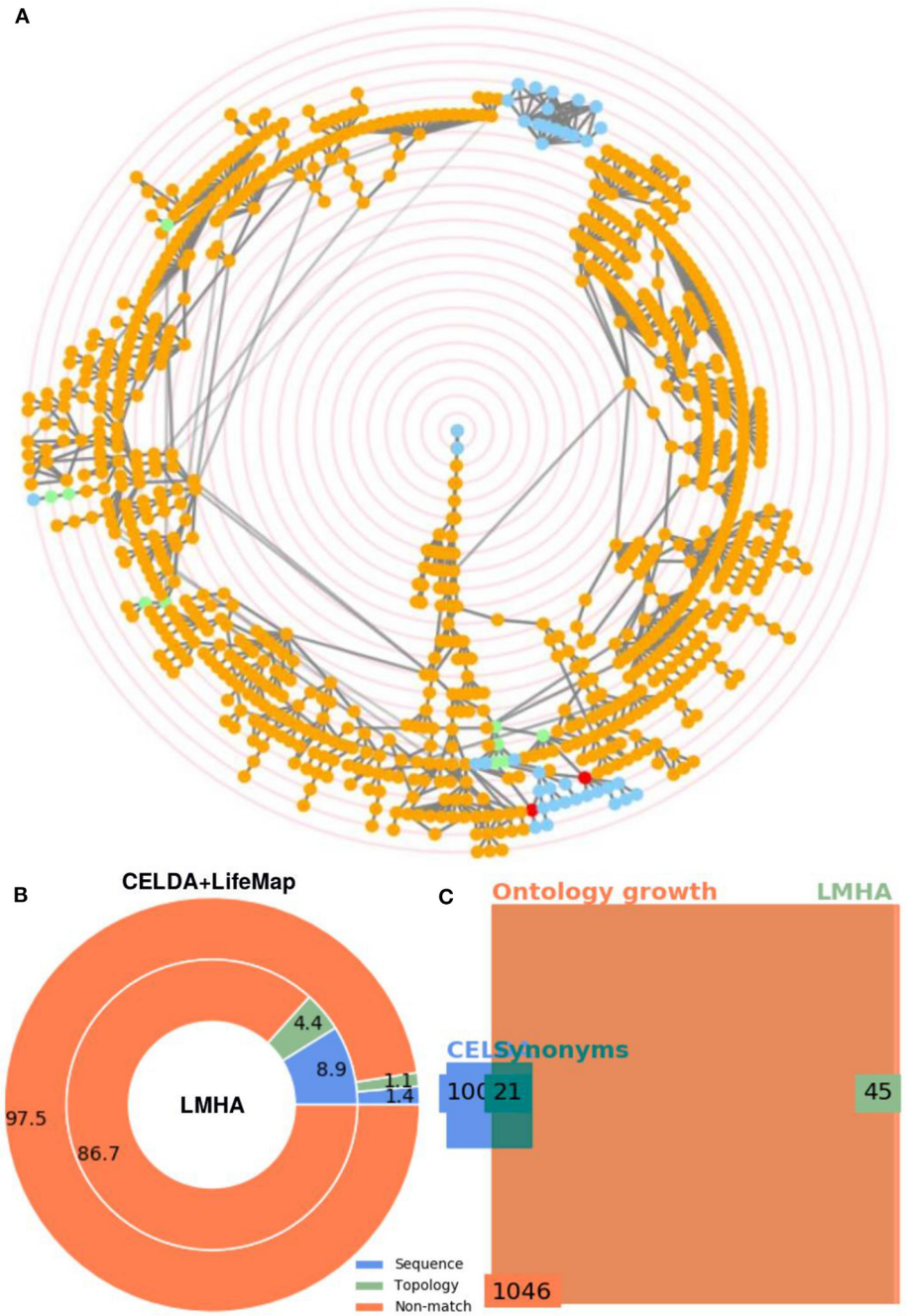

FIGURE 8 | Merging of CELDA + LifeMap with LungMAP Human Anatomy (LMHA) ontology. **(A)** Circular Directed Acyclic Graph (DAG) of the merged ontology. The orange and blue nodes are the non-matched contributions from CELDA+LifeMap and LMHA, respectively. The green and red nodes are the nodes with name and structure mapping, respectively. In the interactive application generated automatically in html by FOntCell, the ontology labels associated to the nodes appear when hovering over the nodes. The concentric red rings are zoom guides. **(B)** Donut plot of the percentages of classes added by name mapping vs. the classes added by structure mapping to the merged CELDA + LifeMap (outer circle) from LMHA (inner circle). **(C)** Square Euler-Venn diagram with the number of classes before and after the merge. The blue and light green rectangles frame the number of classes in CELDA + LifeMap and LMHA before the merging, respectively, the dark green rectangle frames the sum of name and structure equivalent classes, and the orange rectangle frames the total number of classes in the resultant CELDA + LifeMap + LMHA merged ontology. Alignment parameters $W = 4$, $\theta_{LN} = 0.7$ and $\theta_N = 0.85$.

Then, the performance scores obtained when merging two ontologies from domains other than cell types are above the average with respect to the rest of the OAEI tools. However, for these cases, there are less specific algorithms that are capable of aligning the ontologies, some of them better than FOntCell.

IMPLEMENTATION AND SOFTWARE AVAILABILITY

FOntCell is developed in Python v3.7 and uses the Python library NetworkX to derive the digraph relation of the ontology and to transform each class to a node and each hierarchy step to an edge. NetworkX graphs allows FOntCell access the sorted list of nodes without repeats, and produce digraphs compatible with graph visualization tools such as graphviz and matplotlib. For specific data manipulation, FOntCell uses numpy, pyexcell_ods, argparse, stringdist, and basic Python libraries such as os, collections and itertools. As other merging algorithms (Faria et al., 2018) the algorithm complexity (Big O) is quadratic time $O(n^2)$, however it is possible to reduce the time complexity in the matching problem from quadratic to linear implementing a hash-based searching strategy. For parallelization and the structure-mapping, FOntCell uses BigMPI4py (Ascension and Araúzo-Bravo, 2020). We added a demo function to the FOntCell distribution package merging CELDA with LifeMap.

The automatic installation installs all the dependencies. Additional installation information is provided at https://www.arauzolab.org/tools.html and at https://pypi.org/project/fontcell/. Full instructions of the prerequisites for installation, the downloading of FOntCell, the user manual, an example of how to run FOntCell and an example of the html output created by FOntCell are provided in the **Supplementary Material**.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

JC-L and MA-B: conceptualization. JC-L: data curation, software, and validation. JC-L, DG, and MA-B: formal analysis, investigation, methodology, writing-original draft preparation, and writing-review and editing. DG and MA-B: funding acquisition and project administration, resources, and supervision. JC-L, AA, MA-E, and MA-B: visualization.

FUNDING

JC-L has been supported by Ministry of Economy and Competitiveness, Spain, MINECO Predoctoral Grant No. BES-2017-080625. AA has been supported by Basque Government, Spain, Predoctoral Grant No. PRE_2018_1_0008. MA-E has been supported by Instituto de Salud Carlos III, Spain, i-PFIS Predoctoral Grant No. IFI18/00044, DG and MA-B have been supported by Grant No. DFG109/20 from Diputación Foral de Guipúzcoa, Spain, Ministry of Economy and Competitiveness, Spain, MINECO Grant No. BFU2016-77987-P and Instituto de Salud Carlos III (AC17/00012) co-funded by the European Regional Development Fund (ERDF/ESF, Investing in your future), by the European Union 4D-HEALING project (ERA-Net program ERACoSysMed/H2020 JTC-2 2017, Grant Agreement No. 643271) and by European Union FET project Circular Vision (H2020-FETOPEN, Project 899417).

ACKNOWLEDGMENTS

The authors would like to thank to two reviewers for their insightful feedback and Olga Ibáñez-Solé for fruitful discussion during the preparation of this manuscript.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcell.2021.562908/full#supplementary-material

REFERENCES

Ascension, A. M., and Araúzo-Bravo, M. J. (2020). "BigMPI4py: python module for parallelization of big data objects discloses germ layer specific DNA demethylation motifs," in *IEEE/ACM Transactions on Computing Biology and Bioinformatics* (New York, NY: IEEE).

Bard, J., Rhee, S. Y., and Ashburner, M. (2005). An ontology for cell types. *Genome Biol.* 6:R21. doi: 10.1186/gb-2005-6-2-r21

Blondel, V. D., Gajardo, A., Heymans, M., Senellart, P., and Van Dooren, P. (2004). A measure of similarity between graph vertices: applications to synonym extraction and web searching. *SIAM Rev.* 46, 647–666. doi: 10.1137/S00361445024 15960

Boldog, E., Bakken, T. E., Hodge, R. D., Novotny, M., Aevermann, B. D., Baka, J., et al. (2018). Transcriptomic and morphophysiological evidence for a specialized human cortical GABAergic cell type. *Nat. Neurosci.* 21, 1185–1195. doi: 10.1038/s41593-018-0205-2

Busse, J., Humm, B., Lübbert, C., Moelter, F., Reibold, A., Rewald, M., et al. (2015). Actually, what does "ontology" mean?: A term coined by philosophy in the light of different scientific disciplines. *J. Comp. Inform. Technol.* 23, 29–41. doi: 10.2498/cit.1002508

da Silva, J., Delgado, C., Revoredo, K., and Araújo Baião, F. (2019). ALIN Results for OAEI 2019. *CEUR Workshop Proc.* 2536, 164–168. Available online at: http://ceur-ws.org/Vol-2536/oaei19_paper2.pdf

Doan, A., Madhavan, J., Domingos, P., and Halevy, A. (2004). Ontology matching: a machine learning approach. *Handb. Ontol.* 385–403. doi: 10.1007/978-3-540-24750-0_19

Edgar, R., Mazor, Y., Rinon, A., Blumenthal, J., Golan, Y., Buzhor, E., et al. (2013). LifeMap Discovery^{TM}: the embryonic development, stem cells, and regenerative medicine research portal. *PLoS ONE* 8:e66629. doi: 10.1371/journal.pone.0066629

Ehrig, M., and Staab, S. (2004). QOM - quick ontology mapping. *Lect. Notes Comp. Sci.* 3298, 683–697. doi: 10.1007/978-3-540-30475-3_47

Faria, D., Pesquita, C., Mott, I., Martins, C., Couto, F. M., Cruz, I., et al. (2018). Tackling the challenges of matching biomedical ontologies. *J. Biomed. Sem.* 9:4. doi: 10.1186/s13326-017-0170-9

Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I. F., and Couto, F. M. (2013). The AgreementMakerLight ontology matching system. *Lect. Notes Comp. Sci.* 8185, 527–541. doi: 10.1007/978-3-642-41030-7_38

Gerovska, D., and Araúzo-Bravo, M. J. (2016). Does mouse embryo primordial germ cell activation start before implantation as suggested by single-cell transcriptomics dynamics? *Mol. Hum. Reprod.* 22, 208–225. doi: 10.1093/molehr/gav072

Gerovska, D., and Araúzo-Bravo, M. J. (2019). Computational analysis of single-cell transcriptomics data elucidates the stabilization of Oct4 expression in the E3.25 mouse preimplantation embryo. *Sci. Rep.* 9:8930. doi: 10.1038/s41598-019-45438-y

Giunchiglia, F., Shvaiko, P., and Yatskevich, M. (2004). S-match: an algorithm and an implementation of semantic matching. *Lect. Notes Comp Sci.* 3053, 61–75. doi: 10.1007/978-3-540-25956-5_5

Grindberg, R. V., Yee-Greenbaum, J. L., McConnell, M. J., Novotny, M., O'Shaughnessy, A. L., Lambert, G. M., et al. (2013). RNA-sequencing from single nuclei. *Proc. Natl. Acad. Sci. U. S. A.* 110, 19802–19807. doi: 10.1073/pnas.1319700110

Hertling, S., and Paulheim, H. (2019). DOME results for OAEI 2019. *CEUR Workshop Proc.* 2536, 123–130. Available online at: http://ceur-ws.org/Vol-2536/oaei19_paper6.pdf

Hwang, B., Lee, J. H., and Bang, D. (2018). Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp. Mol. Med.* 50. doi: 10.1038/s12276-018-0071-8

Jiménez-Ruiz, E. (2019). Logmap family participation in the OAEI 2019. *CEUR Workshop Proc.* 2536, 160–163.

Jiménez-Ruiz, E., and Cuenca Grau, B. (2011). LogMap: logic-based and scalable ontology matching. *Lect. Notes Comp. Sci.* 7031, 273–288. doi: 10.1007/978-3-642-25073-6_18

Kalfoglou, Y., and Schorlemmer, M. (2003). "IF-Map: an ontology-mapping method based on information-flow theory," in *Journal on Data Semantics I. Lecture Notes in Computer Science*, Vol. 2800, eds S. Spaccapietra, S. March, and K. Aberer (Berlin; Heidelberg: Springer). doi: 10.1007/978-3-540-39733-5_5

Kotis, K., and Vouros, G. A. (2004). The HCONE approach to ontology merging. *Lect. Notes Comp. Sci.* 3053, 137–151. doi: 10.1007/978-3-540-25956-5_10

Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., and Gargouri, F. (2017). "POMap: an effective pairwise ontology matching system," in *IC3K 2017 - Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management* (Funchal).

Lambrix, P., and Tan, H. (2006). SAMBO-A system for aligning and merging biomedical ontologies. *Web Seman.* 4, 196–206. doi: 10.1016/j.websem.2006.05.003

Lambrix, P., and Tan, H. (2008). "Ontology alignment and merging," in *Anatomy Ontologies for Bioinformatics* (London: Springer), 133–149. doi: 10.1007/978-1-84628-885-2_6

Lambrix, P., Tan, H., Jakoniene, V., and Stromback, L. (2007). Ch. 4 biological ontologies. *Seman. Web* 85–99. doi: 10.1007/978-0-387-48438-9_5

Le, B. T., Dieng-Kuntz, R., and Gandon, F. (2004). "On ontology matching problems - For building a corporate semantic web in a multi-communities organization," in *ICEIS 2004 - Proceedings of the Sixth International Conference on Enterprise Information Systems* (Porto), 236–243.

Levenshtein, V. I. (1966). Binary codes capalbe of correcting deletions, insertions, and reversas. *Sov. Phys. Doklady* 10, 707–710.

Lütke, A. (2019). AnyGraphMatcher submission to the OAEI knowledge graph challenge 2019? *CEUR Workshop Proc.* 2536, 86–93. Available online at: http://ceur-ws.org/Vol-2536/oaei19_paper1.pdf

Malone, J., Holloway, E., Adamusiak, T., Kapushesky, M., Zheng, J., Kolesnikov, N., et al. (2010). Modeling sample variables with an Experimental Factor Ontology. *Bioinformatics* 26, 1112–1118, doi: 10.1093/bioinformatics/btq099

McGuinness, D. L., Fikes, R., Rice, J., and Wilder, S. (2000). "An environment for merging and testing large ontologies," in *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning KR2000*, 483–493. Available online at: https://kr2020.inf.unibz.it/

Mitra, P., and Wiederhold, G. (2002). "Resolving terminological heterogeneity in ontologies declaratively," in *Proceedings of Workshop on Ontologies and Semantic Interoperability at the 15th European Conference on Artificial Intelligence (ECAI)* (Lyon), 45–50.

Noy, N. F., and Musen, M. A. (2000). "PROMPT: algorithm and tool for automated ontology merging and alignment," in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence* (Austin, TX), 450–455.

Osumi-Sutherland, D. (2017). Cell ontology in an age of data-driven cell classification. *BMC Bioinform.* 18(Suppl. 17):558. doi: 10.1186/s12859-017-1980-6

Prasad, S., Peng, Y., and Finin, T. (2002). "Using explicit information to map between two ontologies," in *Proceedings of the AAMAS Workshop on Ontologies in Agent Systems* (Bologna), 15–19.

Raunich, S., and Rahm, E. (2011). "ATOM: automatic target-driven ontology merging," in *Proceedings - International Conference on Data Engineering* (Hannover), 1276–1279.

Rozenblatt-Rosen, O., Stubbington, M. J. T., Regev, A., and Teichmann, S. A. (2017). The Human Cell Atlas: from vision to reality. *Nature* 550, 451–453. doi: 10.1038/550451a

Sarntivijai, S., Lin, Y., Xiang, Z., Meehan, T. F., Diehl, A. D., Vempati, U. D., et al. (2014). CLO: The cell line ontology. *J. Biomed. Seman.* 5:37. doi: 10.1186/2041-1480-5-37

Sas, A. R., Carbajal, K. S., Jerome, A. D., Menon, R., Yoon, C., Kalinski, A. L., et al. (2020). A new neutrophil subset promotes CNS neuron survival and axon regeneration. *Nat. Immunol.* 21, 1496–1505. doi: 10.1038/s41590-020-00813-0

Seltmann, S., Stachelscheid, H., Damaschun, A., Jansen, L., Lekschas, F., Fontaine, J. F., et al. (2013). CELDA - an ontology for the comprehensive representation of cells in complex systems. *BMC Bioinform.* 14:228. doi: 10.1186/1471-2105-14-228

Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., et al. (2007). The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat. Biotechnol.* 25, 1251–1255. doi: 10.1038/nbt1346

Smith, M. K., Welty, C., and McGuinness, D. L. (2004). *OWL Web Ontology Language Guide, W3C Recommendation*. Available online at: http://www.w3.org/TR/2004/REC-owl-guide-20040210/ (accessed November 12, 2009).

Stumme, G., and Maedche, A. (2001). "FCA-MERGE: Bottom-up merging of ontologies," in *IJCAI International Joint Conference on Artificial Intelligence* (Seattle, WA), 225–230.

Su, X., Hakkarainen, S., and Brasethvik, T. (2004). "Semantic enrichment for improving systems interoperability," in *Proceedings of the ACM Symposium on Applied Computing* (Nicosia), 1634–1641.

Wang, P., and Xu, B. (2008). Lily: Ontology alignment results for OAEI 2008. *CEUR Workshop Proc.* 431, 167–175. Available online at: http://ceur-ws.org/Vol-551/oaei09_paper8.pdf

Zhao, M., Zhang, S., Li, W., and Chen, G. (2018). Matching biomedical ontologies based on formal concept analysis. *J. Biomed. Seman.* 9, 1–27. doi: 10.1186/s13326-018-0178-9

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The handling editor declared a past collaboration with one of the authors MA-B.