



HAL
open science

Interconnected Kingdoms: Comparing 'A Song of Ice and Fire' Adaptations Across Media Using Complex Networks

Arthur Amalvy, Madeleine Janickyj, Shane Mannion, Pádraig Maccarron,
Vincent Labatut

► **To cite this version:**

Arthur Amalvy, Madeleine Janickyj, Shane Mannion, Pádraig Maccarron, Vincent Labatut. Interconnected Kingdoms: Comparing 'A Song of Ice and Fire' Adaptations Across Media Using Complex Networks. *Social Network Analysis and Mining*, 2024, 14, pp.199. 10.1007/s13278-024-01365-z . hal-04722579

HAL Id: hal-04722579

<https://hal.science/hal-04722579v1>

Submitted on 5 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License

Interconnected Kingdoms: Comparing ‘A Song of Ice and Fire’ Adaptations Across Media Using Complex Networks

Arthur Amalvy^{1*}, Madeleine Janickyj^{3,4*}, Shane Mannion^{2*},
Pádraig MacCarron^{2*} and Vincent Labatut^{1*}

^{1*}Laboratoire Informatique d’Avignon – UPR 4128, Avignon, France.

²MACSI, Department of Mathematics and Statistics, University of Limerick, Limerick, Ireland.

³Centre for Fluid and Complex Systems, Coventry University, Coventry, United Kingdom.

⁴ \mathbb{L}^4 Collaboration & Doctoral College for the Statistical Physics of Complex Systems, Leipzig-Lorraine-Lviv-Coventry, Europe.

*Corresponding author(s). E-mail(s): arthur.amalvy@univ-avignon.fr;
janickym@uni.coventry.ac.uk; shane.mannion@ul.ie;
padraig.maccarron@ul.ie; vincent.labatut@univ-avignon.fr;

Abstract

In this article, we propose and apply a method to compare adaptations of the same story across different media. We tackle this task by modelling such adaptations through character networks. We compare them by leveraging two concepts at the core of storytelling: the characters involved, and the dynamics of the story. We propose several methods to match characters between media and compare their position in the networks; and perform narrative matching, i.e. match the sequences of narrative units that constitute the plots. We apply these methods to the novel series *A Song of Ice and Fire*, by G.R.R. Martin, and its comics and TV show adaptations. Our results show that interactions between characters are not sufficient to properly match individual characters between adaptations, but that using some additional information such as character affiliation or gender significantly improves the performance. On the contrary, character interactions convey enough information to perform narrative matching, and allow us to detect the divergence between the original novels and its TV show adaptation.

Keywords: Character networks, Adaptations across media, Character matching, Narrative matching, Centrality study.

1 Introduction

It has been a long-standing tradition that if a story is highly successful in one medium, it gets adapted to another. For example, the 13th century folk-tales of *Robin Hood* was turned into stage plays two centuries later, and more recently into film and TV shows, or even the 10th century *Beowulf* being turned into a 21st century film. Hence, it is no surprise that recent successful stories are adapted into novels, movies, TV shows, computer games, comics, etc. In this study, we are interested in how that adaptation process affects the way the plot and the characters' roles change to conform to the constraints of a different medium. According to Hutcheon's terminology [1], we compare the *adaptation* of a *story* into different *forms*. We consider the *story* itself to be the underlying object (encompassing plot, characters, themes...) that is being transcribed into a *form* (film, book, video game...). For simplicity, we refer to the different versions of the same story as *adaptations*, even when that includes the original version of the story. Meanwhile, the *medium* refers to the physical support (electronic, book...) of each adaptation, while the *plot* is the arrangement of the series of events that is being shown. Following the *Living Handbook of Narratology*¹, we define a character as a "media-based figure in a storyworld, usually human or human-like".

In order to model the adaptations, we leverage character networks, a tool which has frequently been used for this purpose in recent years [2]. Many authors have tried to use such complex networks to compare different adaptations. Chaturvedi et al. [3] want to detect remakes among a collection of movies. They work with Wikipedia articles describing the movie plots, and extract a number of text- and graph-related features to train a classifier to distinguish between remakes from original films. Chowdhury et al. [4] tackle the problem of comparing novels and their movie adaptations, through their conversational networks. Their method focuses on four centrality metrics to summarise and compare the networks. Massey [5] works on the four *Gospels* and the *Acts of the Apostles*. He proposes a method to align their corresponding character networks, in order to compute some similarity scores between these adaptations. These scores are then used to perform a hierarchical clustering which highlights the relationships between the different adaptations. Finally, Zhang et al. [6] compare two books telling the same ancient Chinese story from the perspective of a historian and from that of a novelist. They work on the English translations to extract both character networks, and compare them in terms of standard topological properties (small-worldness, scale-freeness).

Although the works we are dealing with here are fictional, by having a self-contained dataset we can identify difficulties when working with adaptations, or different tellings of similar events. This is particularly problematic when dealing with historical social networks. A *hagiography* deals with the life of a saint, comparing these

¹<https://www-archiv.fdm.uni-hamburg.de/lhn/>

to the same events in history texts, different structural properties are observed [7]. This work identified some female characters, central to the events, that were barely mentioned in one source but more integrated in the network in another. Similar issues arise in dealing with medieval sagas, for example some Irish narratives appear in different manuscripts and have large differences [8]. These alternative versions have different numbers of characters [9], so it is important to be able to understand how the overall network differs and whether we can determine if this is due to the adaptation process.

Here, we analyse the book series *A Song of Ice and Fire*, the comic of the same name adapted from this, and its better known TV series *Game of Thrones*. When the author, George R. R. Martin – also a TV show writer – envisaged the book series as he began writing it, he deemed it too massive for a TV series to capture. However, less than 20 years after its publication, it spawned one of the most successful TV series of all time. As a sprawling epic across a large geographical region, the book series introduces many characters, almost 2,000 named characters by the end of the fifth book (the series is not yet completed). However, the TV adaptation, which is completed, contains over 500 named characters in total.

There are previous works analysing *A Song of Ice and Fire* through the prism of character networks. Beveridge and Shan [10] use traditional social network analysis tools to study a network they extract from the third novel, *A Storm of Swords*. Liu and Albergante [11] use structural balance to study the tensions between the main houses in the TV show. Beveridge and Chemers [12] perform analysis through character networks they obtain from the script of the TV show, notably around the concept of *fractal protagonist*. Garza et al. [13] study the first three seasons of the TV show, and focus on identifying the most important relationships. Stavanja et al. [14] try to predict the next kill in the TV Show, while Gessey-Jones et al. [15] try to assess the realism of the novels’ network. However, to the best of our knowledge, it is the first time that character networks are used to compare three adaptations of the same story, developed for three distinct media. More precisely, we focus on two research questions. First, is it possible to match the characters from one adaptation to the other, based on their interactions (RQ1)? Second, is it possible to use these interactions to align their plots (RQ2)? For instance, to determine whether one TV episode relates to a certain chapter of the novel. Answering these two research questions will allow understanding what tools are applicable when studying adaptations through character networks.

Concretely, our first contribution is to constitute a corpus of networks representing the original novels as well as their comics and TV show adaptations. In order to answer the RQ1, we formulate the task as a *Graph Matching* problem. Our contributions are empirical here, as we apply existing graph matching methods to experiment with various types of information when identifying the best match. We also consider a relaxed version of the problem, and perform a centrality analysis to provide some insight into our matching results. To tackle RQ2, we formulate the task as a many-to-many matching problem, which is called *Narrative Matching* in the literature [16, 17]. Our contributions here are more methodological, as we propose three approaches aiming at solving this task: the first is a baseline that relies on textual representations of the stories, the second leverages the structure of the dynamic networks modelling the stories, and the third combines these two types of information. We also propose

the notion of block, an intermediary temporal scale to represent the TV show, which leads to improved matching performance.

The rest of this article is organised as follows. We first present the raw data and the methods used to extract the character networks (Section 2). Next, we apply and compare several methods to match characters between networks, and answer RQ1 (Section 3). We then turn to RQ2, describing and assessing several approaches to align the dynamics of different adaptations (Section 4). Finally, we discuss our main findings and perspectives (Section 5).

2 Data Description

With three different forms of media being explored within this work, it is necessary to use three different methods to gather the raw data and extract the networks. Following the common practice regarding character networks [2], each vertex in our networks represents a character, whereas each edge models an interaction between them. In the following, we first describe how we extract the networks from the adaptations (Section 2.1), and how we represent the dynamics of the adaptations (Section 2.2). We then explain how we supplement the basic networks with vertex and edge attributes (Section 2.3). Next, we compute and discuss the main topological properties of the networks (Section 2.4). Finally, we define distinct groups of characters to conduct our experiments (Section 2.5). Our data² and source code³ are available online.

2.1 Character Interactions

The network extraction process is specific to each type of medium considered in this work. In order to produce comparable graphs, we select data sets and apply extraction methods that allow us to obtain edges with roughly the same semantics: two connected vertices model two characters that interact during the considered narrative unit.

The first medium corresponds to the five novels constituting George R. R. Martin’s original material, as of February 2024. We do not work directly with the raw text, but rather with the manual annotations provided by Gessey-Jones *et al.* [15]. They applied a close-reading approach, and recorded each character appearing in a chapter, and every interaction between two characters in each chapter. Each edge in our network model such an interaction. Note that this does include memories, as frequently important plot information is revealed through a character thinking of past events. As a result, the characters appearing in a chapter are not necessarily present at that particular time point.

The Comics series⁴ is constituted of two volumes, which are directly adapted from the first two novels. These volumes contain 24 and 32 issues respectively. Each issue is, in turn, split into two to four chapters matching those of the novels. These chapters were not published in the exact same order as in the novels, though, probably for editorial reasons related to the number of pages in an issue. Although their title is the same as the TV Show, the comics are a direct and close adaptation of the

²<https://doi.org/10.5281/zenodo.13893061>

³<https://github.com/CompNet/Sachan>

⁴<https://georgerrmartin.com/cover-art-gallery/?id=4512>

novels [18]. Their publication started with the first season of the TV show, presumably for commercial reasons. We manually annotated the comics using the method from [19], which uses the scene as the narrative unit. Here, a scene is defined as a sequence of consecutive panels involving the exact same group of characters, without interruption. In our network, we connect all characters interacting in the same scene.

The TV show⁵, entitled *Game of Thrones*, comprises eight seasons that cover the five novels, but also go beyond the original plot. Moreover, it is a much looser adaptation of the novels [20] than the comics. The plot is relatively close to Martin’s books at first, but gradually diverges, especially after the fifth season. The later seasons were developed from information made known by Martin to the show runners, and under their own direction. Jeffrey Lancaster provides a very rich dataset concerning the TV show⁶. We leverage his annotations describing scene co-occurrences. As before, we assume that two characters occurring in the same scene interact, and connect them in our network.

2.2 Representation of Time

Based on the annotated interactions, we extract several types of networks. First, we consider *static* networks by integrating all the interactions over certain periods of time, in order to constitute large snapshots of the adaptations. This integration connects two characters if there is at least one interaction between them during the considered period. Moreover, as in many character network-related works from the literature [2], we compute edge weights corresponding to the number of such interactions over the period of interest. In our experiments, we consider three periods: the first two novels, comic volumes, and TV show seasons, noted **U2**; the first five novels and seasons, noted **U5**; and all 8 seasons of the TV show, noted **U8**. We select these periods so that the concerned adaptations cover the same part of the plot, in order to allow a fair comparison. Table **SM1** in the Supplementary Material provides a summary of the structure of the adaptations in terms of their constitutive narrative units, whereas Figure **SM1** shows how much their timelines overlap.

In addition to the static networks, we extract different forms of *dynamic* networks. All of them are based on the same notion of temporal integration as before, but this time using a *sliding window*. As a result, one gets a sequence of graphs called *time slices*, each one representing a position of the sliding window in the adaptation. The size of this window can be expressed in terms of different narrative units, depending on the media. For the novels, the raw data only allows us to use the *chapter*. For comics, from the smallest to the largest unit, we use the *scene*, the *chapter*, and the *issue*. For the TV show, in the same order, we have the *scene* and the *episode*. However, there is a scale problem when comparing the TV show and novels, because a scene is much shorter than a chapter in terms of plot, and an episode is much larger. For this reason, we introduce the concept of the *block* to segment the TV show. Blocks are lists of contiguous scenes from the same episode, that ideally correspond to a *point of view* sequence, i.e. a part of the story narrated from the perspective of a given character. We make the assumption that a new block in the TV show starts when

⁵<https://www.hbo.com/game-of-thrones>

⁶<https://github.com/jeffreylancaster/game-of-thrones>

there is a change in geographical location, as it is likely that this indicates that the focus switches to a different character group. Since the annotations for the TV show include the geographical location of each scene, we automatically extract blocks by considering that two contiguous scenes are from the same block if they occur in the same location.

We extract a dynamic network for all available time scales (scene, block, chapter, issue, episode) and period (U2, U5, U8). Moreover, we adopt two representations of time, resulting in two distinct types of dynamic networks. In *cumulative* networks, each time slice integrates all the interactions since the *beginning* of the period. On the contrary, in *instant* networks, each time slice integrates only the interactions occurring in the corresponding *time window*. The former tend to be more stable, whereas the latter better reflect sudden changes in the dynamics, which makes them suitable for different uses.

2.3 Vertex & Edge Attributes

As mentioned before, edge weights are computed when performing the temporal integration of character interactions, simply by counting the number of such interactions occurring during some time period. Due to the difference between adaptations and time scales, these values cannot be compared directly. For this reason, we max-normalise them, i.e. we divide each weight by the largest weight in its network, which produces values in the interval $[0;1]$.

The two data sources that we use to extract the novels and TV show networks also provide some information regarding individual characters. Based on this information, we extract two vertex attributes representing the biological sex and the main affiliation of the characters. We solve any disagreement between the sources thanks to two Wiki Website populated by fans of the novels (*A Wiki of Ice and Fire*⁷) and of the TV show (*Wiki of Westeros*⁸). We also leverage the online resources to complement our own annotations of the comics and define the same vertex attributes for this medium. The *Sex* attribute corresponds to the biological sex as inferred from the adaptations. It can take the values *Male* and *Female*, but also *Unknown* in some certain cases, and *Mixed* when dealing with vertices representing groups of characters. The *Affiliation* attribute identifies the main organisation to which the character belongs. The term organisation must be taken in a very broad sense, since they can be institutions such as the *Gold Cloaks* (city watch of *King's Landing*) or the *Faith of the Seven* (clergy), and noble houses such as *House Stark* or *House Lannister*, but also more informal groups such as the *Brave Companions* (group of sellswords). Table SM2 in the Supplementary Material shows the main characters with their *Sex* and *Affiliation* attributes.

Finally, a critical vertex attribute is the name of the characters. There is some variability in these names from one adaptation to the other, and even within the same adaptation. For instance, some women are referred to by both their maiden and married names (eg. *Talisa Maegyr* vs. *Talisa Stark*); some characters have a nickname (ex. *Fat Walda* is *Walda Frey*); there are many homonyms that must be distinguished (ex. five persons are named *Walder Frey*). This heterogeneity prevents a

⁷https://awoiaf.westeros.org/index.php/Main_Page

⁸https://gameofthrones.fandom.com/wiki/Wiki_of_Westeros

direct comparison of the networks based on vertex names. For this reason, we leverage the previously mentioned online Wikis to curate a list of names used as a reference, and normalise the network names so that they all take the exact same form. We also create a *Named* Boolean attribute, which indicates whether a character has a proper noun. Thus, it is false for characters that are mentioned in the novels but referred to with some descriptive expressions only (e.g. *Stable boy*), as well as characters that are shown in the comics or TV show, but never explicitly introduced. We use this attribute to filter out minor characters, as explained later.

As mentioned previously, the TV show is a much looser adaptation of the Novels. Several pages of the online Wikis describe in detail how both adaptations differ, and three of these differences are particularly relevant to the present work. First, the name of certain characters has been completely changed when adapting the novels to TV. For instance, Dothraki warrior *Jhogo* becomes *Kovarro*, while the Yunkai nobleman *Grazdan mo Eraz* is renamed *Razdal mo Eraz*. Unlike the name differences described above, this is not due to natural language variability, but rather to a voluntarily choice of the showrunners. In general, this is meant to avoid confusion between characters that have the similar names, or identical first names [21]. We consider them as distinct characters in our networks, as these differences are not just cosmetic, but may also involve behavioural aspects. The second relevant difference in the TV adaptation is the merging of certain minor characters into a single character [22]. For instance, red priestess *Kinvara* from the TV show corresponds to both red priests *Benerro* and *Moqorro* from the novels. In our networks, we consider all of these characters as different. Third, some characters were outright created for the TV Show. We counted 85 of them in our data. By comparison, the names used in the comics are exactly the same as in the novels: the adaptation only affects the number of characters explicitly shown, which is constrained by the medium.

2.4 Descriptive Analysis

We initially compare the evolution of the total number of characters in each medium. The left panel in Figure 1 displays this for the three time periods. In each medium, the number of characters grows almost linearly, with the only obvious plateau being in the final time-period for the show. The growth of the number of characters is much slower in the show compared to the other two media.

We create a static network of each time period from the interactions. Table 1 displays some of the standard network properties for each of these. While they have different numbers of vertices, complex networks properties tend to not be related to system size (see, for example, Watts and Strogatz [23]). Here we observe that mean degree $\langle k \rangle$, average shortest path length $\langle \ell \rangle$, clustering coefficient C , and assortativity r do not change much in each time period.

Comparing the different media, the clustering coefficient is higher in the TV show and comics compared to the novels. It is likely an artefact of how interactions are defined in the annotations. In the comics and TV show, an edge is made between all characters participating in the same scene. In the novels, if there is a large group of characters, edges are only made when it is explicit that two of those characters interact.

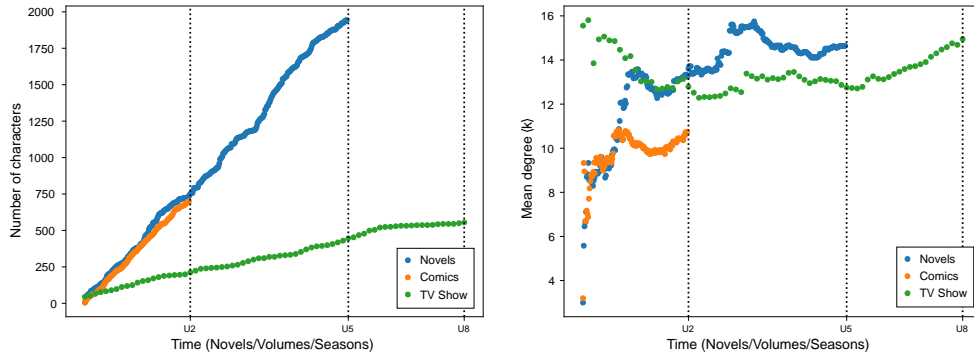


Fig. 1: The left panel shows the number of characters in plot time for each of the three media. On the right, the average degree $\langle k \rangle$ is displayed in time

Table 1: Network statistics for each of the time-periods giving the number of vertices n and edges L , the density δ , the mean degree $\langle k \rangle$, the average shortest path length $\langle \ell \rangle$, the average clustering coefficient $\langle C \rangle$, the degree assortativity r , and the modularity Q

Period	Adaptation	n	L	δ	$\langle k \rangle$	$\langle \ell \rangle$	$\langle C \rangle$	r	Q
U2	<i>Novels</i>	777	5,216	0.017	13.43	3.01	0.60	-0.01	0.53
	<i>Comics</i>	721	3,745	0.014	9.61	3.16	0.72	-0.17	0.68
	<i>TV Show</i>	199	1,308	0.066	13.15	2.88	0.75	0.07	0.57
U5	<i>Novels</i>	1,943	14,232	0.008	14.65	3.33	0.58	-0.02	0.58
	<i>TV Show</i>	430	2,767	0.030	12.87	2.90	0.78	-0.05	0.61
U8	<i>TV Show</i>	555	4,150	0.027	14.95	2.70	0.78	-0.08	0.44

The former methods will likely yield more closed triads. However, other properties are more similar for the TV show and novels.

The comics have a lower average degree and lower assortativity than the other two media. It is worth observing that a lot of the action is not shown in the comics, but described in long text boxes. This is quite unusual, as comics' authors tend to apply the *Show, don't tell* principle. This narrative text mainly concerns important characters, and the interactions described in the text do not appear in the comics annotations. The comic networks consequently miss some of these interactions between high degree vertices, which could explain the lower assortativity. The right panel of Figure 1 displays the mean degree chapter by chapter. Here we see the average degree stabilise mid-way through the first time period, though the novels have a big increase after the third book. This jump is not present in the TV show, but in the final time period we see this value increase as plots and characters converge.

On a global level, however, the network properties are not too different between each of the media. We might have expected the TV show and novel networks to be more different by the end of period U5 due to the stories diverging more, however this

is not the case. Despite fewer characters in the TV show, the topological properties are similar.

2.5 Character Sets

In the annotated data that we use to extract the networks, the numbers of characters are quite different from one adaptation to the other, as shown in Figure 1 and Table 2 (rows noted *All characters*). For period U2, there are 777 characters in the novels, but only 199 in the TV show. The comics contain 721 characters, a number comparable to the novels'. However, only 324 (45%) of them are named, and therefore likely to be of importance, whereas there are 731 (94%) named characters in the novels, and 167 (84%) in the TV show annotations. On the one hand, these variations reflect changes due to the adaptation process. For instance, it is not possible to show too many characters in a Comics panel while keeping them recognisable, and the TV Show needs to show physical persons and is therefore more expensive than the other two mediums. But on the other hand, these variations also show differences in the way the three adaptations are annotated. In particular, (with a handful of exceptions) unnamed characters are not listed in the novels annotations, and only a few of them are explicitly mentioned in the TV show annotations.

Table 2: Numbers of characters in the different character sets and periods, for each adaptation. The rows noted *All characters* correspond to the unfiltered character set. Some periods concern only certain adaptations, hence the empty cells. The **common** character set depends on the compared networks. There are 123 named characters common to all three U2 networks, and 20 in **top-20**

Period	Character set	<i>Novels</i>	<i>Comics</i>	<i>TV Show</i>
U2	All characters	777	721	199
	named	731	324	167
	common			
	Adaptation vs. <i>Novels</i>	–	297	139
	Adaptation vs. <i>Comics</i>	297	–	134
	Adaptation vs. <i>TV Show</i>	139	134	–
U5	All characters	1,943	–	430
	named	1,877	–	285
	common			
	Adaptation vs. <i>Novels</i>	–	–	216
	Adaptation vs. <i>TV Show</i>	216	–	–
U8	All characters	–	–	555
	named	–	–	348

In order to deal with this issue, and also to ease the analysis and comparison of the character networks, we consider three increasingly constrained types of character subsets. The first, which we note **named**, includes all the characters that are referred to by a proper noun. The second type includes all characters from **named** that appear in all three adaptations, or in both compared adaptations, depending on the context,

and it is noted **common**. The sizes of these sets are shown in Table 2 (rows **named** and **common**, respectively).

The third type of character set contains the 20 most important characters from **common**, and is noted **top-20**. In order to assess character importance, we consider the number of occurrences of each character in each adaptation, using the smallest available narrative unit: chapters for the novels, and scenes for the comics and TV show. We max-normalise these values separately for each adaptation, in order to get scores in $[0;1]$, where 1 corresponds to the most frequent character in the adaptation. We average the score of each character over the three adaptations to get a mean score representing their overall importance, and use it to rank the characters. Table SM2 from the Supplementary Material shows the **top-20** frequent characters for U2 and U5.

3 Character Comparison

In this section, we want to answer our first research question: *Is it possible to match characters between adaptations, based on the character network structure?* (RQ1). For this purpose, we focus on period U2, i.e. the first two books, volumes and seasons, in order to cover comparable stories. First, we formulate the task as a *Graph Matching* (GM) problem, and leverage GM methods to assess its feasibility (Section 3.1). In order to provide some insight to these results, we next conduct a centrality analysis aiming at studying the structural differences between the characters (Section 3.2).

3.1 Character Matching

Let us consider two graphs G_1 and G_2 that are similar, but not perfectly identical. They contain the same vertices, connected in roughly the same way. The GM problem consists in identifying a permutation of vertices allowing us to turn one graph into the other while minimising edge disagreement [24]. Put differently, one wants to match each vertex of one graph to one vertex of the other graph, in such a way that an edge (or absence of edge) between two vertices of one graph matches an edge (or absence of edge) between their counterparts in the other graph. The problem can be generalised straightforwardly to weighted edges, and to graphs with different vertex sets.

In order to solve this problem and answer RQ1, we first leverage existing GM methods (Section 3.1.1), before relaxing the problem in order to apply a simpler approach based on neighbourhood similarity (Section 3.1.2).

3.1.1 Graph Matching Methods

We apply five state-of-the-art methods implemented in the `iGraphMatch` R library⁹, covering three families of approaches. The principle underlying the first family is to relax the objective function, which can be done in different ways. Here, we select the methods based on *convex, indefinite* [25], and *concave* [26] relaxations. The *Percolation* method [27] belongs to the second family, which gathers iterative methods. Those start with a few seeds (vertices matched *a priori*) and propagate to the rest of the graph, adding the new best match at each iteration. Finally, *Umeyama's* algorithm [28]

⁹<https://cran.r-project.org/package=iGraphMatch>

belongs to the third family, which relies on the spectral properties of the adjacency matrices.

Two preprocessing steps may affect the resolution of the GM problem. First, it is possible that the compared graphs do not contain exactly the same vertices. It is then necessary to perform an operation called *padding* [24], which consists of adding to each graph, as *isolates*, the vertices that are only present in the other graph. This makes the GM problem harder, as any isolate of one graph could be incorrectly matched to any isolate of the other graph. The second preprocessing step is *centring* [29], which consists in transforming the adjacency matrices in order to give more importance to mismatches concerning the absence (vs. presence) of edges. In certain cases, this is known to improve the results [24].

We apply the five methods to all pairs of networks, with and without centring, and considering three versions of increasingly limited sets of characters, as defined in Section 2.5: 1) all named characters present in at least one of the two compared graphs (noted **named**); 2) only the characters present in both compared networks (**common**); and 3) only the 20 most important characters present in both networks (**top-20**). The **named** character set requires padding, whereas **common** and **top-20** do not. Table 3 shows the best results obtained over all considered methods and parameters, and expressed as proportions of correctly matched characters. For the sake of completeness, a comprehensive presentation of the results is provided in the Supplementary Material (Section SM2.1).

The top part of Table 3 (row 1–2) focuses on the matching performed with and without matrix centring. There is no clear improvement, and some additional experiments even show a decrease in performance when centring our networks (cf. Section SM2.1, Supplementary Material), so in the rest of this section we do not use centring anymore. Another interesting result is the performance increase when the considered character set gets narrower: the scores are always very low for all named characters, gets better when focusing on characters common to both considered networks, and even better when using only the 20 most important characters. In addition, the performance scores are similar when comparing the *Novels* vs. *Comics* or *TV Show* networks, but get better when comparing *Comics* vs. *TV Show*, which suggest the latter are more similar. Overall, the scores are very low: in the best case, we only match half of the top 20 characters correctly.

One way to improve the matching performance is to use the so-called *adaptive seeds* method [24]. A *seed* is a match assumed to be correct, which can be used as an input of a graph matching method, making its work easier. In the general case, a seed is a trusted match because it comes from some ground truth. The adaptive seeds method proceeds differently, though. It consists in applying a matching algorithm to get a first tentative vertex map, as well as a score that estimates the reliability of the corresponding matches. The most reliable matches are then used as seeds when applying the same method again. It is possible to consider the seed as *hard*, i.e. it is assumed to be a perfectly exact match, which should be absolutely respected, or as *soft*, i.e. the match is just viewed as very probable, and does not have to be respected. The second part of Table 3 (rows 3–4) shows the results obtained using adaptive hard (noted *AH*) and soft (*SH*) seeds. The performance of both approaches is poor

Table 3: Best results obtained for the graph matching problem. The left columns indicate the method parameters: considered vertex attributes (*Attr.*), centring of the adjacency matrices (*Centr.*), nature or number of the seeds (*Seeds*). The other columns show the proportions of characters correctly matched for each pair of networks, considering three character sets (cf. main text)

Method Parameters			<i>Novels vs. Comics</i>			<i>Novels vs. TV Show</i>			<i>Comics vs. TV Show</i>		
Attr.	Centr.	Seeds	named	common	top-20	named	common	top-20	named	common	top-20
None	Yes	0	0.13%	3.42%	20.00%	0.13%	10.07%	20.00%	0.57%	21.97%	50.00%
–	No	0	0.53%	3.77%	20.00%	0.26%	3.60%	25.00%	5.10%	21.21%	35.00%
None	No	AH	0.00%	25.34%	15.00%	0.00%	11.51%	5.00%	0.00%	34.85%	50.00%
–	–	AS	0.00%	4.79%	15.00%	0.00%	12.23%	25.00%	0.00%	30.30%	15.00%
None	No	5	4.26%	14.63%	60.00%	2.39%	20.90%	40.00%	12.36%	38.58%	86.67%
–	–	10	6.43%	23.40%	80.00%	2.54%	27.91%	60.00%	11.66%	42.62%	100.00%
–	–	15	7.01%	26.35%	100.00%	3.76%	33.87%	60.00%	10.95%	39.32%	100.00%
Sex	No	0	0.40%	3.77%	15.00%	0.66%	7.91%	15.00%	0.85%	16.67%	80.00%
Aff.	–	0	14.93%	65.75%	60.00%	6.32%	64.75%	40.00%	15.58%	81.06%	80.00%
Both	–	0	16.12%	73.97%	65.00%	6.46%	64.75%	55.00%	15.86%	83.33%	100.00%

on **named** characters; close to zero matches. On the contrary, adaptive seeds improve the results for all pairs of networks when considering **common** characters. There is not much effect when considering **top-20** characters, probably because of the size of this vertex set. These results confirm that the task of matching characters based only on the graph structure is a difficult, as the best case performance stays at 50%.

It is possible that incorporating extra information in addition to the graph structure could help to improve results. To test this assumption, we first leverage *ground truth* hard seeds, i.e. we provide the methods with a few exact matches as inputs. We experiment with 5, 10, and 15 seeds for **named** and **common** characters, which represent between 1% and 11% of the vertices, depending on the considered networks. The seeds are selected among the most important characters. In any case, the performance is assessed only on the remaining non-seed vertices. The third part of Table 3 (rows 5–7) shows the matching performance; it improves in all cases compared to our previous methods. It remains unsatisfying, though, with scores under 50% for **named** and **common** characters. The values are much higher for **top-20** characters (up to 100% correct matches), but the seeds proportionally represent a large part of the network.

Another way of leveraging some extra information to improve matching is to use vertex attributes. In our case, we know the sex and affiliation of each character (cf. Section SM1 in the Supplementary Material for some examples). We use them to compute similarity scores between the characters, and fetch them as soft inputs to the matching methods. The last part of Table 3 (rows 8–10) shows the results obtained when using sex and affiliation (noted *Aff.*) separately, and together (noted *Both*). Sex alone does not bring any noticeable improvement compared to no attribute at all, while the affiliation strongly helps to match the characters in all cases, especially when comparing the *Comics* and *TV Show* networks. Interestingly, combining both

attributes leads to even better results, on par or better than the scores obtained using ground truth seeds.

3.1.2 Similarity-Based Matching

Our results from the previous section reveal that it is difficult to match characters based on graph structure alone. To explore this question further, we consider a relaxed version of this problem: we assume that, when comparing two characters, their neighbours are known and can be used to perform this comparison.

Our proposed method directly leverages this additional information. For each pair of characters (v_1, v_2) belonging to the two compared networks G_1 and G_2 , we assess the inter-character similarity by computing the weighted version of Jaccard’s index, also called Ružička’s similarity [30, 31], of their respective neighbourhoods. Let us note \mathbf{x} and \mathbf{y} the rows (or columns) representing v_1 and v_2 in the adjacency matrices of their respective graphs. We assume that G_1 and G_2 contain the same characters, in the same order (which may require some padding and reordering). The similarity between v_1 and v_2 is defined as

$$R(\mathbf{x}, \mathbf{y}) = \frac{\sum_i \min(x_i, y_i)}{\sum_i \max(x_i, y_i)}. \quad (1)$$

Like the original Jaccard index, this measure ranges from 0 (completely different vectors) to 1 (identical). The most likely match of a character v_1 is the most similar character in G_2 . Of course, the opposite might not be true: there may be another character $v'_1 \neq v_1$ in G_1 that is more similar to v_2 than v_1 . For this reason, we consider that we have a correct match only if it works in both directions.

Table 4: Results obtained when matching vertices using their neighborhood. The left column indicates how time is represented in the considered graphs: static, or dynamic (cumulative vs. instant networks). As in Table 3, the other columns show the proportions of characters correctly matched for each pair of networks, considering three character sets

Time	<i>Novels vs. Comics</i>			<i>Novels vs. TV Show</i>			<i>Comics vs. TV Show</i>		
	named	common	top-20	named	common	top-20	named	common	top-20
Static	10.57%	42.81%	70.00%	4.87%	43.88%	70.00%	16.71%	54.55%	95.00%
Cumulative	13.64%	47.95%	70.00%	–	–	–	–	–	–
Instant	10.28%	64.56%	100.00%	–	–	–	–	–	–

The first row of Table 4 shows the matching results obtained with this method. We observe the same differences as before: 1) the narrower the character set, the higher the performance; and 2) the best match seems to be obtained when comparing the *Comics* and *TV Show* networks. The former point is clear when considering the similarity matrices shown in Figure 2, as the highest values are generally located on their diagonals. By comparison, the full matrices exhibit many off-diagonal high

similarity values (cf. Figure SM7 in the SM). Interestingly, even though the amount of information provided in addition to the structure could be considered as larger than when we leverage ground truth seeds or vertex attributes to perform graph matching in Section 3.1, the results are not clearly better here. Interestingly, the method works well to identify characters undergoing major transformations during the adaptation process, e.g. splitting or merging as discussed in Section 2.3: see the Supplementary Material for more detail (Section SM2.1.6).

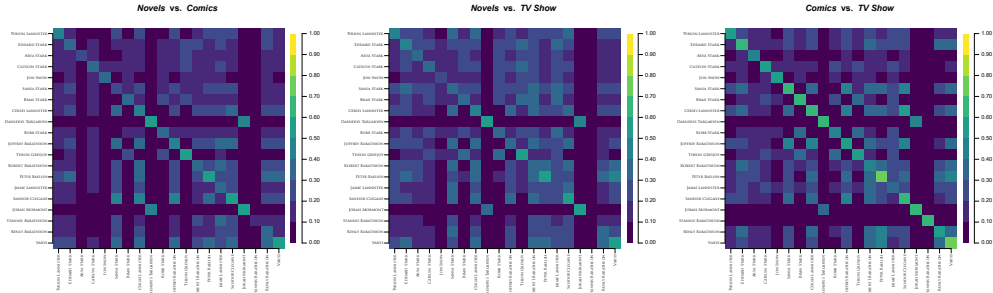


Fig. 2: Similarity matrices obtained with Ružička’s similarity, for the 20 most important characters of each pair of adaptations

We propose a sequential extension of this similarity-based method, in order to take advantage of the dynamic networks. At each time step, we match the characters using Ružička’s similarity, as before. For a given character, we thus obtain a series of (estimated) matches over the whole timeline. We select the statistical mode of these matches as the overall match, and randomly break possible ties. This approach requires two dynamic networks with the exact same numbers of time slices, so we can only apply it to the chapter-based *Novels* and *Comics* networks. The bottom part of Table 4 shows the obtained results. The performance obtained with the cumulative networks is similar to that of the static network, but the instant networks lead to much improvement, on par with what we got earlier when leveraging vertex attributes.

3.1.3 Takeaways

To summarise the main results obtained in this section, it appears that matching characters based only on the graph structure cannot be performed reliably with current state-of-the-art methods. The *Novels* network, in particular, generally leads to fewer matches when compared to the other adaptations. On the one hand, this is a bit surprising, as the comics are a more faithful adaptation of the novels than the TV show. On the other hand, both the comics and TV show are visual media and therefore share a number of adaptation constraints. In any case, the difficulty to match the characters suggests that their positions and roles vary from one adaptation to the other.

However, the performance gets better when focusing on characters common to both compared adaptations, and even more so with only the most important characters. This could reveal that the differences between the networks mainly concern minor

characters, but it could also be due to a size effect. Finally, it is worth stressing that leveraging additional information (seeds, vertex attributes, neighbourhood) in addition to structure greatly improves the matching performance.

3.2 Centrality Analysis

In order to better understand the differences between the networks, we perform a descriptive analysis of the characters. We compute a selection of standard metrics to assess their centrality: degree, betweenness, closeness, and Eigenvector centrality. As our networks are weighted, we consider both unweighted and weighted variants for each metric¹⁰. We use the normalised version of the weights, and standardise (z -score) the centrality scores, in order to get values that are comparable over all adaptations and metrics.

We first compare the behavior of the selected centrality metrics over the three adaptations (Section 3.2.1), before identifying and discussing the characters' centrality profiles (Section 3.2.2).

3.2.1 Centrality Correlation

Figure 3 shows Spearman's correlation between the selected centrality metrics. As in Section 3.1, we focus on period U2, i.e. the first two books, volumes and seasons. Moreover, the figure focuses on **named** and **top-20** character sets. However, results concerning the **common** characters, as well as period U5 (the first five books and seasons), are presented in the Supplementary Material (Section SM2.2.1).

When considering **named** characters (top row of the figure), all three adaptations exhibit similar matrices, with diagonal 2×2 blocks. These show that the unweighted and weighted versions of the same metric are very correlated (to a lesser extent, in the case of the betweenness), and therefore possibly redundant. In addition, in the case of the *TV Show*, the closeness and Eigenvector centrality metrics are also highly correlated (off-diagonal block).

Focusing only on the **top-20** characters (bottom row of the figure) provides a different picture. First, each **top-20** matrix differs from its **named** counterpart; the correlation between the unweighted and weighted versions of the same metric is a bit weaker, but more metrics are correlated (e.g. degree and closeness). Most characters are very minor, and are consequently attached to the rest of the network through low weight edges. Consequently, for these characters, the weighted and unweighted versions of the same metric yield very similar scores. This can explain the higher correlation they exhibit when considering **named** characters. Important characters tend to be considered central according to several metrics at once, which could explain the higher correlation between distinct metrics when focusing only on **top-20** characters. Our second observation is that matrices from the bottom row exhibit more variability than those of the top row. This hints at differences in the way important characters are interconnected within the three adaptations. As already observed in Section 3.1, the comics and TV show exhibit a higher similarity, compared to the novels.

¹⁰Note that the weighted version of the degree is called the *strength*.

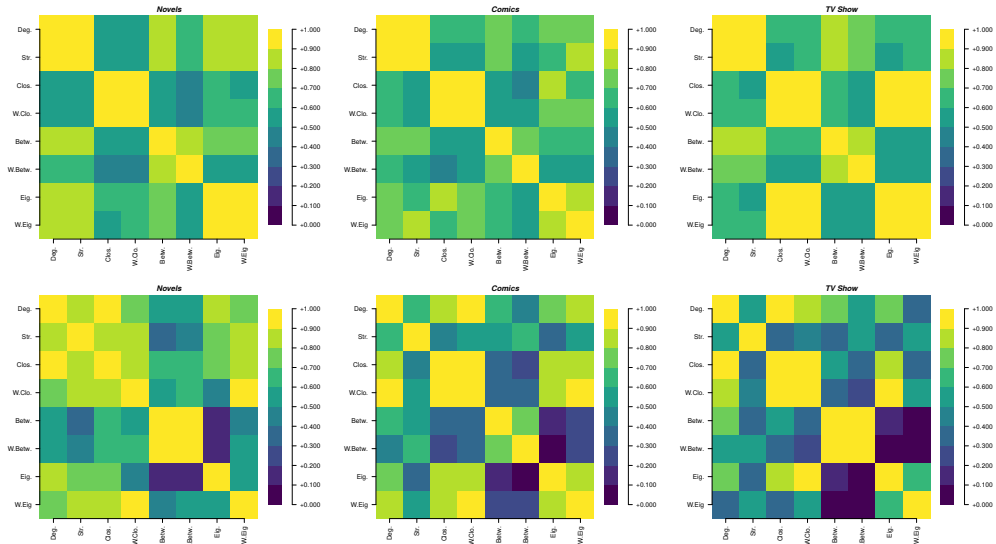


Fig. 3: Spearman’s correlation between the selected centrality metrics, for each of the three adaptations over period U2, considering the named (top row) and top-20 (bottom row) character set

3.2.2 Centrality Profiles

For each character, we now leverage the selected centrality metrics to constitute a so-called *centrality profile*, which corresponds to the vector of its centrality scores. Put differently, each character is represented as a point in an 8D centrality space. Figure 4 shows these centrality profiles as radar plots, for the **common** character set, considering period U2. The five most important characters are represented in color, whereas the others are in gray.

Let us compare the metrics first: it appears that the degree, Eigenvector centrality and Closeness exhibit similar behaviours, with centrality scores that cover the whole range, including low, intermediary and high values. By comparison, only a very few characters have a high betweenness, and it is very low for the others. Regarding the adaptations, one can observe that the separation between the most central characters and the others is much clearer in the novels, and that the comics tend to exhibit higher betweenness scores. In all three adaptations, the most important characters are also the most central. However, they exhibit very different centrality profiles. For instance, in the novels, *Eddard Stark* is the most central character according to all metrics, whereas he is just a central character among others in the comics and TV show.

Using a standard hierarchical method, we perform a cluster analysis in the centrality space. We select the most appropriate cuts in the resulting dendrograms based on the Silhouette measure [32]. This allows us to identify classes of characters holding similar positions in their network. Instead of directly comparing the networks as we do with matching methods in Section 3.1, here we want to do so through these similarity classes. Figure 5 shows the clusters obtained for the first two seasons and

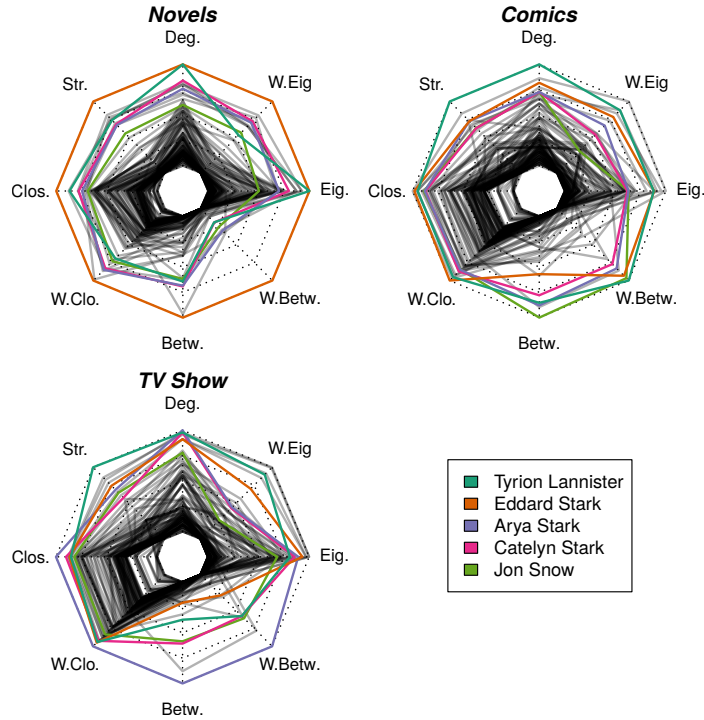


Fig. 4: Centrality profiles of the common character set, over period U2, for all three adaptations. The five most important characters are represented in color

books, as radar plots. Additional results are provided in the Supplementary Material (Section SM2.2.2). For novels and comics, we observe a separation between minor (clusters C_1) and major (C_2) characters. In the case of the TV show, major characters are split depending on whether they possess a low (C_2) or a high (C_3) betweenness. This difference allows us to distinguish between characters that have a local importance, like *Eddard Stark*, from those that connect independent storylines, such as *Arya Stark*.

The Adjusted Rand Index (ARI) [33] is a standard metric when it comes to comparing partitions. It reaches 1 for a perfect match, and 0 for orthogonal partitions. We use it to compare the adaptations as described by their clusters. Note that this comparison does not rely on some typical centrality scores of the clusters, but rather on which characters they contain. It turns out the pairs of adaptations exhibiting the most similar clusters are, by decreasing similarity: *Novels* vs. *Comics* (0.68), *Comics* vs. *TV Show* (0.31), and *Novels* vs. *TV Show* (0.21). This fits the intuition one could get by visually inspecting Figure 5.

3.2.3 Takeaways

In summary, this section confirms some of the observations made in Section 3.1 based on character matching. First, the three adaptations are only partially similar in terms

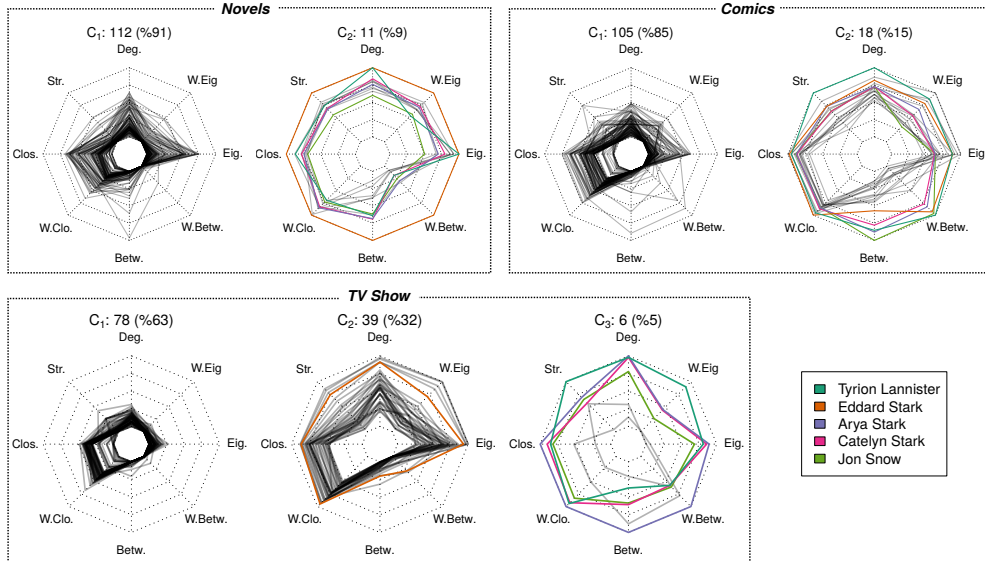


Fig. 5: Classes of common characters, detected based on their centrality profiles, over period U2, for all three adaptations. The five most important characters are represented in color

of their character network structure. Second, they appear to be more similar when focusing on **common** and **top-20** characters. This suggests that the many minor characters are handled differently in both adaptations (comics and TV show) of the novels, but that they tend to respect more the original structure when it comes to the main characters.

However, a closer look at the characters through their individual centrality profiles reveals that, if the main characters are indeed clearly distinguishable from the rest of the cast in all three adaptations, they also hold different positions depending on the adaptation. This could reflect the fact that the adaptation process does not necessarily put the same emphasis on the same characters as in the original material. Moreover, when looking at the characters in terms of centrality classes, it appears that the novels and comics form the most similar pair of adaptations, contrarily to what was observed in Section 3.1 for graph matching.

4 Narrative Alignment & Media Divergence

As explained in Section 2.1, the comics are a direct adaptation of the first two novels and closely follow them. In contrast to this, the TV show is a looser adaptation, which increasingly diverges from the novels, especially after season five. To determine how much the adaptation of a medium diverges from its original source, one can study whether each narrative unit from the original medium (here, the novel chapters) appears in the target medium (e.g. comics issues and TV episodes), and if the chronology of these units is preserved. In this section, our objective is to tackle this

task, which we call *Narrative Alignment*. This will allow us to answer our second research question: *Can we use character interactions to align the plots of adaptations from different media?* (RQ2).

We first provide the general framework for our problem and approach in Section 4.1, before proposing and testing two narrative matching methods: one based on text (Section 4.2), and the other on network structure (Section 4.3). Finally, we combine both approaches in Section 4.4, in order to assess their complementarity. Section 4.5 summarises our findings.

4.1 General Framework

We first formulate the narrative alignment problem and the metrics and ground truth that we use for performance assessment (Section 4.1.1). We then describe the general approach that we adopt to tackle this problem (Section 4.1.2), on which we elaborate later to build three matching methods.

4.1.1 Problem Formulation

The narrative alignment problem is defined as follows: given two media, let $\mathbf{a} = (a_1, \dots, a_n)$ and $\mathbf{b} = (b_1, \dots, b_m)$ be the vectors of their constitutive narrative units, with respective lengths n and m . For example, \mathbf{a} could represent episodes from the TV show while \mathbf{b} could represent chapters from the novels. We must find a matrix $\mathbf{M} \in \{0, 1\}^{n \times m}$ where $M_{ij} = 1$ if narrative unit a_i corresponds to narrative unit b_j , and $M_{ij} = 0$ otherwise. Note that \mathbf{M} describes a many-to-many relationship: a narrative unit from \mathbf{a} may correspond to several narrative units from \mathbf{b} , and vice versa. As an example, an episode from the TV show usually adapts several chapters from the novels, while parts of the same chapter may appear in distinct episodes. It is worth stressing that many-to-many matching is much harder than one-to-one matching, as the relation arity is unknown (i.e. one does not know how many narrative units are involved on either sides).

In order to evaluate our narrative matching methods, we gather gold alignments for all the pairs of media. After manual verification, we adapt the *Novels vs. TV Show* alignment from a fan annotation¹¹ who matched chapters and episodes. We create the *Novels vs. Comics* ourselves, at the levels of chapters. We use both of these to automatically produce the *Comics vs. TV Show* alignment. It is worth stressing that manually extracting such alignments necessitates significant effort, so proposing a method to automate this task with satisfying performance would be very useful.

When assessing a matching \mathbf{M} given a gold standard matrix $\mathbf{M}^* \in \{0, 1\}^{n \times m}$, we have four possible outcomes for each pair of narrative units:

- A *True Positive* (TP) occurs when $M_{ij} = 1$ and $M_{ij}^* = 1$.
- A *False Positive* (FP) occurs when $M_{ij} = 1$ and $M_{ij}^* = 0$.
- A *True Negative* (TN) occurs when $M_{ij} = 0$ and $M_{ij}^* = 0$.
- A *False Negative* (FN) occurs when $M_{ij} = 0$ and $M_{ij}^* = 1$.

¹¹<https://joeltronics.github.io/got-book-show/bookshow.html>

These matrices are sparse, and the true negatives are consequently the most frequent outcome, by far. For this reason, we use the *F1-score*, a standard measure from the field of Information Retrieval (IR) [34], to compute our matching performance. It is the harmonic mean of two other well-known IR measures, *Precision* and *Recall*:

$$Pre = TP / (TP + FP) \tag{2}$$

$$Rec = TP / (TP + FN) \tag{3}$$

$$F_1 = 2 \cdot Pre \cdot Rec / (Pre + Rec). \tag{4}$$

A higher F1-score indicates better performance.

4.1.2 Proposed Methods

Our general approach to perform automatic narrative alignment requires first to compute a similarity matrix $\mathbf{S} \in \mathbb{R}^{n \times m}$ between the two considered media. We elaborate further on this point in the next subsections, but in summary, we experiment with two different types of information to obtain this matrix. First, we leverage textual representations of the three adaptations to derive a *textual similarity* matrix. Second, we take advantage of character interactions to compute a *structural similarity* matrix. Finally, we also try to combine textual and structural similarity to compute a *hybrid similarity*, in order to understand if they are complementary.

After computing the similarity matrix \mathbf{S} between two media, we use an alignment algorithm to derive a matching from it. We experiment with two different approaches: basic thresholding vs. the Smith–Waterman algorithm [35]. The former is straightforward: to produce a matching \mathbf{M} , we set a threshold t and we compute M_{ij} as follows:

$$M_{ij} = \begin{cases} 1, & \text{if } S_{ij} > t \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

For each pair of media, we estimate the optimal threshold t by tuning it on the two other pairs. We consider thresholding as a simple alignment baseline.

The Smith–Waterman alignment algorithm [35] is a dynamic programming algorithm that was originally proposed to perform molecular sequence alignment. It identifies the best local alignment between two sequences by first scoring matches and gaps, and then backtracking through a dynamic programming matrix. Recently, Piau and Skiena [17] propose GNAT, a tool that adapts the Smith–Waterman algorithm by enabling many-to-many matching to perform narrative alignment. Piau et al. [16] apply this tool to align plots between novels and the scripts of their film adaptations. However, in both articles, the authors limit themselves to textual similarity when extracting their alignments. In order to also experiment with structural similarity, we re-implement their many-to-many adaptation of the Smith–Waterman algorithm. Similarly to the thresholding alignment method, we tune the parameters of the algorithm for each pair of media on the two other media pairs.

In summary, our experimental setup involves three different ways of computing the similarity matrix between adaptations, and two algorithms to leverage this matrix and estimate the match, which makes a total of 6 methods. We apply them to the

three possible pairs of media: *Novels vs Comics*, *Novels vs. TV Show* and *Comics vs. TV Show*. As stated in Section 2.2, each medium covers a different time period, so we work on the longest common period for each pair: U2 for the *Novels vs. Comics* and *Comics vs. TV Show* pairs, and U5 for the *Novels vs. TV Show* pair. Since the TV show diverges more and more from the novels as time progresses, we additionally study the alignment of the *Novels vs TV Show* pair over the U2 period in the Supplementary Material (see Section SM3.5).

4.2 Text-Based Method

Our first method leverages textual representations of the adaptations to compute the similarity between them. We originally experimented with long textual representations: entire chapters, dialogues from TV Show episodes, and texts extracted using OCR for the comics. However, we found a much better matching performance when using the following short texts instead.

- **Novels Chapters:** We scrape the existing chapter summaries from the previously mentioned fan Wiki [36]. With a mean length of 4.56 sentences, these summaries are usually much shorter than episode summaries, and longer than comics summaries.
- **Comics Issues:** Most (41/56) issues of the comics contain a summary of the next issue that acts as a teaser. We use OCR to extract these texts, and correct them manually. For the 15 remaining issues, we manually write summaries of approximately the same length. The mean length of these summaries is 2.86 sentences, the shortest amongst all media.
- **TV Show Episodes:** We scrape the episode summaries of the first five seasons available from Wikipedia [37–41]. The mean number of sentences in these summaries is 11: since episodes encompass different points of views and a number of subplots, each of these is represented by one or more sentences.

These textual sources constrain the narrative units used to represent the plots: the alignment is performed at the *chapter* level for the novels, at the *episode* level for the TV show, and at the *issue* level for the comics.

To compute the similarity matrix \mathbf{S} between the narrative units of the pairs of media, we try two different textual similarity functions:

- *tfidf*: compute the cosine similarity of the bag-of-words representation of two summaries weighted by standard TF-IDF [34]. This scheme weights each word according to its *term frequency* (TF) and *inverse document frequency* (IDF) [42], the latter being an indication of how characteristic of a document a term is.
- *sbert*: embed both summaries using SentenceBERT [43], and compute their cosine similarity. SentenceBERT is a variation of the language model BERT [44] specifically fine-tuned to extract semantic sentence embeddings, so that two sentences that are semantically similar should be close in the embedding space.

The results of our textual alignment can be found in Table 5. Matching performance varies highly between media pairs, with the *Novels vs. TV Show* pair being the hardest to match (best F1-score: 22.55) and the *Novels vs. Comics* being the easiest (best F1-score: 55.24). The Smith–Waterman alignment algorithm largely outperforms the

thresholding baseline in almost all configurations. The role of the similarity function is more puzzling: while *tfidf* performs better when using the thresholding baseline, *sbert* outperforms it when using the Smith–Waterman alignment algorithm. It is important to stress, however, that embedding sentences using SentenceBERT is computationally much more expensive than using TF-IDF.

Table 5: Performance obtained when using *text*-based representations to tackle the narrative matching task, expressed in terms of F1-score. Values in bold indicate the best performance for a pair of media

Sim.	Alignment	<i>Novels vs. Comics</i>	<i>Novels vs. TV Show</i>	<i>Comics vs. TV Show</i>
<i>tfidf</i>	Thresholding	19.88	15.98	24.13
	Smith–Waterman	48.25	17.21	30.68
<i>sbert</i>	Thresholding	18.01	8.04	20.28
	Smith–Waterman	55.24	22.55	35.96

4.3 Structure-Based Method

Instead of using text as in the previous section, we now leverage character *interactions* to compute a similarity matrix between the narrative units of a media pair. We use the dynamic networks extracted for each adaptation, as described in Section 2.2. Such a network is constituted of a sequence of graphs, each one representing a narrative unit. As explained before, there are two types of dynamic networks: instant vs. cumulative. The best results, which we present here, are obtained with the former. We deal with the latter in the Supplementary Material (see Section SM3.6).

Given a character network per narrative unit for two media, we experiment with several variants of Jaccard’s index to assess their similarity. First, we consider the standard index computed over the sets of vertices present in the compared networks, as well as their sets of edges. Second, we use the *weighted* version of the index, mentioned in Section 3.1.2, a.k.a. Ružička’s similarity [30]. The most straightforward approach to weight each vertex/edge is to use the number of occurrences of the characters/interactions. However, we obtain much better results when weighting according to the *inverse* number of occurrences. We present only these results in the following. We explain this difference by analogy with the IDF part of TF-IDF: using the inverse effectively gives more importance to less frequent characters/interactions. These are generally more typical of a narrative unit, compared to frequent characters, which are more likely to be involved in many narrative units. In the end, we have four ways of measuring the similarity, depending on whether we compare the vertex or edge sets, using the unweighted or weighted version of the index.

Computing all four of these similarity measures requires an exact mapping between characters from a media to another in order to obtain a meaningful score: we compute these using the normalised *name* vertex attribute that we described in 2.3. As explained in Section 2.5, our networks contain characters that are unnamed or not necessarily present in all media. Therefore, we carry on matching experiments by

progressively filtering our networks using the `named`, `common` and `top-20` character sets.

In the rest of this section, we present the results of structural matching as follows. First, in Section 4.3.1, we apply our methods to the same narrative units as with text in Section 4.2: novel chapters, TV show episodes, and comic issues. The structural approach is not bound to these specific units, though, as they were determined by the sources of the textual content, in the first place. One possible issue with them is that they are not on the same scale, i.e. the chunks of plot they cover are not of comparable sizes. For example, TV show episodes usually adapt several chapters of the novels. To study the effect of the narrative unit scale, we experiment with smaller, commensurate narrative units in Section 4.3.2.

4.3.1 Text-Constrained Narrative Units

The best results of structural alignment with the same narrative units as for text, using the thresholding and Smith–Waterman alignment methods, can be found in Table 6, while detailed results with all the configurations are provided in the Supplementary Material (cf. Table SM13). The performance varies widely, with F1-scores ranging from 1.54 to 62.94, depending on the configuration. Overall, by choosing the best configuration, we obtain F1-scores of 62.94, 32.63 and 51.40 for the *Novels vs Comics*, *Novels vs. TV Show* and *Comics vs. TV Show* pairs, respectively. Structural matching leads to better performance than textual matching, for all three pairs of media: +7.7, +10.1 and +15.4 points, respectively. This is further confirmed in results for the *Novels vs. TV Show* pair over the U2 period. As shown in Table SM15 of the Supplementary Material, structural matching obtains an F1-score of 45.00, while textual matching gets 36.65.

Table 6: Performance obtained when using *structure*-based representations and the *text-constrained* narrative units to tackle the narrative matching task, expressed in terms of F1-score. Only the best results over all possible configurations are shown

Alignment	<i>Novels vs. Comics</i>			<i>Novels vs. TV Show</i>			<i>Comics vs. TV Show</i>		
	named	common	top-20	named	common	top-20	named	common	top-20
Thresholding	29.61	34.74	13.31	20.72	23.43	7.34	43.36	46.95	33.33
Smith–Waterman	58.74	62.94	46.81	28.72	32.63	9.87	49.72	51.40	46.86

Based on these results, and as confirmed by Table SM13, it appears that the Smith–Waterman algorithm largely outperforms thresholding for all configurations. Additionally, the `common` character set obtains the best results, while `top-20` severely underperforms and is almost always the worst option. It may be because more minor characters are specific to certain narrative arcs, while main characters are often together as groups, blending several narrative units together which renders matching more difficult. We cannot draw any conclusion regarding our Jaccard weighting scheme or whether computing Jaccard similarity on edges or vertices is better, as the behaviour of these parameters is inconsistent across configurations.

4.3.2 Commensurate Narrative Units

The chapter constitutes the natural narrative unit of the original material (the novels), and the sole we have for this medium. Both other text-constrained narrative units, comic issues and TV show episodes, have a larger scale. Put differently, the piece of plot they convey is longer than for a chapter: both issues and episodes adapt several novel chapters (or, in the case of episodes, sometimes parts of chapters). We hypothesise this hinders the matching performance.

A solution is to adopt smaller narrative units that are closer to chapters. Scenes are available for both media, however, this unit is much smaller compared to novel chapters, which would cause the same scale difference problem as before. Instead, we turn to the intermediary units described in Section 2.2: comic chapters, which are comparable to novel chapters, and TV show blocks, which are sequences of contiguous scenes. We defined the latter in an attempt to split the plot according to changes in character *point-of-views*, the literary device used in novels to unveil the story.

Although we now compare the comics and TV show using chapters and blocks, we must come back to issues and episodes to conduct our assessment, in order to allow a fair comparison with the performance previously presented for the other matching methods. We do so by considering that a novel chapter matching a comic chapter or a block matches the whole issue or episode. Following the same principle, when matching *Comics vs. TV Show*, we match chapters and blocks, but consider issues and episodes to compute the performance.

Table 7: Performance obtained when using *structure*-based representations and the *commensurate* narrative units to tackle the narrative matching task, expressed in terms of F1-score. Only the best results over all possible configurations are shown

Alignment	<i>Novels vs. Comics</i>			<i>Novels vs. TV Show</i>			<i>Comics vs. TV Show</i>		
	named	common	top-20	named	common	top-20	named	common	top-20
Thresholding	25.66	33.23	14.04	21.02	18.72	7.74	36.25	36.71	29.51
Smith–Waterman	71.81	72.30	63.33	34.46	35.09	30.69	60.42	61.78	61.46

Table 7 shows the best alignment results using the commensurate narrative units, while Table SM13 in the Supplementary Material shows the results for all possible configurations. This scheme strongly increases matching performance when using the Smith–Waterman alignment algorithm, with gains of 9.4, 2.5 and 10.4 points for the *Novels vs. Comics*, *Novels vs. TV Show* and *Comics vs. TV Show* pairs respectively. This performance gain is further confirmed over the U2 time period for the *Novels vs. TV Show* pair with a large gain of 17.9 F1 (see Section SM3.5 in the Supplementary Material). Interestingly, even though we extract TV show *blocks* automatically, which may induce errors, using these for matching still leads to better performance than with full episodes. Our results show that ensuring the *scale* of the narrative units used for alignment are comparable is important for performance.

4.4 Hybrid Method

In this section, we strive to combine our textual and structural methods, in order to assess their complementarity. For each pair of media, we adopt a direct approach that consists in computing a new *hybrid* similarity matrix \mathbf{S}_h , based on the structural and similarity matrices, respectively noted \mathbf{S}_s and \mathbf{S}_t . We first rescale \mathbf{S}_s and \mathbf{S}_t separately using min-max normalisation:

$$\mathbf{S}'_{\star} = \frac{\mathbf{S}_{\star} - \min(\mathbf{S}_{\star})}{\max(\mathbf{S}_{\star}) - \min(\mathbf{S}_{\star})}, \quad (6)$$

where \mathbf{S}_{\star} denotes \mathbf{S}_s or \mathbf{S}_t . We then combine the resulting matrices using a weighted sum:

$$\mathbf{S}_h = \alpha \mathbf{S}'_s + (1 - \alpha) \mathbf{S}'_t, \quad (7)$$

where α is a parameter controlling the relative importance of text vs. structure. As for our other parameters, we tune α for each media pair using the other two pairs as a development set.

While this combination method is pretty simple, note that we performed some additional exploratory experiments to combine textual and structural similarity, but that our attempts failed to improve over the results we present in this article. We experimented with training several machine-learning models to compute \mathbf{S}_h from \mathbf{S}_s and \mathbf{S}_t instead of simply summing them. We also tried to perform early fusion by extracting embeddings from our dynamic character networks and combining them with SentenceBERT or TF-IDF vectors, and then experimented with multiple machine learning model to obtain S_c from the resulting representation.

As in Section 4.3, we experiment with matching using the text-constrained narrative units (Section 4.4.1) as well as the commensurate narrative units (Section 4.4.2).

4.4.1 Text-Constrained Narrative Units

Table 8 shows the results obtained when applying our hybrid method on the text-constrained narrative units already used in Sections 4.2 and 4.3.1. As we experiment with all possible configurations of structural and textual matching, we only report the best performance for the sake of simplicity. We provide the full results in Table SM14 of the Supplementary Material.

Table 8: Performance obtained when using *hybrid* representations and the *text-constrained* narrative units to tackle the narrative matching task, expressed in terms of F1-score. Only the best results across configurations are shown

Alignment	<i>Novels vs. Comics</i>	<i>Novels vs. TV Show</i>	<i>Comics vs. TV Show</i>
Thresholding	39.60	26.90	46.75
Smith–Waterman	67.37	30.95	49.16

We find that combining information from textual and structural matching increases performance for the *Novels vs. Comics* pair (+4.4 F1), but fails to improve performance for the *Novels vs. TV Show* (-1.7 F1) and *Comics vs. TV Show* (-2.2 F1) pairs.

4.4.2 Commensurate Narrative Units

As highlighted in Section 4.3.2, performing structural matching on narrative units of comparable sizes can strongly increase performance. Therefore, we now want to conduct hybrid matching by combining text with the commensurate units from Section 4.3.2. Our combination method requires to sum the textual and structural similarity matrices, and therefore, we need the matrices to be of the same dimension. It is not the case though: as already explained, the text-constrained units have a larger scale, and the corresponding similarity matrices are consequently smaller. Moreover, it is not possible to build textual representations at a smaller scale: for the TV show, episodes summaries can not easily be cut to correspond to the underlying blocks, and comics issues summaries cannot easily be split to apply to underlying chapters. Therefore, we propose to artificially *extend* the textual similarity matrix in order to match the dimension of its structural counterpart. We do so by duplicating the rows and columns corresponding to a text-constrained narrative unit (e.g. TV episode) as many times as the number of commensurate units it contains (e.g. blocks).

Table 9: Performance obtained when using *hybrid* representations and the *commensurate* narrative units to tackle the narrative matching task, expressed in terms of F1-score. Only the best results across configurations are shown

Alignment	<i>Novels vs. Comics</i>	<i>Novels vs. TV Show</i>	<i>Comics vs. TV Show</i>
Thresholding	47.62	23.58	46.24
Smith-Waterman	78.50	39.04	63.87

Table 9 shows the best results of hybrid matching on commensurate narrative units, while the results for all configurations are available in the Supplementary Material (Table SM14). As with purely structural matching, working with these narrative units greatly increases performance. Compared with hybrid matching on text-constrained units, we observe gains of +11.1, +8.1 and +11.3 F1 points on the *Novels vs. Comics*, *Novels vs. TV Show* and *Comics vs. TV Show* pairs respectively. Compared with purely structural matching using commensurate units, we also observe improvements across the board: +6.2 F1 point for the *Novels vs. Comics* pair, +4 points for the *Novels vs. TV Show* pair and +2.1 points for the *Comics vs. TV Show* pair. Overall, matching using hybrid similarity on the commensurate narrative units leads to the best performance.

4.5 Takeaways

The way we formalise the narrative matching task make it difficult: we perform many-to-many matching and use F1 as a metric, meaning any mismatch is strictly penalised.

Even so, our best results (78.50 and 63.87 F1 for the *Novels vs. Comics* and *Comics vs. TV Show* pairs, see Figure 6 shows that our proposed alignment method can obtain good performance. Through our experiments, we are able to derive several important insights valuable to the application of the method. First, we note that structure-based matching using dynamic instant character networks outperforms text-based matching, highlighting the usefulness of networks for the task. To our knowledge, this is the first time that character networks are used for narrative alignment. We also show that combining text-based and structure-based similarities can yield better performance than using either alone. Furthermore, we demonstrate the importance of aligning stories using a comparable narrative scale, as taking commensurate narrative units into account consistently improves our results.

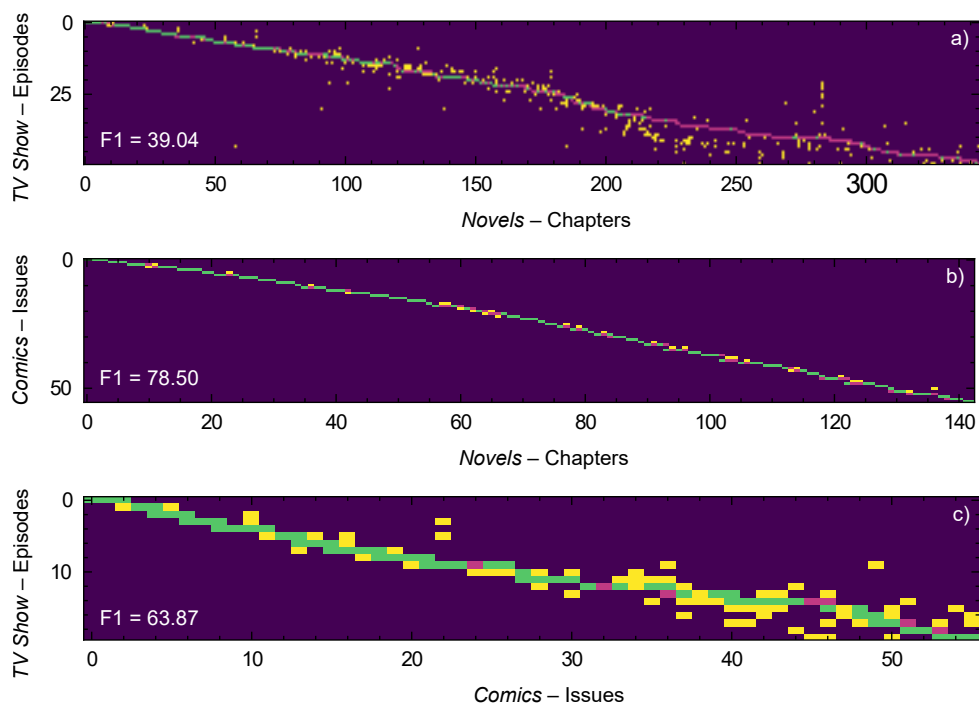


Fig. 6: Best performing alignment for all pairs of media, from top to bottom: a) *Novels vs. TV Show*, b) *Novels vs. Comics*, and c) *Novels vs. Comics*. Green denotes true positives, red false positives, yellow false negatives and purple true negatives

The results on the *Novels vs. TV Show* pair over the U5 period are, however, more lackluster (39.04 F1 at best). We attribute this lower performance to the plot divergence between the original novels and their TV show adaptation, which increased over time. As a confirmation, results restricted to period U2 are much better, with a top F1-score of 64.65 (see Figure SM13 and Table SM15 of the Supplementary Material). As shown in Figure 7, the matching performance progressively decreases as the TV

show progresses for the structural, textual and hybrid similarities alike. Plot divergence is difficult to tackle for the Smith–Waterman algorithm, originally developed to align molecular sub-sequences: it assumes that parts of the sequences it tries to align are ordered similarly, but this assumption is challenged on the *Novels vs. TV Show* pair. As seen in Figure 6, the blocks constituting the later seasons are ordered completely differently from their chapter counterparts, and some chapters are not even adapted, leading to low performance.

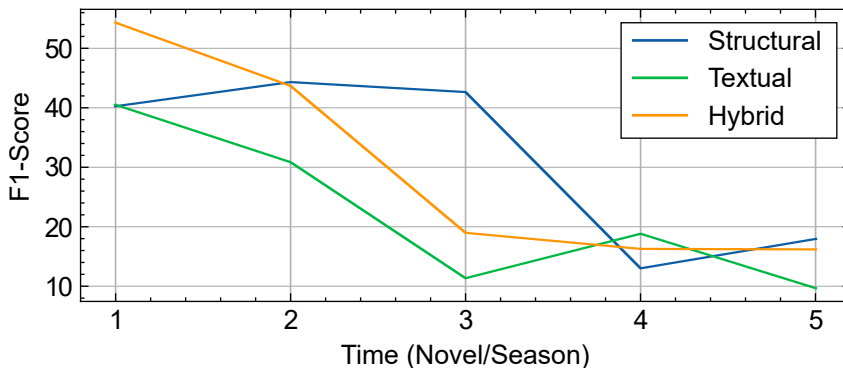


Fig. 7: Performance of the best configurations of narrative alignment over seasons for the *Novels vs. TV Show* pair, expressed in F1-score. As the TV show progressively diverges from the novels, the matching performance starts to decrease. F1-score for last two seasons does not exceed 20

5 Conclusion

In this article, we propose a method based on dynamic character networks to analyse adaptations of a same story across media. This framework is meant to be used to derive insights on how the adaptation process affects the rendition of the plot and characters of a story, depending on the constraints of the target medium. We applied it to adaptations across media of the fantasy series *A Song of Ice and Fire*. We obtained a corpus of three adaptations, including the original novels and two adaptations in the form of comics and TV show. We extracted several types of character networks from these raw data to model the three adaptations. We then focused on two research questions.

The first was to determine whether such structural models allow matching characters from one adaptation to the other. Based on our results obtained with state-of-the-art Graph Matching methods, it appears that character interactions alone are not sufficient to reliably perform such task over *all* characters. However, the performance is much better when focusing on a narrower set of the most *important* characters, which hints at a stronger inter-medium similarity for this category of vertices. Moreover, adding character-related information such as sex or affiliation

greatly improves the results, too. We conducted a centrality study that confirmed these observations.

Our second research question was to determine the possibility of matching the plots of the adaptations themselves, from one medium to the other. We formalised this task as a many-to-many matching problem, in which each narrative unit of one medium can be associated to one or several narrative units of the other medium. We proposed a method based on the computation of an inter-medium similarity matrix, which is then used to estimate a matching matrix. We considered several ways of building the similarity and matching matrices. We experimented with text- and structure-based dynamic representations of the adaptations. According to our results, the structure of the dynamic networks leads to much better performance than the textual representations. Combining them does improve our results, though, which indicates that they are complementary. Our results also show the importance of selecting narrative units of comparable scale to model the adaptations. Finally, our experiments provide objective elements that support the subjective perception of the audience regarding the divergence between the original novels and the TV show adaptation starting novel/season five and onwards.

Our work can be extended in several ways. First, some of our proposed methods could be improved. For the character matching task, the results obtained with the cumulative networks suggest that considering the plot dynamics is promising to increase performance. Concerning the narrative matching task, we hypothesise that performing an early fusion of text and structure by proposing an appropriate representation learning method could help to improve our results. Second, we plan on using this unique corpus to address other research questions. In particular, an interesting point concerns the position of women in the adaptations, and how the TV show differs from the novels. Third, it would be interesting to tackle the same problems on other adaptations across media, in order to study how this affects the performance. A number of other works of fiction were the object of such adaptations: *Harry Potter*, *The Witcher*, *Dune*, etc. However, this would require a significant annotation work, which is why our corpus is unique. The fourth perspective concerns the different tasks proposed in the literature that require to match character networks. For example, some authors compare character networks to real-world ones in order to assess the level of realism of stories [7, 45–47]; others want to automatically distinguish original works from adaptations [3, 4]. These comparisons are typically conducted over static graphs: our proposed methods could be used in this context to take into account the dynamics of the stories. Fifth and finally, an application in which we are most interested in is the analysis of historical and mythological texts. As mentioned earlier, works comparing hagiography and history display different network properties. The social networks of epics and sagas have also been compared (for example see [48]), however the data used here is frequently from medieval manuscripts recorded after centuries of oral transmission. The final recorded version may have significant differences from the original which we no longer have access too. *A Song of Ice and Fire* and two of its adaptations are “closed systems”. By determining differences in character roles, plots, and social networks, insights can be made into how much these can deviate given centuries of oral tradition.

Acknowledgements. We thank Jeffrey Lancaster for constituting his dataset about the *Games of Thrones* TV show and sharing it online, as well as for his feedback. We also thank Tanzir Pial, who kindly provided us with explanations about his implementation and use of the Smith–Waterman algorithm in [16, 17].

References

- [1] Hutcheon, L.: A Theory of Adaptation. Routledge, New York (2006). <https://doi.org/10.4324/9780203957721>
- [2] Labatut, V., Bost, X.: Extraction and analysis of fictional character networks: A survey. *ACM Computing Surveys* **52**(5), 89 (2019) <https://doi.org/10.1145/3344548>
- [3] Chaturvedi, S., Srivastava, S., Roth, D.: Where have i heard this story before? identifying narrative similarity in movie remakes. In: Conference of the North American Chapter of the Association for Computational Linguistics, vol. 2, pp. 673–678 (2018). <https://aclanthology.coli.uni-saarland.de/papers/N18-2106/n18-2106>
- [4] Chowdhury, T., Muhuri, S., Chakraborty, S., Chakraborty, S.N.: Analysis of adapted films and stories based on social network. *IEEE Transactions on Computational Social Systems* **6**(5), 858–869 (2019) <https://doi.org/10.1109/tcss.2019.2931721>
- [5] Massey, S.E.: Form and relationship of the social networks of the new testament. *Social Network Analysis and Mining* **9**, 32 (2019) <https://doi.org/10.1007/s13278-019-0577-7>
- [6] Zhang, C., Zhang, Q., Yu, S., Yu, J.J.Q., Song, X.: Complicating the social networks for better storytelling: An empirical study of chinese historical text and novel. *IEEE Transactions on Computational Social Systems* **8**(3), 754–767 (2021) <https://doi.org/10.1109/TCSS.2021.3061702>
- [7] Gramsch, R., MacCarron, M., MacCarron, P., Yose, J.: Medieval historical, hagiographical and biographical networks. In: *Maths Meets Myths: Quantitative Approaches to Ancient Narratives. Understanding Complex Systems*, pp. 45–69. Springer, ??? (2016). Chap. 4. https://doi.org/10.1007/978-3-319-39445-9_4
- [8] Kinsella, T., Le Brocquy, L.: *The Táin*. Oxford University Press, ??? (2002). <https://global.oup.com/academic/product/the-tin-9780192803733?>
- [9] Kenna, R., MacCarron, P., Casey, D.: Comparative network analysis of the Táin Bó Cúailnge. In: *5th International Conference on the Ulster Cycle of Tales* (2024)
- [10] Beveridge, A., Shan, J.: Network of thrones. *Mathematical Association of America* **23**(4), 18–22 (2016) <https://doi.org/10.4169/mathhorizons.23.4.18>

- [11] Liu, D., Albergante, L.: Balance of thrones: a network study on 'Game of Thrones'. arXiv **cs.SI**, 1707–05213 (2017)
- [12] Beveridge, A., Chemers, M.: The game of game of thrones: Networked concordances and fractal dramaturgy. In: Reading Contemporary Serial Television Universes, pp. 201–225. Routledge, ??? (2018). Chap. 12. <https://doi.org/10.4324/9781315114668-13>
- [13] Garza, P., Yañez, C., Arenas, S., Padilla, F.: Beyond words: Quantitative analysis of complex serial visual narrative through interaction-based modeled observations. *Revista Panamericana de Comunicación* **2**(2), 21–34 (2020) <https://doi.org/10.21555/rpc.v0i2.2333>
- [14] Stavanja, J., Klemen, M., Šubelj, L.: Predicting kills in game of thrones using network properties / napovedovanje umorov v seriji igra prestolov z uporabo analize omrežij. *Uporabna Informatika* **28**(2), 55–65 (2020)
- [15] Gessey-Jones, T., Connaughton, C., Dunbar, R., Kenna, R., Mac Carron, P., O’Conchobhair, C., Yose, J.: Narrative structure of a song of ice and fire creates a fictional world with realistic measures of social complexity. *Proceedings of the National Academy of Sciences* **117**(46), 28582–28588 (2020) <https://doi.org/10.1073/pnas.2006465117>
- [16] Pial, T., Aunti, S., Pethe, C., Kim, A., Skiena, S.: Analyzing film adaptation through narrative alignment. In: Bouamor, H., Pino, J., Bali, K. (eds.) Conference on Empirical Methods in Natural Language Processing, pp. 15560–15579 (2023). <https://doi.org/10.18653/v1/2023.emnlp-main.962>
- [17] Pial, T., Skiena, S.: GNAT: A general narrative alignment tool. In: Bouamor, H., Pino, J., Bali, K. (eds.) Conference on Empirical Methods in Natural Language Processing, pp. 14636–14652 (2023). <https://doi.org/10.18653/v1/2023.emnlp-main.904>
- [18] A Wiki of Ice and Fire: A Game of Thrones (comics) (2021). [https://awoiaf.westeros.org/index.php/A_Game_of_Thrones_\(comics\)](https://awoiaf.westeros.org/index.php/A_Game_of_Thrones_(comics)) Accessed 2024/01/22
- [19] Labatut, V.: Complex network analysis of a graphic novel: The case of the bande dessinée Thorgal. *Advances in Complex Systems* **25**(5&6), 2240003 (2022) <https://doi.org/10.1142/S0219525922400033>
- [20] A Wiki of Ice and Fire: Differences between A Song of Ice and Fire and Game of Thrones (2022). https://awoiaf.westeros.org/index.php/Differences_between_A_Song_of_Ice_and_Fire_and_Game_of_Thrones Accessed 2024/01/22
- [21] Wiki of Westeros: Differences in adaptation/Character names, appearances, or ages (2023). https://gameofthrones.fandom.com/wiki/Differences_in_adaptation/Character_names,_appearances,_or_ages Accessed 2024/01/22

- [22] Wiki of Westeros: Differences in adaptation/Significantly changed characters (2023). https://gameofthrones.fandom.com/wiki/Differences_in_adaptation/Significantly_changed_characters Accessed 2024/01/22
- [23] Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. *Nature* **393**(6684), 440–442 (1998) <https://doi.org/10.1038/30918>
- [24] Qiao, Z., Sussman, D.: iGraphMatch: an r package for the analysis of graph matching. arXiv **stat.CO**, 2112–09212 (2021)
- [25] Lyzinski, V., Fishkind, D.E., Fiori, M., Vogelstein, J.T., Priebe, C.E., Sapiro, G.: Graph matching: Relax at your own risk. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**(1), 60–73 (2016) <https://doi.org/10.1109/tpami.2015.2424894>
- [26] Zaslavskiy, M., Bach, F., Vert, J.-P.: A path following algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**(12), 2227–2242 (2009) <https://doi.org/10.1109/tpami.2008.245>
- [27] Kazemi, E., Hassani, S.H., Grossglauser, M.: Growing a graph matching from a handful of seeds. *Proceedings of the VLDB Endowment* **8**(10), 1010–1021 (2015) <https://doi.org/10.14778/2794367.2794371>
- [28] Umeyama, S.: An Eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10**(5), 695–703 (1988) <https://doi.org/10.1109/34.6778>
- [29] Sussman, D., Park, Y., Priebe, C.E., Lyzinski, V.: Matched filters for noisy induced subgraph detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **42**(11), 2887–2900 (2019) <https://doi.org/10.1109/tpami.2019.2914651>
- [30] Ružička, M.: Anwendung mathematisch-statistischer methoden in geobotanik (synthetische bearbeitung von aufnahmen). *Biologia* **13**, 647–661 (1958)
- [31] De Cáceres, M., Legendre, P., He, F.: Dissimilarity measurements and the size structure of ecological communities. *Methods in Ecology and Evolution* **4**(12), 1167–1177 (2013) <https://doi.org/10.1111/2041-210x.12116>
- [32] Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* **20**(1), 53–65 (1987) [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7)
- [33] Hubert, L., Arabie, P.: Comparing partitions. *Journal of Classification* **2**(1), 193–218 (1985) <https://doi.org/10.1007/BF01908075>
- [34] Manning, C.D., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*.

- Cambridge University Press, Cambridge, UK (2008). <https://doi.org/10.1017/CBO9780511809071>
- [35] Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. *Journal of molecular biology* **147**(1), 195–197 (1981)
- [36] A Wiki of Ice and Fire: Chapter summaries (2023). <https://awoiaf.westeros.org/index.php/Chapters> Accessed 2024/01/22
- [37] Wikipedia: Game of Thrones (season 1) (2023). [https://en.wikipedia.org/wiki/Game_of_Thrones_\(season_1\)](https://en.wikipedia.org/wiki/Game_of_Thrones_(season_1)) Accessed 2024/01/22
- [38] Wikipedia: Game of Thrones (season 2) (2023). [https://en.wikipedia.org/wiki/Game_of_Thrones_\(season_2\)](https://en.wikipedia.org/wiki/Game_of_Thrones_(season_2)) Accessed 2024/01/22
- [39] Wikipedia: Game of Thrones (season 3) (2023). [https://en.wikipedia.org/wiki/Game_of_Thrones_\(season_3\)](https://en.wikipedia.org/wiki/Game_of_Thrones_(season_3)) Accessed 2024/01/22
- [40] Wikipedia: Game of Thrones (season 4) (2023). [https://en.wikipedia.org/wiki/Game_of_Thrones_\(season_4\)](https://en.wikipedia.org/wiki/Game_of_Thrones_(season_4)) Accessed 2024/01/22
- [41] Wikipedia: Game of Thrones (season 5) (2024). [https://en.wikipedia.org/wiki/Game_of_Thrones_\(season_5\)](https://en.wikipedia.org/wiki/Game_of_Thrones_(season_5)) Accessed 2024/01/22
- [42] Spärck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* **60**, 493–502 (1972) <https://doi.org/10.1108/eb026526>
- [43] Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3982–3992 (2019). <https://doi.org/10.18653/v1/D19-1410>
- [44] Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 4171–4186 (2019). <https://doi.org/10.18653/v1/N19-1423>
- [45] Stiller, J., Nettle, D., Dunbar, R.I.M.: The small world of Shakespeare’s plays. *Human Nature* **14**(4), 397–408 (2003) <https://doi.org/10.1007/s12110-003-1013-1>
- [46] Gleiser, P.M.: How to become a superhero. *Journal of Statistical Mechanics* **2007**(09), 09020 (2007) <https://doi.org/10.1088/1742-5468/2007/09/P09020>
- [47] Kenna, R., Mac Carron, P.: Maths meets myths: Network investigations of ancient

- narratives. *Journal of Physics: Conference Series* **681**, 012002 (2016) <https://doi.org/10.1088/1742-6596/681/1/012002>
- [48] Mac Carron, P., Kenna, R.: Universal properties of mythological networks. *Europhysics Letters* **99**(2), 28002 (2012)
- [49] Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. *Software: Practice and Experience* **21**(11), 1129–1164 (1991) <https://doi.org/10.1002/spe.4380211102>
- [50] Silva, M.O., Oliveira, G.P., Moro, M.M.: Analyzing character networks in portuguese-language literary works. In: *Brazilian Workshop on Social Network Analysis and Mining*, pp. 115–126 (2023). <https://doi.org/10.5753/brasnam.2023.230585>
- [51] Masías, V.H., Baldwin, P., Laengle, S., Vargas, A., Crespo, F.A.: Exploring the prominence of Romeo and Juliet’s characters using weighted centrality measures. *Digital Scholarship in the Humanities* **32**(4), 837–858 (2017) <https://doi.org/10.1093/llc/fqw029>
- [52] Pang, N., Sun, M., Zhu, H.: Louise or Ferdinand? exploring the protagonists of Love and Intrigue using social network analysis. *Digital Scholarship in the Humanities* **in press** (2023) <https://doi.org/10.1093/llc/fqad007>

Supplementary Material

Interconnected Kingdoms: Comparing ‘A Song of Ice and Fire’ Adaptations Across Media Using Complex Networks

Arthur Amalvy^{1*}, Madeleine Janickyj^{3,4*}, Shane Mannion^{2*}, Pádraig MacCarron^{2*} and Vincent Labatut^{1*}

SM1 Dataset and Descriptive Analysis

This section aims at providing more details regarding the dataset and its preparation. Section SM1.1 is dedicated to the raw data and the adaptations themselves, whereas Section SM1.2 focuses on the networks that we extract from these adaptations.

SM1.1 Adaptations

In this section, we give more information regarding the three adaptations studied in the main article (Section SM1.1.1), and the way we constitute the character sets used in our experiments (Section SM1.1.2).

SM1.1.1 Organisation

The original material is constituted of five published books out of a total of seven envisioned novels:

1. *A Game of Thrones* (1996)
2. *A Clash of Kings* (1998)
3. *A Storm of Swords* (2000)
4. *A Feast for Crows* (2005)
5. *A Dance with Dragons* (2011)
6. *The Winds of Winter* (forthcoming)
7. *A Dream of Spring* (planned)

A few draft chapters of the last two novels have also been published online. A part of these were integrated in the TV show, in addition to the first five novels. The showrunner also had access to unpublished material (and to the author). By comparison, the comics aim to be a straightforward adaptation of the first two novels [18]. Table SM1 shows how the three adaptations break down into various types of narrative units, whereas Figure SM1 shows the overlap between the three adaptations, in terms of their largest narrative units: books, volumes, and seasons.

Table SM1: Numbers of narrative units for all three adaptations studied in the main article

Narrative unit	Novels	Comics	TV Show
Scenes	–	1,437	4,165
Blocks	–	–	739
Chapters	344	143	–
Issues/Episodes	–	56	73
Books/Volumes/Seasons	5	2	8

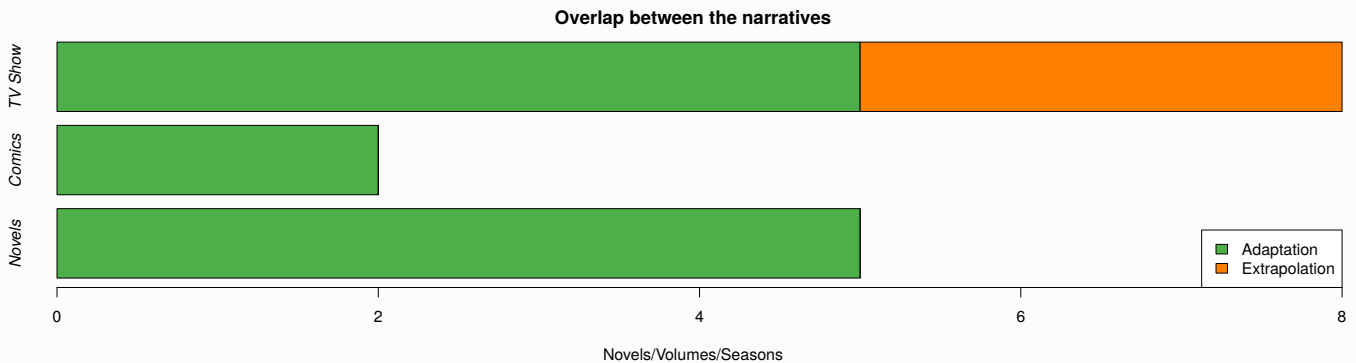


Fig. SM1: Overlap between the three considered adaptations, in terms of books (for the novels), volumes (for the comics), and seasons (for the TV show)

SM1.1.2 Characters

In order to normalise the character names, we first scrape a *main list* of names from the Wiki mentioned in the main article, and dedicated to the novels: *A Wiki of Ice and Fire*¹² (AWoIaF). We then list the names appearing in all three adaptations, and automatically match them to similar names in the main list. Next, the many remaining names are matched manually. The TV show contains a number of additional characters compared to the novels. We complement the main list based on the other Wiki mentioned in the main article, *Game of Thrones Wiki*¹³ (GoTW), which is dedicated to the TV show.

In the process, we identify and delete a few spurious characters, e.g. the names of some actors and staff in the *TV Show* dataset, and some expressions not matching any character in the *Novels* dataset. We also identify some characters appearing twice under different names (e.g. *The Three-Eyed Raven* and *The Last Greenseer*). Sometimes, different versions of the same character are explicitly distinguished (e.g. normal vs. young, or live vs. wight): we merge them for the sake of consistency.

After this, the main list contains a total of 3,863 characters (some of them appearing in other media than those we consider in this work): 3,778 characters from the AWoIaF Website, and 85 additional characters retrieved from the GoTW Website. In the end, we can leverage this list to produce conversion maps specific to each medium, allowing to normalise all character names.

Table SM2: Lists of the 20 most important characters (top-20) for periods U2 (left) and U5 (right), by decreasing order of importance. The scores correspond to normalised numbers of occurrences (cf. Section 2.5 in the main article). Stars indicate the four characters that are present only in the left or only the right table

Character	Affiliation	Sex	Nov.	Com.	TV	Mean	Character	Nov.	TV	Mean
Tyrion Lannister	House Lannister	M	0.62	1.00	1.00	0.87	Tyrion Lannister	0.76	1.00	0.88
Eddard Stark	House Stark	M	1.00	0.64	0.58	0.74	Cersei Lannister	0.86	0.76	0.81
Arya Stark	House Stark	F	0.62	0.88	0.62	0.71	Jon Snow	0.57	0.89	0.73
Catelyn Stark	House Stark	F	0.65	0.80	0.49	0.65	Sansa Stark	0.70	0.68	0.69
Jon Snow	House Stark	M	0.50	0.78	0.60	0.63	Arya Stark	0.67	0.65	0.66
Sansa Stark	House Stark	F	0.64	0.57	0.54	0.59	Eddard Stark	1.00	0.25	0.62
Bran Stark	House Stark	M	0.63	0.63	0.40	0.56	Jaime Lannister	0.75	0.47	0.61
Cersei Lannister	House Lannister	F	0.76	0.30	0.57	0.54	Robb Stark	0.89	0.30	0.59
Daenerys Targaryen	House Targaryen	F	0.20	0.75	0.62	0.52	Joffrey Baratheon	0.77	0.41	0.59
Robb Stark	House Stark	M	0.76	0.29	0.42	0.49	Tywin Lannister*	0.73	0.31	0.52
Joffrey Baratheon	House Baratheon	M	0.68	0.24	0.44	0.45	Daenerys Targaryen	0.31	0.72	0.52
Theon Greyjoy	House Greyjoy	M	0.25	0.42	0.52	0.39	Stannis Baratheon	0.71	0.31	0.51
Robert I Baratheon	House Baratheon	M	0.80	0.15	0.15	0.37	Robert I Baratheon	0.92	0.06	0.49
Petyr Baelish	House Baelish	M	0.39	0.26	0.36	0.34	Catelyn Stark	0.67	0.32	0.49
Jaime Lannister	House Lannister	M	0.68	0.11	0.21	0.33	Bran Stark	0.56	0.37	0.46
Sandor Clegane	House Lannister	M	0.38	0.19	0.37	0.31	Sandor Clegane	0.42	0.31	0.37
Jorah Mormont	House Targaryen	M	0.19	0.32	0.37	0.29	Samwell Tarly*	0.20	0.49	0.35
Stannis Baratheon	House Baratheon	M	0.53	0.07	0.24	0.28	Theon Greyjoy	0.31	0.37	0.34
Renly Baratheon*	House Baratheon	M	0.47	0.17	0.18	0.27	Jorah Mormont	0.20	0.47	0.33
Varys*	House Baratheon	M	0.33	0.18	0.28	0.26	Petyr Baelish	0.35	0.30	0.33

Table SM2 provides the lists of the 20 most important characters, according to the method described in the main article, for periods U2 (first two books, volumes, and seasons) and U5 (first five books and seasons). Period U5 concerns only the novels and TV show, since the comics only cover the first two books. The characters are almost all the same in both lists, albeit in a different order, except for Renly Baratheon and Varys (only in U2), and Samwell Tarly and Petyr Baelish (only in U5). In addition to the character names, both lists contain the scores used to rank the characters in each adaptation and overall, which correspond to max-normalised numbers of occurrences. Finally, the left list also shows the *Sex* and *Affiliation* attributes, later used to match the characters from one network to the other.

SM1.2 Networks

Figure SM2 shows the static character network obtained for each adaptation, when considering the largest period it covers: five books for the novels, two volumes for the comics, and eight seasons for the TV show. The 5 most important characters are represented in colour, and vertex size reflects the importance score (see Table SM2). Edge thickness corresponds to the number of interactions over the considered period. We used the Fruchterman–Reingold method [49] to provide similar layouts and ease visual comparison. The *Novels* network gives the impression of being denser, but this is not the case: as shown by Table 1 from the main article, the TV Show network is clearly the densest.

Figure SM3 represents, for each adaptation, the subnetwork of the 20 most important characters overall, for period U2 (i.e. first two novels, volumes, and seasons). As in Figure SM2, the layout is based on the Fruchterman–Reingold method [49]. However, this time we fix the layout across networks, as the characters are exactly the same (by construction) for all three of them. Vertex colour represents character importance.

Table SM3 shows the same topological measure as in Table 1 from the main article, but for the three different character sets (and not *all* of them). The increasing filtering of characters obviously affects the number of vertices, but also increases

¹²https://awoiaf.westeros.org/index.php/Main_Page

¹³https://gameofthrones.fandom.com/wiki/Game_of_Thrones_Wiki

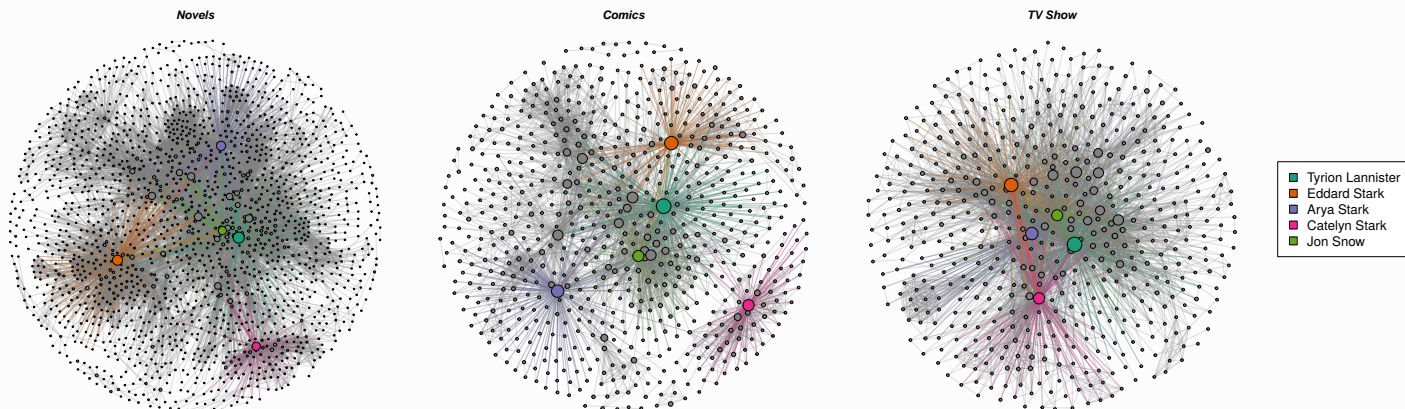


Fig. SM2: Static networks including all the characters, for all three adaptations, for the longest period they cover. The 5 most important characters are highlighted in colour

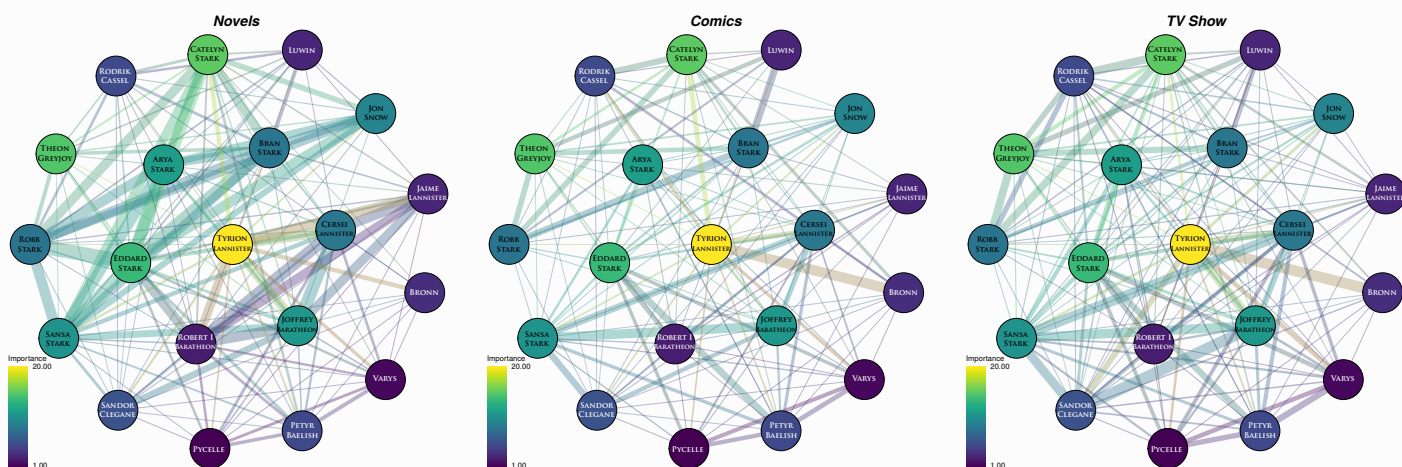


Fig. SM3: Static networks limited to the top20 character set, for all three adaptations, for period U2 (first two novels, volumes, and seasons). The vertex layout is fixed for all three graphs

the density. This means that the more important characters are, the more they tend to be interconnected. This also shows in the assortativity, which tends to increase (even if it remains quite low), the average clustering coefficient, which also increases, and the modularity, which decreases, indicating that the important characters are more tightly knit.

Figure SM4 shows the evolution of the number of vertices (i.e. characters) in each adaptation, over books/volumes/seasons. By comparison, Figure 1 from the main article includes all characters (named or not), and shows that 1) their number increases linearly for all adaptations, 2) faster for novels and comics than for the TV show, and 3) the growth rate is similar for novels and comics. When considering only named characters in Figure SM4, we see that that the first two observations still holds, but not the third one: comics exhibit a smaller growth rate. This is because the annotations for this medium include a large proportion of unnamed characters (typically, standing in the background and only witnessing a scene). When focusing on characters that are common to the three media, all of them exhibit a sublinear growth of the number of vertices, and a very similar evolution.

Figure SM5 shows the evolution of the average degree over time. By comparison, Figure 1 from the main article includes all characters, and shows that it is relatively stable in the TV show, whereas the novels and comics are characterised by an increase during the first seasons, before stabilising too. These observations holds when focusing on named and common characters (Figure SM5). Moreover, novels and comics appear to be very similar on this aspect.

SM2 Character Comparison

This section aims at providing additional results for Section 3 from the main article. Like in the main article, we first consider the GM problem used to match characters between adaptations (Section SM2.1), before turning to a centrality analysis of the characters (Section SM2.2).

SM2.1 Graph Matching

In this section, we present the comprehensive results obtained when applying the five selected state-of-the-art GM methods to our networks. By comparison, the main article (Section 3.1) focuses only on the best methods, for the sake of concision.

Table SM3: Network statistics for each of the time-periods, giving the number of vertices n and edges L , the density δ , the mean degree $\langle k \rangle$, the average shortest path length $\langle \ell \rangle$, the average clustering coefficient $\langle C \rangle$, the degree assortativity r , and the modularity Q . This table shows the statistics for each character set: by comparison, Table 1 from the main article deals with *all* available characters

Characters	Period	Adaptation	n	L	δ	$\langle k \rangle$	$\langle \ell \rangle$	$\langle C \rangle$	r	Q
named	U2	Novels	731	4,959	0.019	13.57	2.98	0.60	0.00	0.53
		Comics	324	2,039	0.039	12.59	2.88	0.72	-0.03	0.68
		TV Show	167	1,185	0.085	14.19	2.83	0.76	0.11	0.58
	U5	Novels	1,877	13,859	0.008	14.77	3.32	0.58	-0.02	0.58
		TV Show	285	2,090	0.052	14.67	2.71	0.75	-0.01	0.61
	U8	TV Show	348	2,974	0.049	17.09	2.51	0.76	-0.07	0.42
common	U2	Novels	123	907	0.121	14.75	2.38	0.68	0.12	0.45
		Comics	123	699	0.093	11.37	2.67	0.68	0.06	0.62
		TV Show	123	850	0.113	13.82	2.69	0.76	0.16	0.54
	U5	Novels	216	1,859	0.080	17.21	2.46	0.67	0.03	0.47
		TV Show	216	1,534	0.066	14.20	2.65	0.76	0.05	0.59
	U8	TV Show	153	1,330	0.114	17.39	2.28	0.74	-0.02	0.42
top-20	U2	Novels	20	127	0.668	12.70	1.41	0.79	-0.08	0.01
		Comics	20	111	0.584	11.10	1.28	0.73	0.13	0.13
		TV Show	20	123	0.647	12.30	1.20	0.82	0.15	0.09
	U5	Novels	20	127	0.668	12.70	1.34	0.83	-0.07	0.01
		TV Show	20	117	0.616	11.70	1.44	0.88	0.22	0.33
	U8	TV Show	20	151	0.795	15.10	1.21	0.92	-0.19	0.00

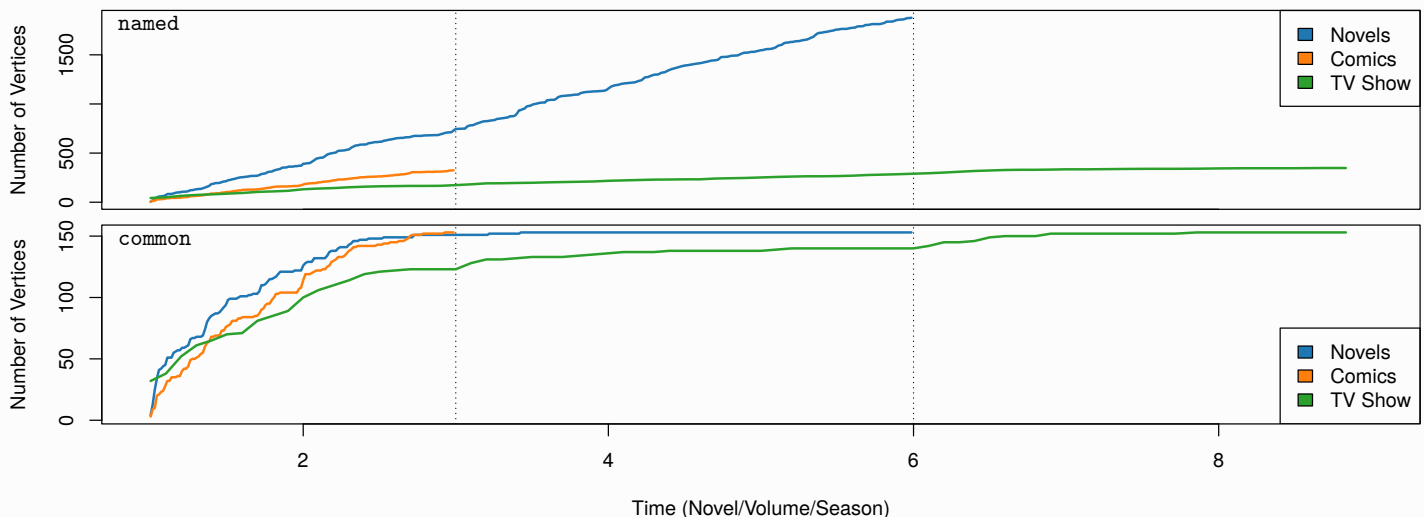


Fig. SM4: Evolution of the number of vertices in each adaptation, for the **named** (top) and **common** (bottom) character sets. Table 1 from the main article provides the same plot, but for all existing characters

SM2.1.1 Basic Methods

Table SM4 shows the results obtained when applying all methods to all pairs of networks, without using any seeds or vertex attributes, with and without adjacency matrix centring. When applying the percolation method, which requires at least one seed, we use the most important character.

When considering all available **named** characters, there is no unique best approach, as their performance vary over network pairs. The best results are obtained when comparing the comics and TV show. Centring does not seem to affect the results much.

Focusing only on **common** characters (i.e. named characters appearing in *both* compared graphs) systematically improves the performance. This allows avoiding padding the smaller graphs with many isolates in order to get same-sized graphs, which makes the problem much easier. The *Convex* method produces the best performance overall, with *Concave* a close second. Again, the best match is between the *Comics* and *TV Show* networks. It seems that centring does not affect performance in our case, or even makes it worse, so we do not use it in the rest of our experiments.

Finally, the performance is even better when focusing only on the 20 most important characters (**top-20**). This could be due to the networks being more similar in terms of interconnections between these specific characters, or this could be just because of the smaller number of vertices.

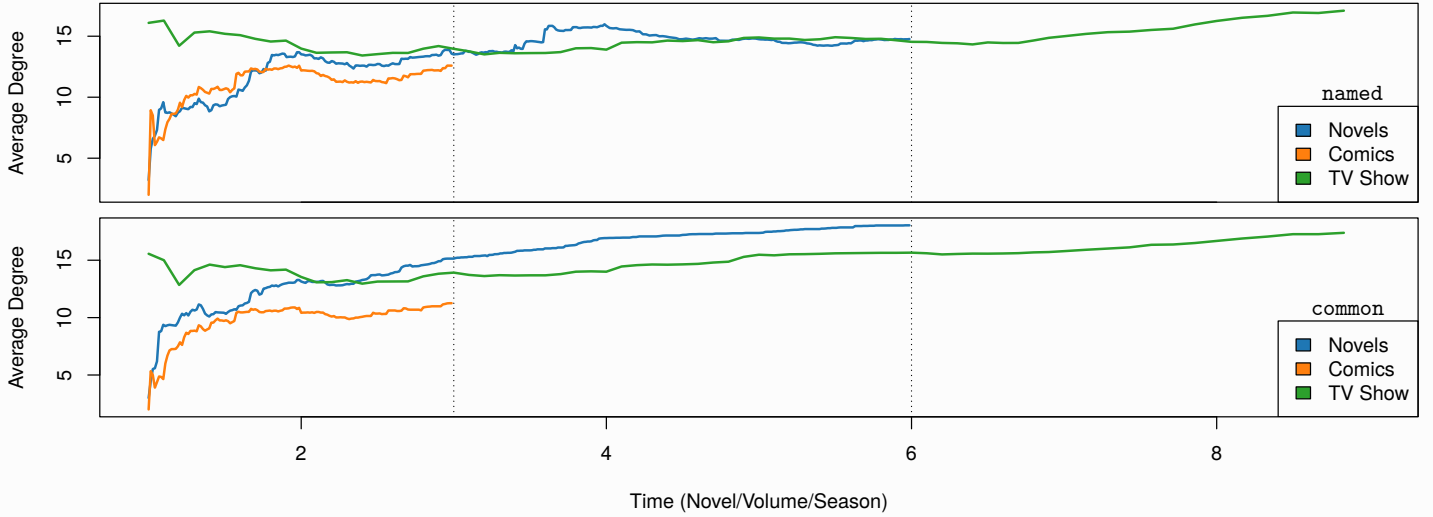


Fig. SM5: Evolution of the average degree in each adaptation, for the **named** (top) and **common** (bottom) character sets. Table 1 from the main article provides the same plot, but for all existing characters

Table SM4: Graph matching results obtained for all methods when using no seeds and no attributes. The table highlights how centring affects the performance, which is expressed in terms of correctly matched for each pair of networks, considering the three character sets used throughout the paper: all named characters (**named**), only those common to both compared networks (**common**), and only the 20 most important characters (**top-20**)

Method	Centring	<i>Novels vs. Comics</i>			<i>Novels vs. TV Show</i>			<i>Comics vs. TV Show</i>		
		named	common	top-20	named	common	top-20	named	common	top-20
Convex	Yes	0.13%	3.42%	10.00%	0.00%	2.16%	0.00%	0.28%	16.67%	10.00%
Indefinite	Yes	0.00%	1.03%	5.00%	0.00%	10.07%	0.00%	0.00%	4.55%	50.00%
Concave	Yes	0.00%	2.05%	20.00%	0.00%	1.44%	0.00%	0.28%	21.97%	10.00%
Percolation	Yes	0.13%	2.40%	15.00%	0.13%	2.88%	20.00%	0.57%	5.30%	10.00%
Umeyama	Yes	0.13%	0.00%	10.00%	0.00%	0.72%	5.00%	0.00%	0.76%	10.00%
Convex	No	0.40%	3.08%	10.00%	0.00%	0.72%	5.00%	5.10%	16.67%	25.00%
Indefinite	No	0.26%	1.03%	5.00%	0.00%	1.44%	0.00%	2.83%	10.61%	35.00%
Concave	No	0.53%	2.05%	5.00%	0.13%	2.16%	5.00%	4.82%	21.21%	30.00%
Percolation	No	0.53%	3.77%	20.00%	0.26%	3.60%	25.00%	0.85%	10.61%	15.00%
Umeyama	No	0.00%	0.00%	15.00%	0.00%	0.00%	0.00%	0.00%	0.76%	5.00%

SM2.1.2 Adaptive Seeds

As explained in the main article, the adaptive seeds approach is an iterative method consisting in first applying one of the previous algorithms as before, and selecting the first few best matches according to some heuristic measure. These are then used as *hard* seeds in the next iteration (as if they were ground truth). By using an increasing number of seeds at each iteration, the matching is supposed to get better and better. Alternatively, it is possible to use the previous best matches as *soft* seeds, which allow including some uncertainty by allowing multiple matching.

Table SM5: Graph matching results obtained for all methods when using adaptive seeds, no attributes, and no centring. The table compares the use of *hard* (top part) and *soft* (bottom part) adaptive seeds. Performance is expressed as in Table SM4

Method	Adaptive Seed	<i>Novels vs. Comics</i>			<i>Novels vs. TV Show</i>			<i>Comics vs. TV Show</i>		
		named	common	top-20	named	common	top-20	named	common	top-20
Convex	Hard	0.40%	6.16%	15,00%	1.45%	0,00%	5,00%	12.36%	34,85%	10,00%
Indefinite	Hard	0.13%	0,00%	0,00%	0.13%	11,51%	0,00%	2.81%	4,55%	0,00%
Concave	Hard	1.32%	5,14%	10,00%	1.32%	10,79%	5,00%	14.89%	28,03%	10,00%
Percolation	Hard	4.35%	25,34%	0,00%	0.00%	2,88%	0,00%	1.97%	0,76%	50,00%
Umeyama	Hard	0.00%	0,00%	15,00%	0.00%	0,00%	0,00%	0.00%	0,76%	5,00%
Convex	Soft	0.40%	1,71%	0,00%	0.79%	12,23%	5,00%	15.73%	30,30%	0,00%
Indefinite	Soft	0.26%	2,40%	0,00%	0.13%	5,76%	0,00%	1.12%	0,00%	0,00%
Concave	Soft	0.40%	2,74%	10,00%	0.92%	9,35%	15,00%	9.83%	21,97%	10,00%
Percolation	Soft	0.40%	4,79%	15,00%	0.66%	6,47%	25,00%	3.09%	8,33%	15,00%
Umeyama	Soft	0.00%	0,00%	15,00%	0.00%	0,00%	0,00%	0.00%	0,76%	5,00%

Table SM5 shows the results obtained when applying the adaptive seeds method using all the previous algorithms. Considering *hard* seeds, compared to the seedless approach, there is a clear improvement for the *Percolation* algorithm when applied to the *Novels vs. Comics*, and for the *Indefinite* algorithm when applied to the *Comics vs. TV Show*. No method dominates the others, as each best score for a network pair was obtained through a different algorithm. Using *soft* seeds instead leads to lesser results, and the best methods are not the same. The best performance for the *Novels vs. Comics* networks, obtained with the *Concave* method, increases, though.

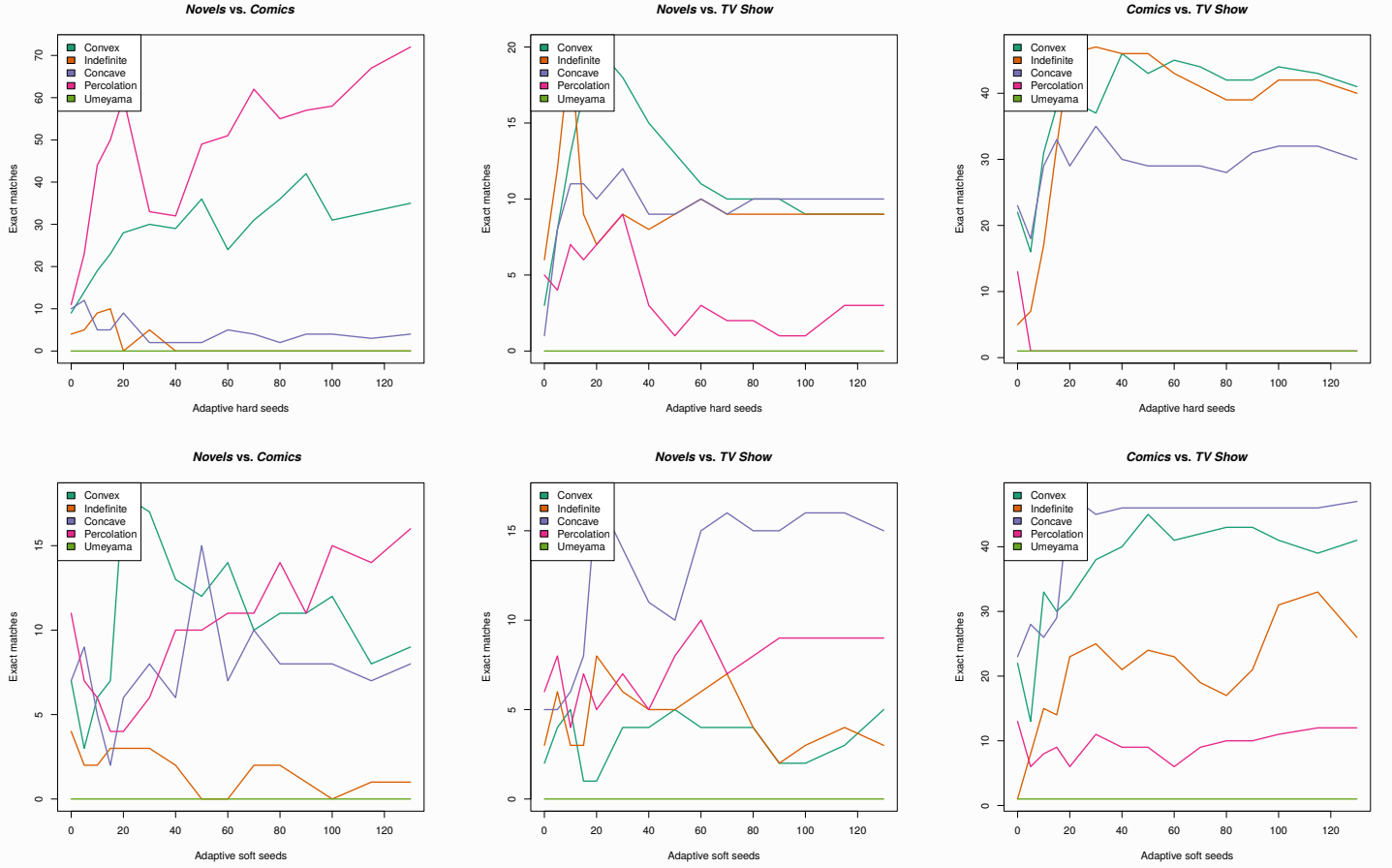


Fig. SM6: Evolution of the number of matches as a function of the number of *hard* (top row) and *soft* (bottom row) adaptive seeds, leading to the results shown in the top part of Table SM5

Figure SM6 shows how the number of exact matches evolves as a function of the number of adaptive seeds. The rightmost points in the plots correspond to the values from Table SM5. It shows that no algorithm dominates the others on more than one comparison. Moreover, the performance is not always improved by increasing the number of seeds: for instance, in the *Novels vs. TV Show* comparison, the *Convex* algorithm correctly matches 20 characters when using 22 seeds, but its performance is only 9 with 130 seeds.

We also experiment with what we call adaptive *temporal* seeds, which takes time into account. Instead of iteratively applying the vertex matching algorithms using an increasing number of seeds on the same static network, we do so on the series of time slices constituting the dynamic networks, using the matches from the previous time slice as the seeds of the next one. We only focus on the *Novels vs. Comics* comparison, as they have the same temporal scale (chapters), and use the cumulative networks in order to be sure to find the previous best matched characters in the next time slice. We focus on the hard seeds, which seem to work better on the static networks. The method does not perform well for our dataset, with all scores almost zero (not shown here).

SM2.1.3 Ground Truth Seeds

As explained in the main article, we relax the problem by leveraging hard seeds from the ground truth. We consider using the 5, 10 and 15 most important characters as seeds. Table SM6 presents the obtained performances.

Increasing the number of seeds increases the performance, for (almost) all network pairs and all matching methods. It is worth noting that *Comics vs. TV Show*, the network pair whose performance is the best without seeds, gets the smallest improvement when increasing the seed number. As observed before, the performance is systematically the lowest for the named character set, and the higher for the top-20 character set.

Table SM6: Graph matching results obtained for all methods when using ground truth seeds, no attributes, and no centring. Performance is expressed as in Tables SM4 & SM5

Method	Seeds	<i>Novels vs. Comics</i>			<i>Novels vs. TV Show</i>			<i>Comics vs. TV Show</i>		
		named	common	top-20	named	common	top-20	named	common	top-20
Convex	5	3.99%	10.80%	60.00%	2.39%	20.15%	26.67%	12.36%	38.58%	86.67%
Indefinite	5	3.06%	7.67%	20.00%	1.99%	20.15%	13.33%	10.34%	36.22%	80.00%
Concave	5	4.26%	14.63%	60.00%	2.25%	20.90%	26.67%	12.07%	35.43%	86.67%
Percolation	5	0.53%	8.01%	26.67%	1.06%	5.22%	40.00%	4.31%	18.90%	46.67%
Umeyama	5	0.00%	0.00%	13.33%	0.00%	0.00%	0.00%	0.00%	1.57%	0.00%
Convex	10	6.43%	20.92%	80.00%	2.40%	27.13%	60.00%	10.79%	38.52%	100.00%
Indefinite	10	5.09%	12.06%	40.00%	2.27%	27.91%	60.00%	9.91%	42.62%	100.00%
Concave	10	6.29%	23.40%	80.00%	2.54%	27.13%	60.00%	11.66%	40.16%	100.00%
Percolation	10	2.01%	9.93%	60.00%	0.53%	11.63%	40.00%	6.71%	20.49%	50.00%
Umeyama	10	0.00%	0.00%	10.00%	0.00%	0.00%	10.00%	0.00%	0.82%	20.00%
Convex	15	7.01%	22.02%	100.00%	3.49%	33.87%	60.00%	10.36%	35.90%	100.00%
Indefinite	15	6.87%	24.55%	100.00%	2.28%	29.84%	60.00%	8.28%	39.32%	100.00%
Concave	15	7.01%	26.35%	100.00%	3.76%	33.06%	60.00%	12.95%	39.32%	100.00%
Percolation	15	5.39%	18.41%	60.00%	1.48%	13.71%	60.00%	5.62%	17.09%	60.00%
Umeyama	15	0.00%	0.00%	40.00%	0.00%	0.00%	60.00%	0.00%	0.85%	20.00%

SM2.1.4 Vertex Attributes

We relax the problem differently by leveraging some vertex attributes to help the methods matching the characters. The algorithms implemented in `iGraphMatch` take an optional vertex similarity matrix as input, which can be used to leverage additional knowledge. In particular, it makes it possible to indirectly consider vertex attributes. A simple way of doing so in the case of categorical attributes is to put a 1 if the characters have the same attribute value, and 0 otherwise. We experiment with attributes sex, character affiliation, and both.

Table SM7: Graph matching results obtained for all methods when using vertex attributes, no seeds, and no centring. Performance is expressed as in Tables SM4–SM6

Method	Attr.	<i>Novels vs. Comics</i>			<i>Novels vs. TV Show</i>			<i>Comics vs. TV Show</i>		
		Named	Common	Top-20	named	common	top-20	named	common	top-20
Convex	Sex	0.13%	2.74%	5.00%	0.13%	0.72%	0.00%	0.28%	3.79%	10.00%
Indefinite	Sex	0.00%	3.77%	10.00%	0.13%	5.76%	5.00%	0.28%	5.30%	80.00%
Concave	Sex	0.00%	2.05%	15.00%	0.00%	2.16%	0.00%	0.00%	3.03%	10.00%
Percolation	Sex	0.40%	3.42%	10.00%	0.66%	7.91%	15.00%	0.85%	16.67%	35.00%
Umeyama	Sex	0.13%	0.00%	15.00%	0.00%	0.72%	0.00%	0.00%	2.27%	5.00%
Convex	Aff.	0.00%	0.00%	0.00%	0.00%	0.00%	5.00%	0.00%	0.00%	0.00%
Indefinite	Aff.	9.51%	65.75%	60.00%	3.43%	63.31%	40.00%	9.07%	81.06%	80.00%
Concave	Aff.	0.00%	0.00%	0.00%	0.00%	0.72%	0.00%	0.00%	0.00%	10.00%
Percolation	Aff.	14.93%	57.88%	15.00%	6.32%	64.75%	20.00%	15.58%	58.33%	80.00%
Umeyama	Aff.	0.13%	1.03%	5.00%	0.13%	3.60%	0.00%	0.00%	0.76%	5.00%
Convex	Both	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Indefinite	Both	9.51%	73.97%	65.00%	3.43%	64.75%	55.00%	8.50%	83.33%	100.00%
Concave	Both	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	10.00%
Percolation	Both	16.12%	56.85%	10.00%	6.46%	54.68%	20.00%	15.86%	61.36%	80.00%
Umeyama	Both	0.13%	0.68%	5.00%	0.13%	3.60%	0.00%	0.00%	0.76%	5.00%

Table SM7 shows the performance obtained without any seed or centring. Leveraging the attributes clearly improves the results for the *Indefinite*, *Concave* and *Percolation* algorithms, especially the former. The effect is much stronger for affiliation than for sex, but the latter still helps to improve the results a little bit. As before, focusing only on the characters which are common to both compared networks improves performance, and even more so for the top-20 character set.

Interestingly, using ground truth seeds in addition to attributes only marginally improves the results (not shown here). This suggests that both types of additional information are equally relevant for the character matching task.

SM2.1.5 Vertex Similarity

As in the main article, we use Ružička’s similarity to measure the similarity between two characters based on their respective neighborhoods. Figure SM7 shows the similarity matrices obtained for all characters, when focusing only on the common character set. Each matrix corresponds to a specific pair of adaptations. It is important to stress that these matrices are not symmetric, as the compared pairs of characters belong to two distinct networks. Therefore, the similarity between a character v_1 in the novels and a character v_2 in the comics is not necessarily equal to that between v_2 in the novels and v_1 in the comics.

In all three matrices, the diagonal appears clearly, i.e. the similarity between both instances of the same character is high. However, a number of off-diagonal cells are also highlighted, which means that there are potential mismatches. It is even

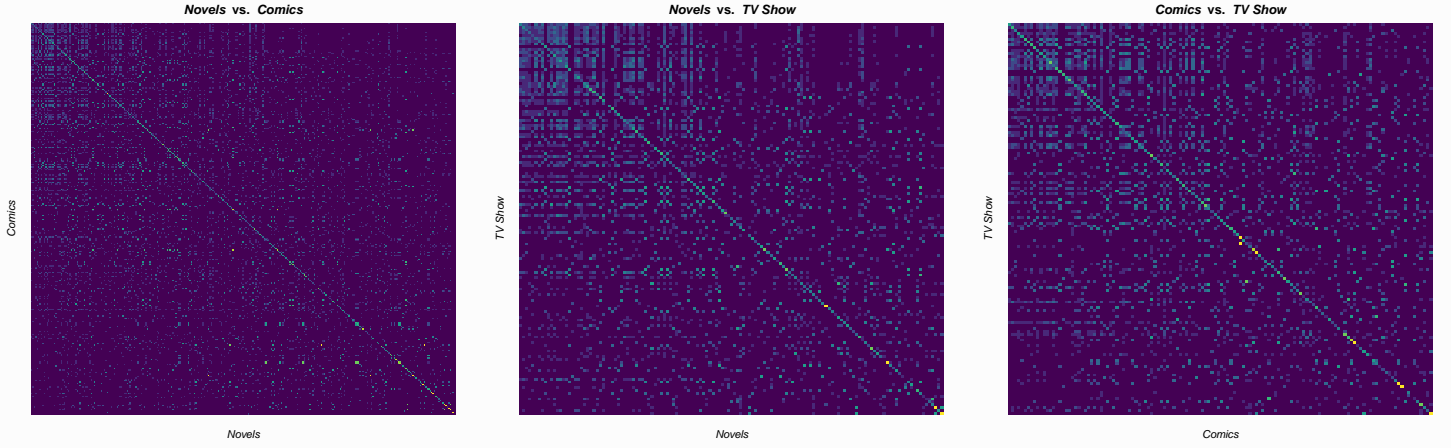


Fig. SM7: Similarity matrices obtained with Ružička’s similarity, for the **common** character set and period U2. By comparison, in Figure 2, the main article focuses on **top-20**, i.e. the 20 most important characters

more the case when considering all **named** characters (not shown here). The main article provides similar matrices focusing on **top-20**, the 20 most important characters (Figure 2), showing that the number of off-diagonal high values (i.e. mismatches) appears lower for this subset of characters.

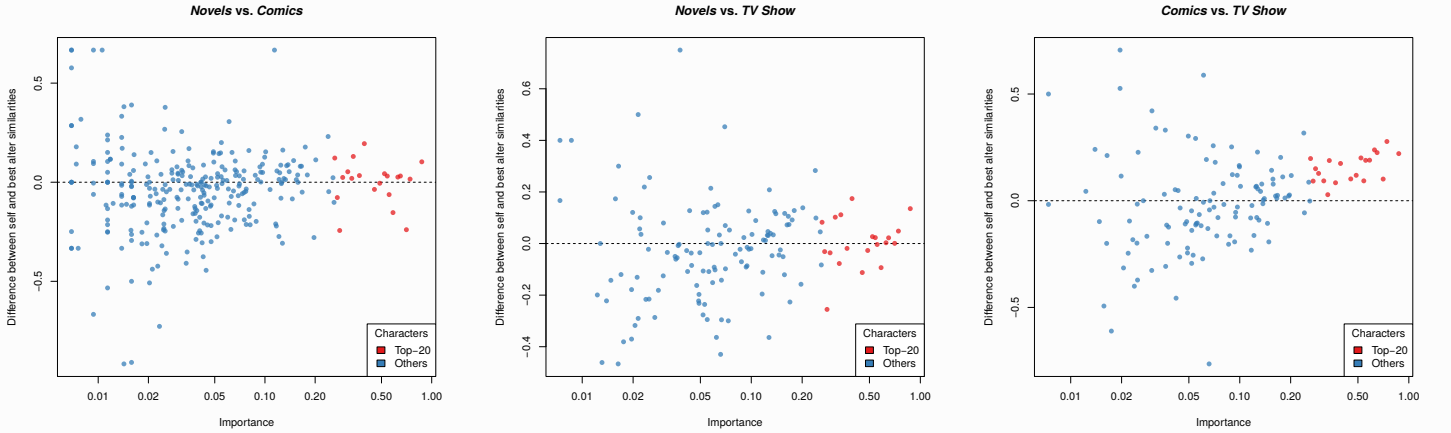


Fig. SM8: Similarity difference between, on one side, a character and itself, and, on the other side, the same character and the most similar alternate character, as a function of character importance. The considered character set is **common**. Notice the logarithmic scale on the y -axis

We call *self-similarity* the similarity between a character in a given network, and the same character in a different network. The *best alter-similarity* is the similarity between the same character and the best alternate character in the second network. Consequently, the difference between these quantities indicates whether the match is correct (positive value, i.e. the best match is the same character) or not (negative value, i.e. the best match is another character). Figure SM8 shows this difference as a function of character importance. The most important characters are located on the right side of the plots, the 20 most important characters being represented in red. These **top-20** characters tend to be located in the positive half (i.e. they are correctly matched), especially for the *Comics vs. TV Show* comparison. The correlation between these variables is intermediary (0.36) but significant ($p < .001$) for *Comic vs. TV Show*, according to Spearman’s coefficient. On the contrary, there is no significant correlation for both other pairs of adaptations (*Novels vs. Comics*, and *Novels vs. TV Show*).

SM2.1.6 Characters of Interest

Based on the fan Wikis^{7,8} mentioned in Section 2.3 of the main article, we elaborate two list of characters of particular interest [20–22]. The first one contains *pairs* of characters that have a different name in the novels and TV show, but are known to correspond to the same individual, sometimes with various additional differences due to the adaptation process. Unlike the **named** character set, **common** and **top-20** do not contain the pair of characters of interest, by construction. For this reason, in the first case the computation of character similarity is exactly as described in Section 3.1.2 of the main article, but the two other character sets require adjusting the procedure. This is done by forcing the inclusion of the listed characters of interest in **common** and **top-20**.

Table SM8 shows the result of the matching process based on Ružička’s similarity. Each row corresponds to a pair of characters, and shows the difference between, on one side, the similarity between the characters of the considered pair, and,

Table SM8: Similarity difference between, on one side, pairs of characters of interest from the Novels and TV Show, and, on the other side, the best alternative. Positive values correspond to correct matches

<i>Novels</i> Character	<i>TV Show</i> Character	Period U2			Period U5		
		named	common	top-20	named	common	top-20
Jeyne Westerling	Talisa Stark	-	-	-	-0.04	-0.21	0.55
Vargo Hoat	Locke	-	-	-	0.01	0.01	0.10
Cleos Frey	Alton Lannister	0.02	0.03	0.12	-0.03	-0.06	0.05
Asha Greyjoy	Yara Greyjoy	0.06	0.21	0.84	0.04	0.19	0.45
Robert Arryn	Robin Arryn	0.02	-0.15	-0.10	0.06	0.09	0.08
Jhogo	Kovarro	-0.13	-0.17	0.48	-0.14	-0.18	0.53
Grazdan mo Eraz	Razdal mo Eraz	-	-	-	-0.15	0.15	0.21
Grazdan mo Ullhor	Greizhen mo Ullhor	-	-	-	-0.04	0.01	0.21
Stalwart Shield	White Rat	-	-	-	0.00	-0.13	0.00

on the other side, the best alternative. Consequently, positive values correspond to correct matches. Some characters only enter the story after the second novel or season, which is why some values are missing. As remarked before, we obtain better performance when focusing on narrower character sets. Note that in the present case, **top-20** means that we compare the two character of interest based only on the relationships with the 20 most important characters. In the case of period U5, it is worth stressing that all matches are correct when using **top-20**.

Table SM9: Similarity difference between, on one side, pairs of characters of interest from the Novels and TV Show, and, on the other side, the best alternative. Positive values correspond to correct matches

<i>Novels</i> Character 1	Character 2	<i>TV Show</i> Character	Period U2			Period U5		
			named	common	top-20	named	common	top-20
Dirk	Clubfoot Karl	Karl	-	-	-	-0.07	-0.05	0.05
Gendry	Edric Storm	Gendry	0.08	0.17	0.12	0.07	0.18	0.21

The second list contains *triples* of characters: two from the novels and one from the TV show. The TV Show character is known to result from the merging of both novels characters. In certain cases, the TV show character bears the same name as one of the novels characters: this means that the TV Show character corresponds mainly to their homonym from the novels, but also includes a significant amount of the remaining character’s characteristics and/or plot. We originally identified 7 such triples, but only the two shown in Table SM9 are usable: some of the concerned characters do not appear at all in our annotations (as already mentioned, these are relatively minor characters). Like before, it appears that the character set has a strong influence on the result, and both cases are successfully matched for period U5 using **top-20**.

SM2.2 Centrality Analysis

This section aims at complementing Section 3.2 from the main article, by providing additional results related to character centrality.

SM2.2.1 Centrality Correlation

Figure 3 from the main article shows only the centrality metric correlations for the **named** and **top-20** character sets. Figure SM9 provides the same view for **common** characters. For all adaptations, we get results very similar to those obtained for the TV show in Figure 3. This is because the TV show annotations contain much fewer unnamed characters, and the plot is much more focused on the core characters present in all three adaptations. Consequently, using only common characters does not change much for this adaptation, whereas it makes both other more similar to the TV show.

Figure SM10 shows the centrality correlation matrices obtained when considering more than two books or seasons: first five books of the novels (left column in the figure), first five seasons of the TV show (centre column), and all eight seasons of the TV Show (right column). For the novels, using five books instead of two only noticeably affect the **top-20** matrix, which exhibits clearer blocks that gather the degree, closeness, and Eigencentrality, whereas the betweenness is isolated. For the first five seasons of the TV show, we observe a higher general correlation level, which increases even more when considering all eight seasons. This could be due to the densification of the network.

SM2.2.2 Centrality Profiles

In [50], Silva *et al.* adopt an approach similar to ours, with a corpus of Portuguese-language literary works, except they only use *unweighted* centrality metrics, and the *k*-means method to perform the cluster analysis. They identify the following clusters, which we note C' for convenience:

- C'_1 : medium betweenness; high Eigenvector, closeness, and degree. These are important figures with a significant impact on the story.
- C'_2 : low betweenness and degree; medium closeness; high Eigenvector. These are characters with a high but local importance.

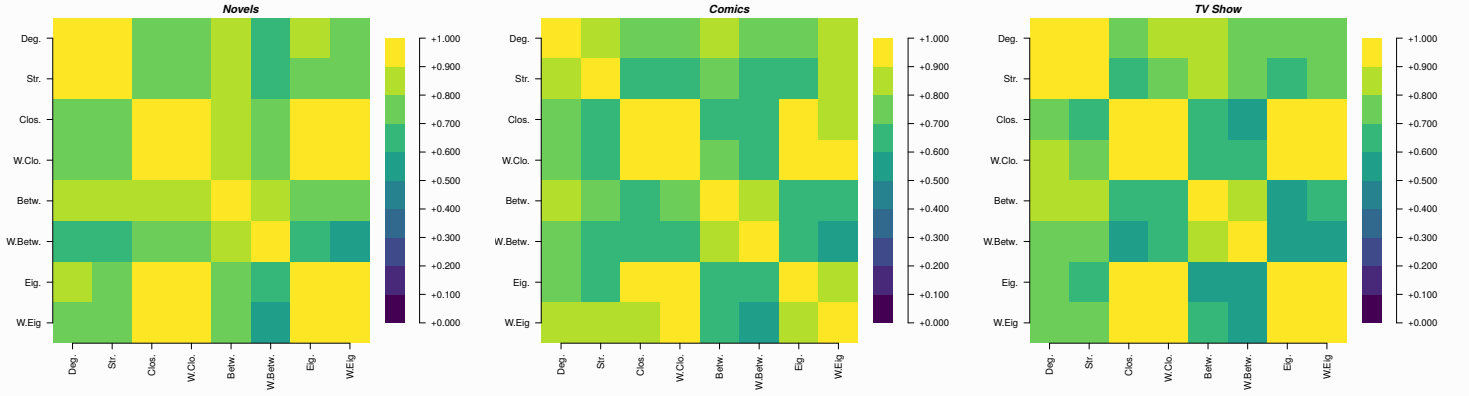


Fig. SM9: Spearman's correlation between the selected centrality metrics, for each of the three adaptations, considering the common character set. By comparison, Figure 3 from the main article shows the matrices for named and top-20 character sets

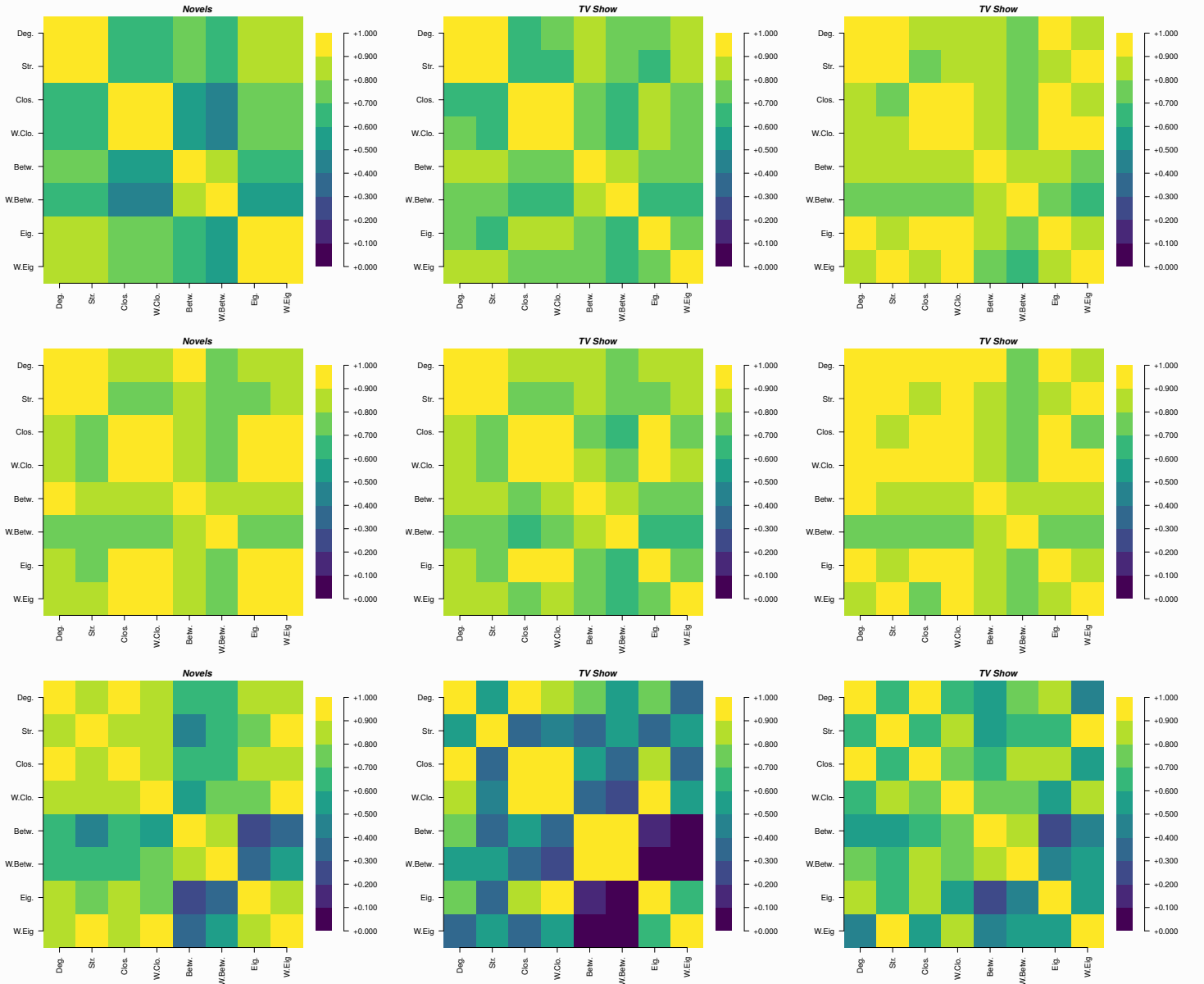


Fig. SM10: Spearman's correlation between the selected centrality metrics, for the first 5 books of the novels (left column), the first 5 seasons of the TV Show (centre column) and all 8 seasons of the TV Show (right column). Rows differ in the considered character set: named (top row), common (centre row), and top-20 (bottom row). By comparison, Figures 3 and SM9 focus on period U_2 , i.e. the first two books, volumes, and seasons

- C'_3 : low betweenness, Eigenvector, and degree; medium closeness. These are secondary characters that act as mediators or connectors between different subplots or social groups.

- C'_4 : low betweenness, Eigenvector, closeness, and degree. These are minor characters.

Table SM10 provides a visual summary of these clusters, as well as the ones identified in Section 3.2.2 of the main article. It is not straightforward to match these clusters, because ours are not as precisely defined in terms of centrality scores: certain metrics cover a range of values, e.g. the degree scores in C_1 (novels) range from low to medium. Still, it appears that, for all three adaptations, the first cluster is similar to C'_4 (minor characters), while the second (third for the TV show) can be matched to C_1 (major characters).

Table SM10: Summary of the classes of centrality identified through clustering for the three adaptations considered in this article and shown in Figure 5, and those studied by Silva *et al.* [50]

Adaptation	Cluster	Degree	Eigenvector	Betweenness	Closeness
<i>Novels</i>	C_1	Low / Medium	Low / Medium	Low	Low / Medium
	C_2	High	High	Medium	High
<i>Comics</i>	C_1	Low / Medium	Low / Medium	Low	Medium
	C_2	High	High	Low / High	High
<i>TV Show</i>	C_1	Low	Low	Low	Low / Medium
	C_2	Medium / High	Medium / High	Low	Medium / High
	C_3	High	High	High	High
Silva <i>et al.</i> [50]	C'_1	High	High	Medium	High
	C'_2	Low	High	Low	Medium
	C'_3	Low	Low	Low	Medium
	C'_4	Low	Low	Low	Low

In order to get more comparable clusters, we select the dendrogram cuts that correspond to $k = 4$ clusters, as shown in Figure SM11. These are not the best cuts, which are discussed in the main article (cf. Figure 5). But they allow a direct comparison between, on the one hand, the three adaptations, and on the other hand, the results of Silva *et al.* [50].

When considering four clusters, the adaptations exhibit relatively similar classes of centrality. Clusters C_1 gather minor characters, with low to medium closeness, and low degree, Eigenvector centrality, and betweenness. These clusters match Silva *et al.*'s C'_3 and C'_4 . Clusters C_2 contain minor characters that are a bit more central, exhibiting higher levels of closeness, degree, and Eigenvector centrality, while their betweenness stays low. This case is not covered in Silva *et al.*'s typology. Clusters C_3 are constituted of major characters with high closeness, degree, and Eigenvector centrality, but low to medium betweenness, which matches Silva *et al.*'s C'_1 . Finally, clusters C_4 contain the few major characters that are central in terms of all four metrics, and not covered in Silva's *et al.* typology.

Table SM11: Summary of the classes of centrality identified through clustering ($k = 4$) for the three adaptations considered in this article and shown in Figure SM11, and those studied by Silva *et al.* [50]

Adaptation	Cluster	Degree	Eigenvector	Betweenness	Closeness
ASoIaF & GoT	C_1	Low	Low	Low	Low / Medium
	C_2	Medium	Medium	Low	High
	C_3	High	High	Medium	High
	C_4	High	High	High	High
Silva <i>et al.</i> [50]	C'_1	High	High	Medium	High
	C'_2	Low	High	Low	Medium
	C'_3	Low	Low	Low	Medium
	C'_4	Low	Low	Low	Low

To be complete, we have to mention the work of Masías *et al.* [51] and Pang *et al.* [52]. They adopt a similar approach when identifying classes of characters by clustering them based on their centrality. However, their results are not comparable to ours, because the methods they used are too different. First, they do not use standard centrality metrics, but rather some parametric weighted versions, which allow handling two types of weights at once. Second, they study the clusters in terms of their stability relative to these parameters. Third, they do not normalise the metric values to account for scale differences. Nevertheless, it is worth mentioning that they typically identify 3 clusters corresponding to 3 increasing levels of centrality. They do not discuss each metric separately, so we assume that these levels concern all metrics indiscriminately.

SM3 Narrative Matching

This section provides additional results for Section 4 of the main article. First, Section SM3.1 highlights combinations of parameters leading to the best matching performance, using both *text-constrained* and *commensurate* narrative units. Then, Section SM3.2 shows the similarity matrices corresponding to the best matching performance for the *text-constrained* units. Section SM3.3 and SM3.4 present results for all possible configurations of structural and hybrid matching. Section SM3.5 discusses the results of narrative matching for the *Novels vs. TV Show* pair over period U2, as opposed to U5 in the main

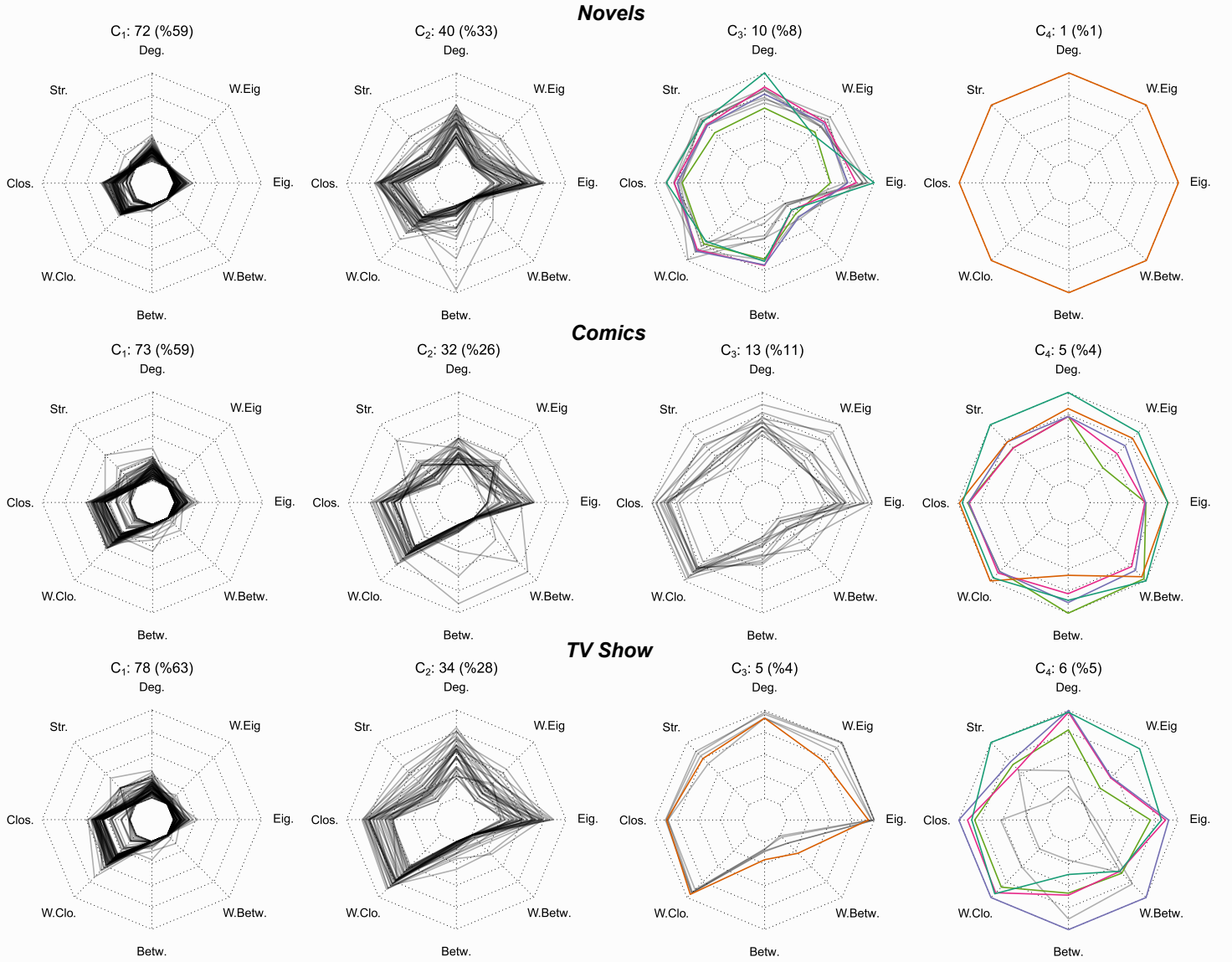


Fig. SM11: Centrality profiles of the common character set, for all three adaptations, when fixing the number of clusters to four. The five most important characters are represented in color, as in Figures 4 and 5 from the main article

article. Finally, Section SM3.6 displays the results of structural matching using cumulative dynamic character networks, whereas we present results with instant networks in the main article.

SM3.1 Best Configurations

The top part of Table SM12 shows the best configurations of narrative matching using text-constrained narrative units, while its bottom part deals with the best configurations using commensurate narrative units. Overall, the best performance is obtained using commensurate units and hybrid similarity.

Table SM12: Best configurations for the task of narrative matching using the *text-constrained* and *commensurate* narrative units

Narr. Unit	Adaptations Pair	Similarity	Repr.	Measure	Character Set	Text Sim.	Alignment	F1
Text-Constr.	<i>Novels vs. Comics</i>	Hybrid	Vertices	Ružička	common	<i>tfidf</i>	Smith–Waterman	67.37
	<i>Novels vs. TV Show</i>	Structural	Edges	Jaccard	common	–	Smith–Waterman	32.63
	<i>Comics vs. TV Show</i>	Structural	Vertices	Ružička	common	–	Smith–Waterman	51.40
Commens.	<i>Novels vs. Comics</i>	Hybrid	Edges	Ružička	named	<i>tfidf</i>	Smith–Waterman	78.50
	<i>Novels vs. TV Show</i>	Hybrid	Vertices	Ružička	common	<i>tfidf</i>	Smith–Waterman	39.04
	<i>Comics vs. TV Show</i>	Hybrid	Edges	Jaccard	top20	<i>sbert</i>	Smith–Waterman	63.87

SM3.2 Similarity Matrices

Figure SM12 shows the similarity matrices \mathbf{S} that lead to the best alignment for each pair of media (except for configurations with blocks), as described in Table SM12. These matrices exhibit patterns where an overall alignment can loosely be observed. These similarity matrices however are not sufficient to align narrative units, as we find that the performance of the simple thresholding alignment baseline is lower than that of the Smith–Waterman algorithm.

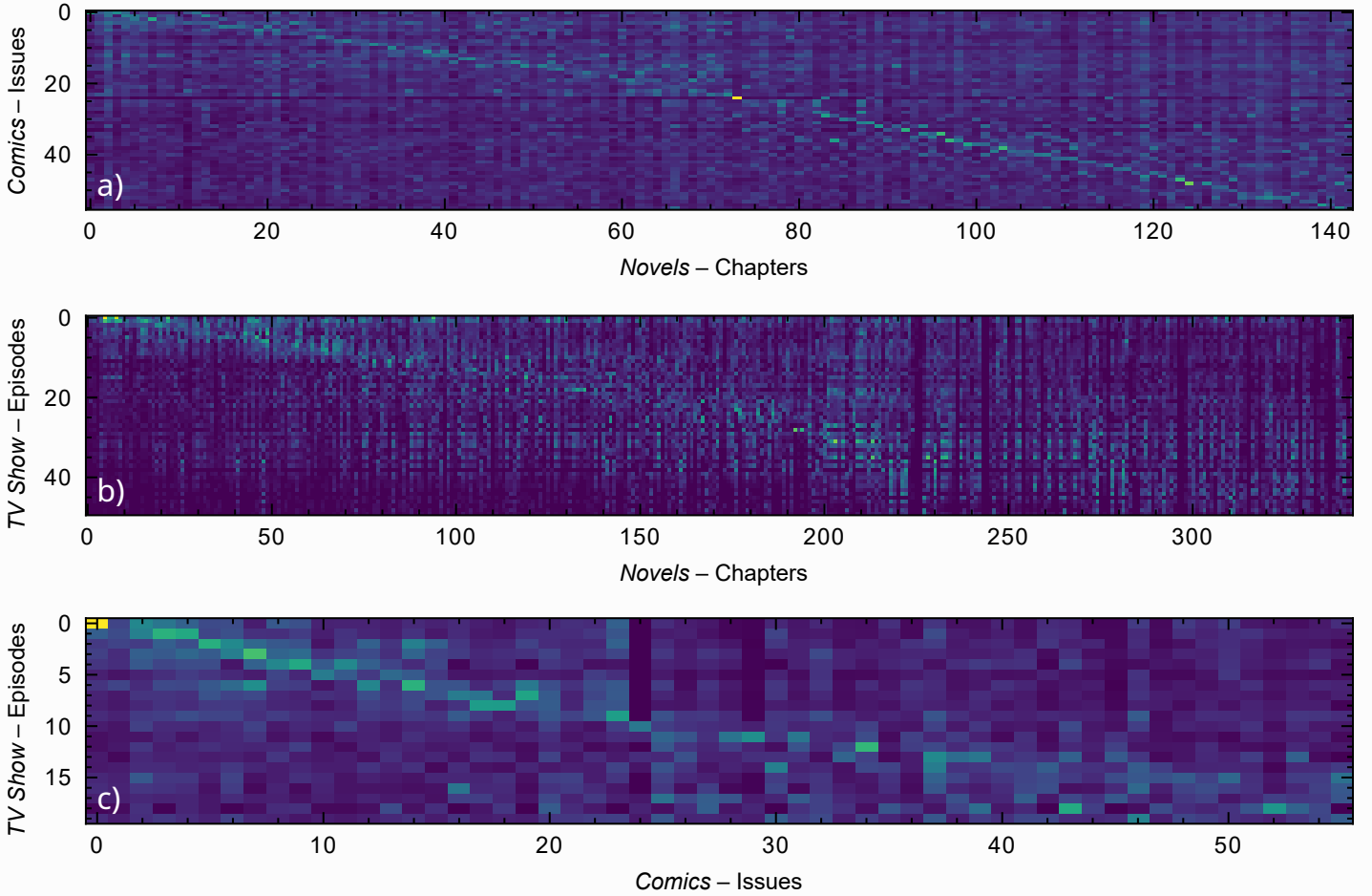


Fig. SM12: Similarity matrices that produce the best alignments for all pairs of media: a) *Novels* vs. *Comics*; b) *Novels* vs. *TV Show*; c) *Comics* vs. *TV Show*. Configurations with commensurate units are excluded, since matrices are of different shapes. See Table SM12 for the precise configurations

SM3.3 Structural Matching

Table SM13 shows the results of structural alignment for all surveyed configurations. By comparison, Table 6 in the main article only focuses on the best results.

We can conclude that the Smith–Waterman algorithm performs better than thresholding. In addition, the best results are obtained with the `common` character set, whereas `top-20` performs the worst. However, it is difficult to reach a conclusion for the other two parameters: compared objects (edges vs. vertices), and weight usage (Jaccard vs. Ruzička).

SM3.4 Hybrid Matching

Table SM14 shows all the results of narrative matching using the *hybrid*-based similarity for the *text-constrained* and *commensurate* narrative units respectively. When using the *commensurate* units, hybrid similarity improves the results compared to structural similarity by between 8.1 and 11.3 F1 depending on the media pair. Overall, we obtain the best overall matching performance using the *commensurate* units to perform hybrid matching.

SM3.5 *Novels* vs. *TV Show* over U2

In this section, we present the results of narrative matching for the *Novels* vs. *TV Show* media pair over the U2 time period. By comparison, the main article focuses on U5. Since the TV show diverges more and more from the novels across seasons, we expect that aligning adaptations over the U2 period is easier than over the U5 period.

Table SM13: Narrative matching performance, expressed in F1-score, for all configurations of *structure*-based matching. The first column indicates the type of narrative unit used: text-constrained (*Text-Constr.*) or commensurate (*Commens.*). The *Repr.* column indicates whether the similarity is computed over vertex or edge sets. The *Measure* column indicates whether the similarity measure ignores weights (*Jaccard*) or take them into account (*Ružička*)

Narrative Units	Repr.	Measure	Alignment	<i>Novels vs. Comics</i>			<i>Novels vs. TV Show</i>			<i>Comics vs. TV Show</i>		
				named	common	top-20	named	common	top-20	named	common	top-20
Text-Constr.	Edges	Jaccard	Thresholding	25.86	28.46	11.66	12.86	13.18	6.59	39.85	44.08	31.28
			Smith–Waterman	54.55	53.85	38.49	28.38	32.63	1.54	47.78	46.93	45.71
	Ružička	Thresholding	29.61	34.74	13.31	20.72	23.43	7.34	43.36	46.95	33.33	
		Smith–Waterman	55.63	57.54	43.84	25.43	24.12	1.75	39.55	49.72	46.86	
	Vertices	Jaccard	Thresholding	13.73	17.61	8.53	11.67	13.72	6.15	28.30	24.31	19.85
			Smith–Waterman	53.71	56.54	26.71	28.72	23.61	8.04	39.77	49.16	39.33
Ružička	Thresholding	15.75	26.57	6.98	13.38	18.73	6.81	35.20	34.74	19.19		
	Smith–Waterman	58.74	62.94	46.81	25.91	23.61	9.87	49.72	51.40	38.42		
Commens.	Edges	Jaccard	Thresholding	24.67	27.95	11.33	17.97	16.91	6.71	37.72	37.86	34.18
			Smith–Waterman	71.52	71.14	49.35	33.17	31.58	22.54	47.62	47.87	46.99
	Ružička	Thresholding	25.66	33.23	14.04	21.02	18.72	7.74	44.34	44.79	38.18	
		Smith–Waterman	71.38	71.57	59.93	31.33	34.38	24.33	47.62	49.46	46.99	
	Vertices	Jaccard	Thresholding	16.79	19.84	8.24	15.67	12.54	6.42	8.39	5.97	7.50
			Smith–Waterman	71.81	70.23	60.93	31.78	32.16	30.69	52.91	52.91	44.32
Ružička	Thresholding	14.80	23.82	7.77	15.62	17.63	6.55	35.35	26.92	7.14		
	Smith–Waterman	70.95	72.30	63.33	34.46	35.09	25.91	51.06	51.06	40.88		

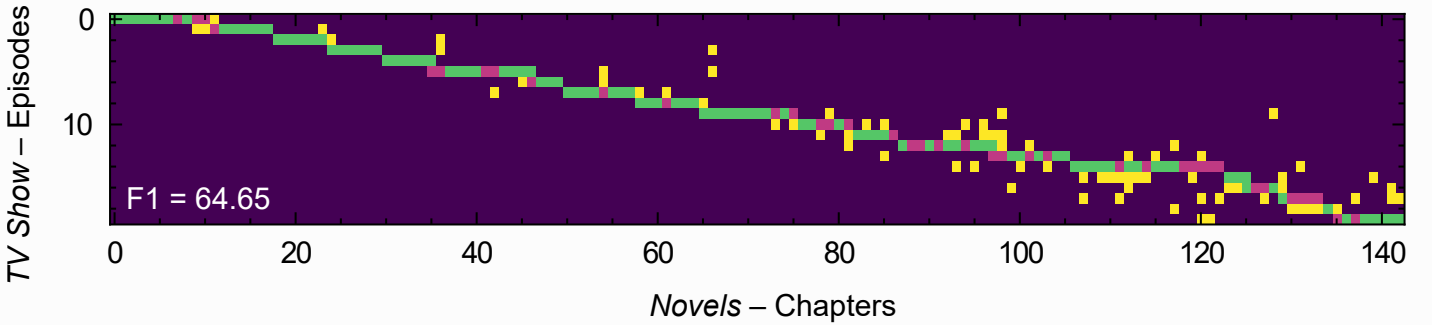


Fig. SM13: Best performing alignment for the *Novels vs. TV Show* pair over time period U2

The best results over all configurations can be found in Table SM15, while the best alignment can be found in Figure SM13. We restrict ourselves to the best results per similarity due to the great number of possible configurations. As we expect, the performance is higher for the U2 than the U5 period, due to the divergence of the later seasons. We also observe that, as when aligning over U5, matching using commensurate units yields a positive performance boost, with a large increase of 17.9 F1. Combining textual and structural matching increases performance (+5.8 F1), but less than taking commensurate into account.

SM3.6 Cumulative Networks

Table SM16 presents results of structural narrative matching using *cumulative* networks on the text-constrained narrative units. By comparison, the main article focuses on *instant* networks.

Since the cumulative network of a narrative unit is the aggregation of previous instant networks, differences between narrative units are less and less noticeable, leading to poor matching performance. The Smith–Waterman algorithm is still able to achieve mild performance in some cases (47.18 F1 on the *Novels vs. Comics* pair), but overall matching with instant networks clearly yields more performance.

Table SM14: Narrative matching performance, expressed in F1-score, for all configurations of *hybrid*-based matching. The first column indicates the type of narrative unit used: text-constrained (*Text-Constr.*) or commensurate (*Commens.*). The *Text Sim.* column indicates the textual similarity measure in usage. The *Struct Repr.* column indicates whether the similarity is computed over vertex or edge sets. The *Struct Measure* column indicates whether the similarity measure ignores weights (*Jaccard*) or take them into account (*Ružička*)

Narrative Unit	Text Sim.	Struct. Repr.	Struct. Measure	Alignment	<i>Novels vs. Comics</i>			<i>Novels vs. TV Show</i>			<i>Comics vs. TV Show</i>		
					named	common	top-20	named	common	top-20	named	common	top-20
Text-Constr.	<i>sbert</i>	Edges	Jaccard	Thresholding	27.00	18.60	6.90	17.33	15.69	7.04	40.13	42.70	24.90
			Smith–Waterman	63.16	63.86	61.75	27.06	27.41	1.53	38.86	45.71	44.07	
		Ružička	Thresholding	11.54	15.95	13.79	22.58	24.64	10.98	45.09	46.75	25.17	
		Smith–Waterman	57.75	60.35	58.95	25.16	26.21	17.80	39.55	40.68	38.42		
		Vertices	Jaccard	Thresholding	25.64	9.09	13.23	11.43	12.00	6.32	32.22	33.99	22.22
			Smith–Waterman	65.25	62.90	61.97	22.11	28.01	12.50	35.29	35.29	32.94	
	Ružička	Thresholding	29.32	29.51	13.78	15.79	20.07	6.32	36.67	46.15	22.73		
	Smith–Waterman	62.46	65.96	57.45	28.95	27.40	21.73	44.07	44.71	32.94			
	<i>tfidf</i>	Edges	Jaccard	Thresholding	27.51	20.93	7.73	19.95	22.57	12.65	34.10	35.40	25.42
			Smith–Waterman	57.24	57.54	58.95	30.24	29.31	9.89	47.78	42.46	42.94	
		Ružička	Thresholding	21.18	33.67	14.96	23.73	26.90	14.02	33.72	39.14	24.97	
		Smith–Waterman	60.28	60.28	58.95	29.10	27.09	17.02	46.93	42.46	42.94		
Vertices		Jaccard	Thresholding	24.31	19.51	14.69	15.06	13.00	12.80	31.00	29.90	24.73	
		Smith–Waterman	61.70	60.99	51.06	21.93	26.95	16.62	42.46	48.04	31.14		
Ružička	Thresholding	38.74	39.60	15.17	20.50	22.60	12.31	33.59	39.29	26.89			
Smith–Waterman	65.26	67.37	47.52	30.61	30.95	16.44	49.16	49.16	31.14				
Commens.	<i>sbert</i>	Edges	Jaccard	Thresholding	29.41	39.06	8.28	18.08	21.40	6.82	44.28	46.24	25.85
			Smith–Waterman	70.71	71.19	70.51	33.50	34.26	30.20	53.68	58.64	63.87	
		Ružička	Thresholding	43.27	45.98	13.26	23.58	22.58	9.87	39.41	43.81	27.05	
		Smith–Waterman	72.11	70.75	66.22	36.86	36.43	30.50	58.33	52.08	60.73		
		Vertices	Jaccard	Thresholding	7.84	11.25	13.14	17.30	14.96	6.13	37.35	33.72	25.23
			Smith–Waterman	69.39	75.93	68.90	32.36	35.86	31.78	59.38	60.42	61.46	
	Ružička	Thresholding	21.05	47.62	12.58	16.49	19.17	6.25	32.36	33.99	25.79		
	Smith–Waterman	72.60	74.75	70.43	34.04	33.63	35.56	59.07	54.74	61.46			
	<i>tfidf</i>	Edges	Jaccard	Thresholding	22.10	31.96	13.26	19.17	15.62	11.71	31.96	31.58	24.56
			Smith–Waterman	72.97	74.07	75.84	32.58	35.79	33.29	59.69	58.33	52.08	
		Ružička	Thresholding	27.32	45.91	12.57	22.39	21.37	12.46	32.34	33.28	24.30	
		Smith–Waterman	78.50	74.50	74.92	35.00	38.30	33.62	58.03	58.03	56.54		
Vertices		Jaccard	Thresholding	17.05	20.77	12.22	20.69	15.25	12.75	32.38	28.07	22.99	
		Smith–Waterman	71.86	74.92	70.00	34.30	36.27	29.97	55.96	59.69	46.07		
Ružička	Thresholding	35.24	46.84	13.33	16.91	21.08	11.92	29.58	31.18	23.37			
Smith–Waterman	75.00	74.75	72.48	36.18	39.04	31.27	55.96	58.33	46.07				

Table SM15: Performance obtained on the U2 time period of the *Novels vs. TV Show* pair, expressed in terms of F1-score. Only the best results across configurations are shown. The mentions *Text-Constrained* and *Commensurate* refer to the corresponding narrative units, see Section 4.3 in the main article.

Alignment	Textual Matching		Structural Matching		Hybrid Matching	
	Text-Constrained	Commensurate	Text-Constrained	Commensurate	Text-Constrained	Commensurate
Thresholding	29.01	37.24	32.56	37.95	41.18	
Smith–Waterman	36.65	45.00	62.87	50.78	64.65	

Table SM16: Performance obtained when using *structural*-based representations using *cumulative networks* and the *text-constrained* narrative units to tackle the narrative matching task, expressed in terms of F1-score. Columns are organised as in previous tables.

Representation	Measure	Alignment	<i>Novels vs. Comics</i>			<i>Novels vs. TV Show</i>			<i>Comics vs. TV Show</i>		
			named	common	top-20	named	common	top-20	named	common	top-20
Edges	Jaccard	Thresholding	4.05	3.94	3.61	6.36	5.12	4.89	19.98	20.20	19.95
		Smith–Waterman	16.85	41.40	10.79	2.00	22.34	0.00	19.88	29.71	21.05
Vertices	Jaccard	Thresholding	3.80	3.98	3.62	5.66	5.17	4.90	20.21	20.41	20.22
		Smith–Waterman	22.14	47.18	11.72	7.75	13.11	1.27	18.29	26.59	21.05
Edges	Jaccard	Thresholding	4.10	3.90	3.53	4.82	5.18	4.89	22.24	20.78	19.79
		Smith–Waterman	12.69	40.43	1.01	1.19	25.63	0.42	20.25	33.33	7.74
Vertices	Jaccard	Thresholding	3.76	3.88	3.53	7.72	5.19	4.89	20.81	20.23	19.95
		Smith–Waterman	17.02	32.62	1.01	2.07	24.97	0.42	22.50	35.63	7.74