Detecting Collaboration Profiles in Success-based Music Genre Networks

Supplementary Material

1 Data Processing and Network Characterization

When processing Spotify chart data, we make some decisions regarding the definition of success of a genre collaboration and the genre reduction through our mapping:

- In our chart analyses, we consider the number of streams as the success measure for a hit song. Therefore, in our temporal and regional analyses, we define the success of a genre collaboration (i.e. the edge weight) as the average value of total streams of all songs involving those genres within the considered market and period.
- In the mapping proccess from the Spotify-assigned genres to our *super-genres*, we detect 76 out of 896 genres which do not fit into any category. For example, *talent show*, in which artists may belong to other well-established genres (e.g. *pop*, *country* or *hip hop*). Thus, these genres are categorized as *other*. As Spotify artists can be assigned to more than one genre, this categorization do not prejudice our further analyses.

Table 1 presents the main information and preliminary statistics on our dataset. In addition, the complete global and regional chart overview and the full characterization of the networks are presented by Tables 2 and 3, respectively. To compute the network metrics, we use *NetworkX*[1], a network analysis Python package.

Table 1: Dataset main statistics.

Years	3
Weeks	156
Markets	9
Charts	1,330
Hit Songs	13,380
Artists	3,612
Genres (before mapping)	896
Genres (after mapping)	162

[1]NetworkX: `https://networkx.github.io/`

Table 2: Most popular music genres in each considered market in the years 2017, 2018 and 2019.

	2017			2018			2019		
	Genre	**Songs**	**Arts.**	**Genre**	**Songs**	**Arts.**	**Genre**	**Songs**	**Arts.**
Global	pop	635	252	pop	701	257	pop	678	256
	hip hop	362	101	rap	587	134	hip hop	432	180
	dance pop	346	121	hip hop	546	181	rap	412	123
	rap	344	86	pop rap	398	93	trap	319	103
	pop rap	286	81	trap	354	96	dance pop	288	94
Australia	pop	635	266	pop	718	262	pop	684	262
	dance pop	360	138	rap	427	106	rap	325	111
	hip hop	300	101	dance pop	358	116	dance pop	295	102
	rap	280	85	hip hop	358	113	hip hop	275	121
	pop rap	236	84	pop rap	300	86	pop rap	233	87
Brazil	pop	447	158	pop	468	166	pop	358	129
	dance pop	212	76	sertanejo	283	63	brazilian funk	280	114
	sertanejo	178	40	brazilian funk	276	111	sertanejo	265	66
	brazilian funk	170	67	dance pop	149	66	dance pop	103	39
	electro	142	54	electro	140	66	electro	101	41
Canada	pop	703	254	rap	850	151	pop	667	239
	rap	559	113	hip hop	732	152	rap	584	141
	hip hop	510	115	pop	715	245	hip hop	455	139
	pop rap	468	95	pop rap	628	111	pop rap	413	107
	trap	372	83	trap	534	107	trap	370	101
France	pop	1,000	271	pop	1,299	287	pop	1,176	276
	hip hop	770	135	hip hop	1,180	191	hip hop	984	177
	rap	728	125	rap	1,120	166	rap	899	153
	francoton	414	52	francoton	434	59	francoton	366	56
	dance pop	174	86	dance pop	162	73	dance pop	119	61
Germany	hip hop	796	171	hip hop	971	216	hip hop	1,048	238
	pop	621	290	pop	687	293	pop	635	281
	rap	372	105	rap	536	129	rap	429	134
	dance pop	299	126	dance pop	282	114	dance pop	210	86
	pop rap	177	72	pop rap	214	74	trap	166	71
Japan	pop	197	119	pop	417	152	j-pop	489	108
	j-pop	138	65	j-pop	387	125	pop	287	125
	dance pop	131	72	dance pop	259	80	j-rock	183	42
	r&b	89	43	r&b	165	63	dance pop	173	57
	rap	63	36	j-rock	164	55	other	125	43
UK	pop	682	246	pop	763	243	pop	665	234
	dance pop	383	131	hip hop	490	161	hip hop	441	152
	hip hop	355	109	rap	480	122	rap	360	105
	rap	276	83	dance pop	424	137	dance pop	296	107
	pop rap	224	78	pop rap	319	90	pop rap	209	72
USA	rap	673	122	rap	939	159	rap	715	165
	pop	650	210	hip hop	783	159	pop	635	203
	hip hop	594	124	pop rap	653	107	hip hop	548	156
	pop rap	540	93	pop	642	201	pop rap	492	110
	trap	444	91	trap	611	116	trap	488	121

Table 3: Network characterization for all global and regional markets, grouped according to their similar network evolution. Underlined values are the highest metric value for a specific market throughout the considered period.

Metric	***Global***			***Group 1: USA & Canada***						***Group 3: Other English-speaking markets***					
				USA			*Canada*			*UK*			*Australia*		
	2017	**2018**	**2019**	**2017**	**2018**	**2019**	**2017**	**2018**	**2019**	**2017**	**2018**	**2019**	**2017**	**2018**	**2019**
G	72	79	89	76	73	83	70	71	82	74	76	79	65	71	79
C	564	583	709	542	522	670	540	558	680	610	605	627	512	514	577
AD	15.7	14.8	15.9	14.3	14.3	16.1	15.4	15.7	16.6	16.5	15.9	15.9	15.8	14.5	14.6
AWD	256.9	247.4	236.7	324.6	287.9	241.4	366.3	307.6	212.4	216.5	203.6	159.5	220.6	170.8	140.0
D	0.221	0.189	0.181	0.190	0.199	0.197	0.224	0.225	0.205	0.226	0.212	0.204	0.246	0.200	0.200
ACC	0.743	0.757	0.754	0.762	0.760	0.726	0.739	0.749	0.762	0.724	0.754	0.738	0.718	0.700	0.700
SL	24	21	28	25	22	27	22	23	31	28	25	30	22	23	25
IntraG	4.26%	3.60%	3.95%	4.61%	4.21%	4.03%	4.07%	4.12%	4.56%	4.59%	4.13%	4.78%	4.30%	4.47%	4.33%
InterG	95.74%	96.40%	96.05%	95.39%	95.79%	95.97%	95.93%	95.88%	95.44%	95.41%	95.87%	95.22%	95.70%	95.53%	95.67%

Metric	***Global***			***Group 2: Non-English speaking markets***											
				Brazil			*France*			*Germany*			*Japan*		
	2017	**2018**	**2019**	**2017**	**2018**	**2019**	**2017**	**2018**	**2019**	**2017**	**2018**	**2019**	**2017***	**2018**	**2019**
G	72	79	89	58	63	61	63	63	66	69	75	73	56	71	63
C	564	583	709	453	524	392	465	464	434	555	590	523	350	491	418
AD	15.7	14.8	15.9	15.6	16.6	12.9	14.8	14.7	13.2	16.1	15.7	14.3	12.5	13.8	13.3
AWD	256.9	247.4	236.7	136.1	133.0	95.3	185.1	213.2	153.2	213.8	196.6	152.2	84.3	121.7	68.3
D	0.221	0.189	0.181	0.274	0.268	0.214	0.238	0.238	0.202	0.237	0.200	0.200	0.227	0.198	0.214
ACC	0.743	0.757	0.754	0.770	0.758	0.677	0.778	0.772	0.773	0.759	0.800	0.700	0.748	0.765	0.697
SL	24	21	28	24	29	27	20	22	24	23	24	23	20	24	19
IntraG	4.26%	3.60%	3.95%	5.30%	5.53%	6.89%	4.30%	4.74%	5.53%	4.14%	4.07%	4.40%	5.71%	4.89%	4.55%
InterG	95.74%	96.40%	96.05%	94.70%	94.47%	93.11%	95.70%	95.26%	94.47%	95.86%	95.93%	95.60%	94.29%	95.11%	95.45%

G: number of genres (nodes). **C**: number of genre collaborations (edges). **AD**: average node degree. **AWD**: average node degree (weighted). **D**: network density. **ACC**: average clustering coefficient. **SL**: number of self-loops. **IntraG**: fraction of intra-genre collaborations. **InterG**: fraction of inter-genre collaborations.

***** As Spotify provides Japanese weekly charts only after 08/31/2017, we build Japan's 2017 genre network with data from then.

2 Exploratory Factor Analysis in R

To perform an exploratory factor analysis (EFA), we use the *psych* R package (Revelle, 2017) with the `fa()` function. Our data consist of 27 music genre networks (i.e., nine music markets), containing six different topological metrics described as follows. For a given node v in the networks, let $\mathcal{N}(v)$ be its set of neighbors.

Preferential Attachment (PA). The probability of a given pair of nodes connecting in the future. The intuition behind this index is that, if a node has a high degree, it attracts more neighbors. Therefore, when analyzing two nodes, the more neighbors they have, the more likely they are to connect in the future. Its value is given by Equation 1.

$$PA(u, v) = |\mathcal{N}(u)||\mathcal{N}(v)| \tag{1}$$

Common Neighbors (CN). The number of neighbors that a given pair of nodes have in common in a network, i.e. the intersection of their neighbor set, as formalized by Equation 2.

$$CN(u, v) = |\mathcal{N}(u) \cap \mathcal{N}(v)| \tag{2}$$

Neighborhood Overlap (NO). The ratio between the common neighbors of a given pair of nodes and the union set of their neighbors. Edges with low NO reveal local bridges in the network, i.e. nodes traveling in "social circles", having almost no common connection. Furthermore, the removal of such an edge may completely disconnect the graph (if $NO = 0$) or difficult the access to other network components ($NO > 0$). We calculate such a metric following Equation 3.

$$NO(u, v) = \frac{|\mathcal{N}(u) \cap \mathcal{N}(v)|}{|\mathcal{N}(u) \cup \mathcal{N}(v) - \{u, v\}|} \tag{3}$$

Edge Betweenness (EB). The fraction of shortest paths that go through an edge in the network. Edges with a high score represent a bridge-like connector between two parts of the network, and their removal may affect the communication between others due to the lost common shortest paths. We then calculate the betweenness centrality c_B of an edge $e = (u, v)$ through Equation 4.

$$c_B(e) = \sum_{s,t \in V} \frac{\sigma(s, t|e)}{\sigma(s, t)} \tag{4}$$

where V is the set of nodes within the network, $\sigma(s, t)$ is the number of shortest (s, t)-paths, and $\sigma(s, t|e)$ is the number of those paths passing through edge e (Brandes, 2008).

Resource Allocation (RA). For a pair (u, v) of nodes, it represents the fraction of a resource (e.g. information) that a node can send to another through its common neighbors, as given by Equation 5. If both nodes have a large number of common neighbors, they Resource

Allocation Index tends to be high. Such an index is even higher if their neighbors have a low degree, as the resource is more likely to travel from u to v.

$$RA(u,v) = \sum_{w \in \mathcal{N}(u) \cap \mathcal{N}(v)} \frac{1}{|\mathcal{N}(w)|} \tag{5}$$

Weight (W). In our network model, edge weight is given by the number of collaborations between two music genres, which is statistically correlated with the possibility that these two genres will collaborate in the future.

2.1 Choosing the Number of Factors

Before conducting the EFA, we must determine an acceptable number of factors. The *psych* package offers a few ways in which the number of factors can be decided. Here, we use the `fa.parallel()` function to obtain the suggested number of factors via the Parallel Analysis (Humphreys and Montanelli Jr, 1975) criteria. The output gives us the textual output of the suggested number of factors and a *scree* plot (Cattell, 1966) of the successive eigenvalues. Figures 1, 2 and 3 show the resulting *scree* plot for each network.

In the *scree* plots generated, blue and red lines show eigenvalues of actual and simulated/resampled data (placed on top of each other), respectively. The number of factors is determined by looking at the large drops in the actual data and spot the point where it levels off to the right. Moreover, we must identify the inflection point where the gap between simulated and actual data tends to be minimum. Analyzing all 27 *scree* plots, we can see that the vast majority suggest a number of factors equal to 3.

2.2 Factor Analysis

Once the number of suggested factors is determined using `fa.parallel()`, EFA can be performed using the `fa()` function. In order to enable reproducibility, we provide the model parameters settings, as summarized in Table 4. Specifically, we use the well known Ordinary Least Squares (OLS) factoring method and an oblique rotation, allowing factors to correlate with each other. We can visualize the EFA results using the `fa.diagram()` function, where it takes a `fa()` result object and creates a path diagram showing actor loadings, ordered from strongest to weakest. Factor loadings represent the correlation between each metric and the underlying factor, and they can range from -1 to 1. Figures 4, 5 and 6 show the resulting factor loadings graph for each network.

Table 4: Parameter Settings for Exploratory Factor Analysis

Parameters	**Description**	**Value**
`nfactors`	Number of factors to extract	3
`rotate`	Type of rotation	*oblimin*
`fm`	Factoring method	*ols*

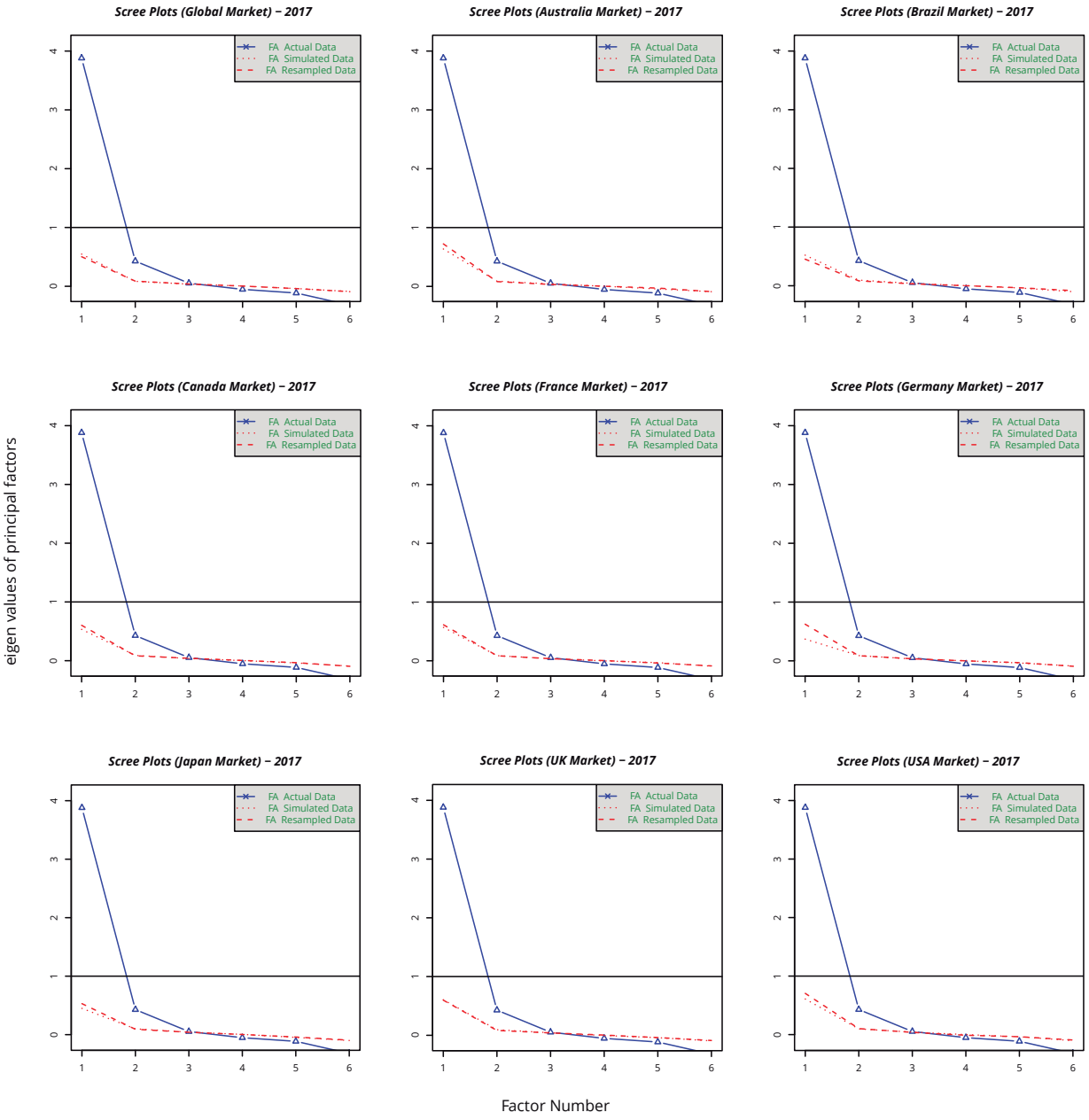

Figure 1: Scree plots resulting from the Parallel Analysis for each genre network in 2017. Blue and red lines show eigenvalues of actual and simulated/resampled data, respectively. The suggested number of factors can be found in the X-axis position right before the "elbow" in the actual data curve.

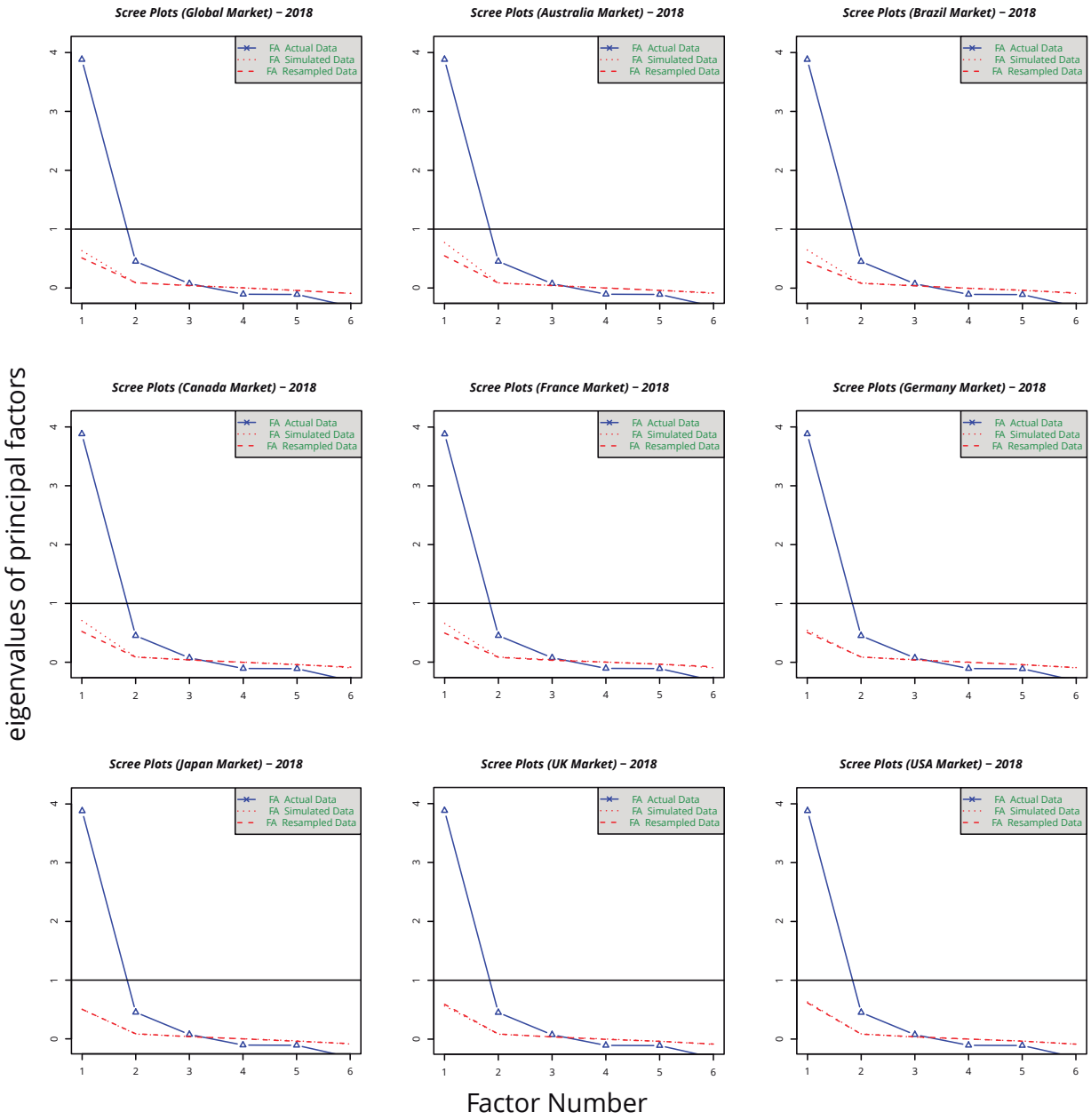

Figure 2: Scree plots resulting from the Parallel Analysis for each genre network in 2018. Blue and red lines show eigenvalues of actual and simulated/resampled data, respectively. The suggested number of factors can be found in the X-axis position right before the "elbow" in the actual data curve.

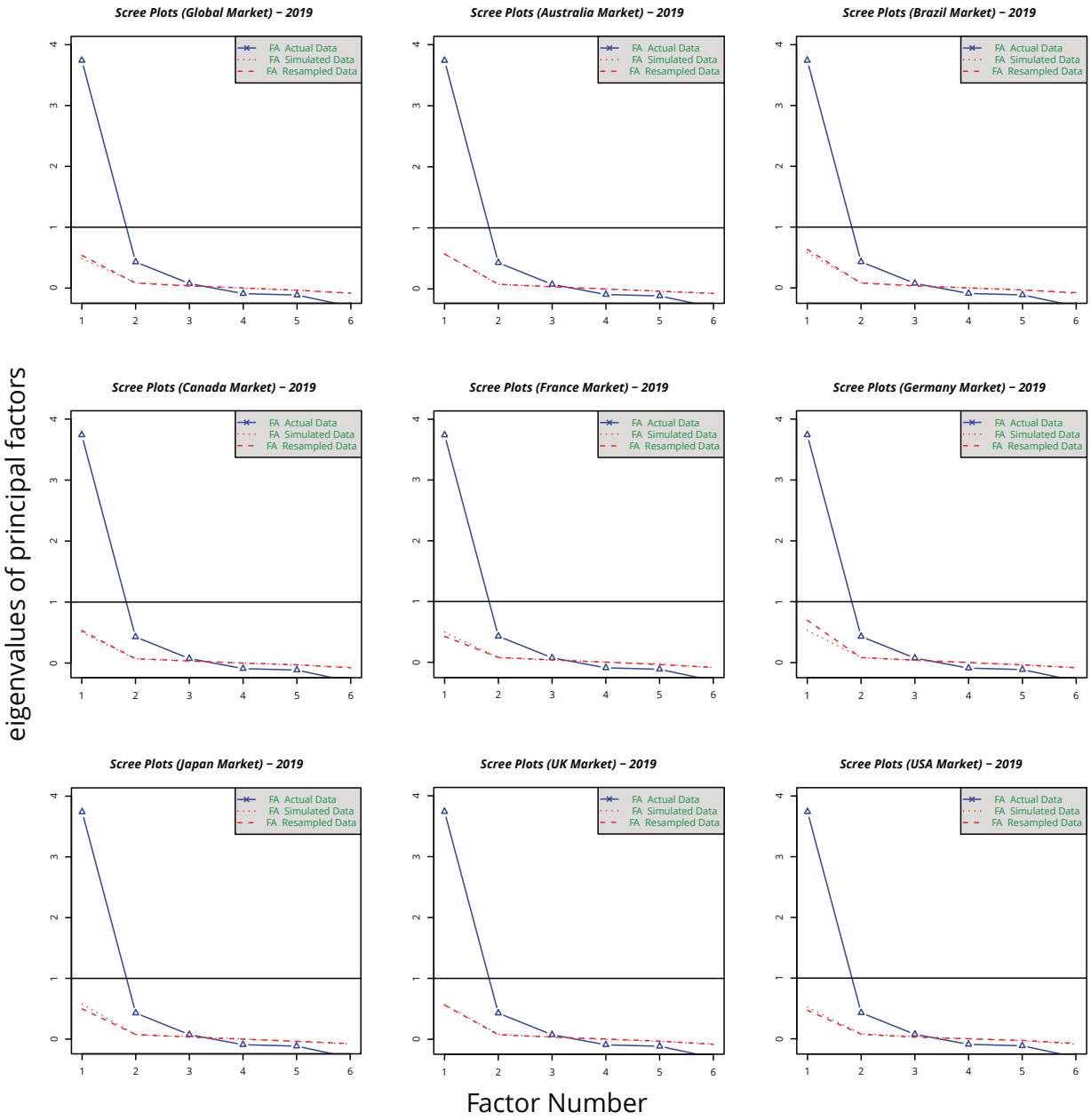

Figure 3: Scree plots resulting from the Parallel Analysis for each genre network in 2019. Blue and red lines show eigenvalues of actual and simulated/resampled data, respectively. The suggested number of factors can be found in the X-axis position right before the "elbow" in the actual data curve.

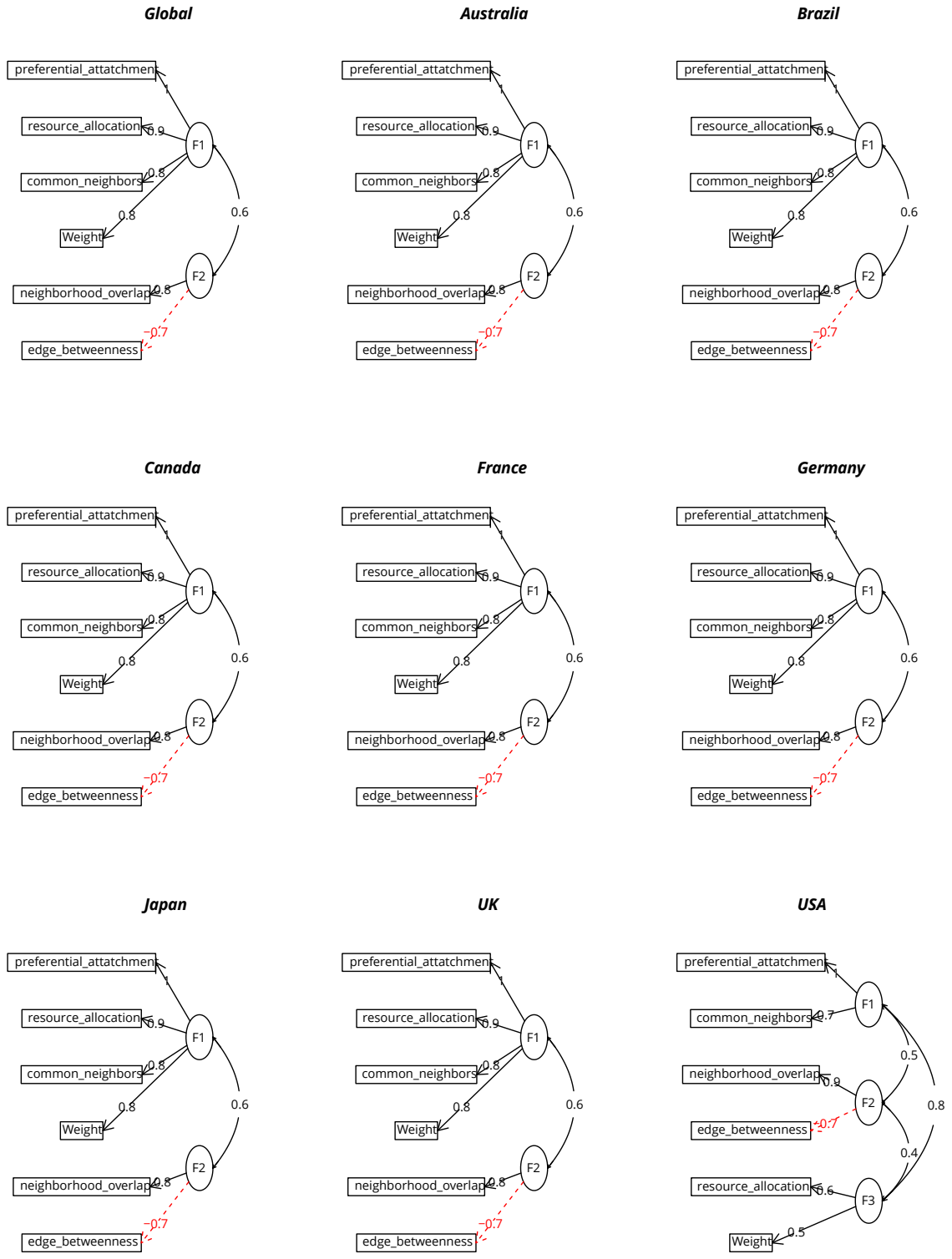

Figure 4: Exploratory Factor Analysis diagram for each genre network in 2017. Solid and dashed lines represent positive and negative correlations, respectively.

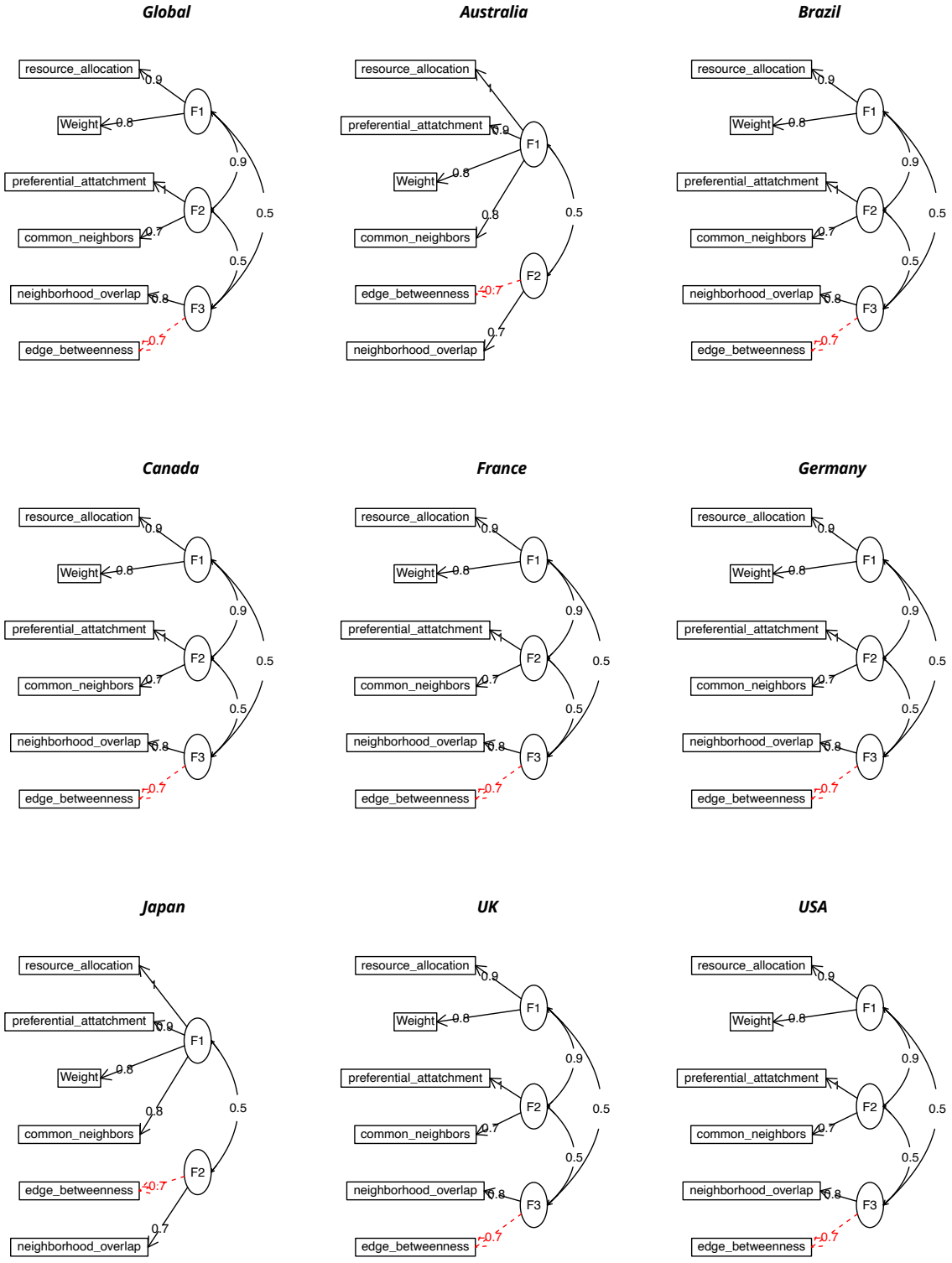

Figure 5: Exploratory Factor Analysis diagram for each genre network in 2018. Solid and dashed lines represent positive and negative correlations, respectively.

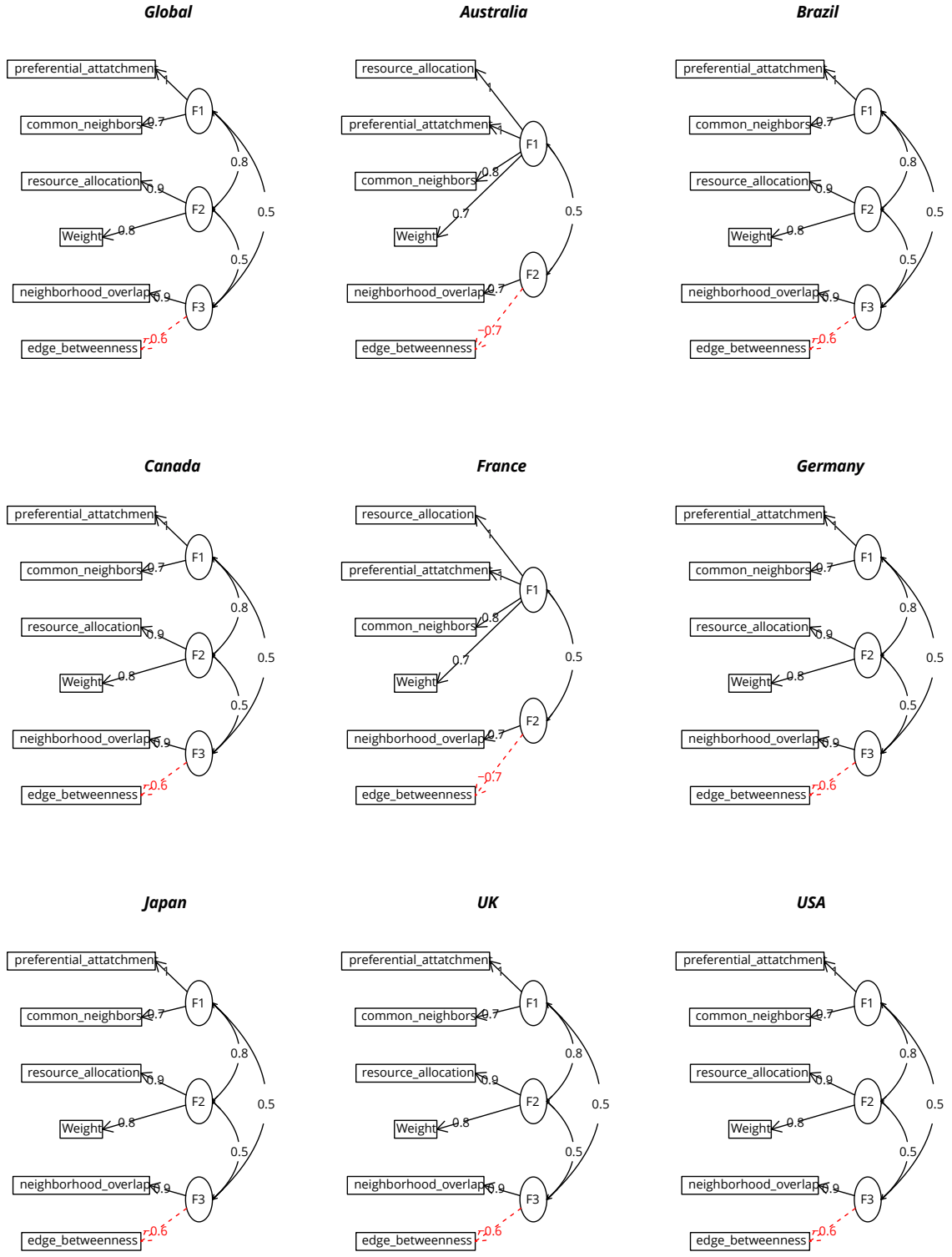

Figure 6: Exploratory Factor Analysis diagram for each genre network in 2019. Solid and dashed lines represent positive and negative correlations, respectively.

3 Cluster Analysis - DBSCAN

We use the DBSCAN algorithm (Ester et al., 1996), which is a classical density-based clustering procedure. DBSCAN clusters the data points by separating areas of high density from areas of low density. It can be used not only to identify clusters of any shape, but also detect noise and outliers in the dataset. Two important parameters are required for DBSCAN:

1. ϵ: It defines the radius of neighborhood around a data point x. It is called as ϵ-neighborhood of x. Such parameter is crucial to choose appropriately. If the ϵ value is chosen too small then large part of the data will be considered as outliers. Otherwise, the clusters will merge and majority of the data points will be in the same clusters.

2. $MinPts$: Minimum number of neighbors (data points) required to form a dense cluster, within ϵ radius. As a general rule, the $MinPts$ can be derived from the number of dimensions D in the dataset as $MinPts \geq D + 1$. Also, the minimum value of $MinPts$ is at least 3.

Here, we use the *dbscan* R package (Hahsler et al., 2019) with the `dbscan()` function. As our dataset has six distinct dimensions (i.e. metrics), we set $MinPts = 7$. Then, to choose the optimal ϵ value, *dbscan* relies on a space-partitioning data structure called a *k-d trees*. This data structure allows us to identify the kNN or all neighbors within a fixed radius ϵ. We now execute the function `kNNdistplot()` with $k = 7$ (must be equal to $MinPts$) to plot the k-distances, which are the average distance of a point to its k-nearest neighbors. Finally, we set ϵ as the k value where where a sharp change take place in the curve. Figures 7, 8 and 9 present such plots for all markets in 2017, 2018 and 2019, respectively. In all markets and years, we observe that this threshold occurs close to $k = 1$, thus being chosen as our ϵ value. The resulting clusters for each market throughout the years are shown by Figures 10, 11 and 12, while the resulting collaboration profiles are presented by Figures 13 and 14.

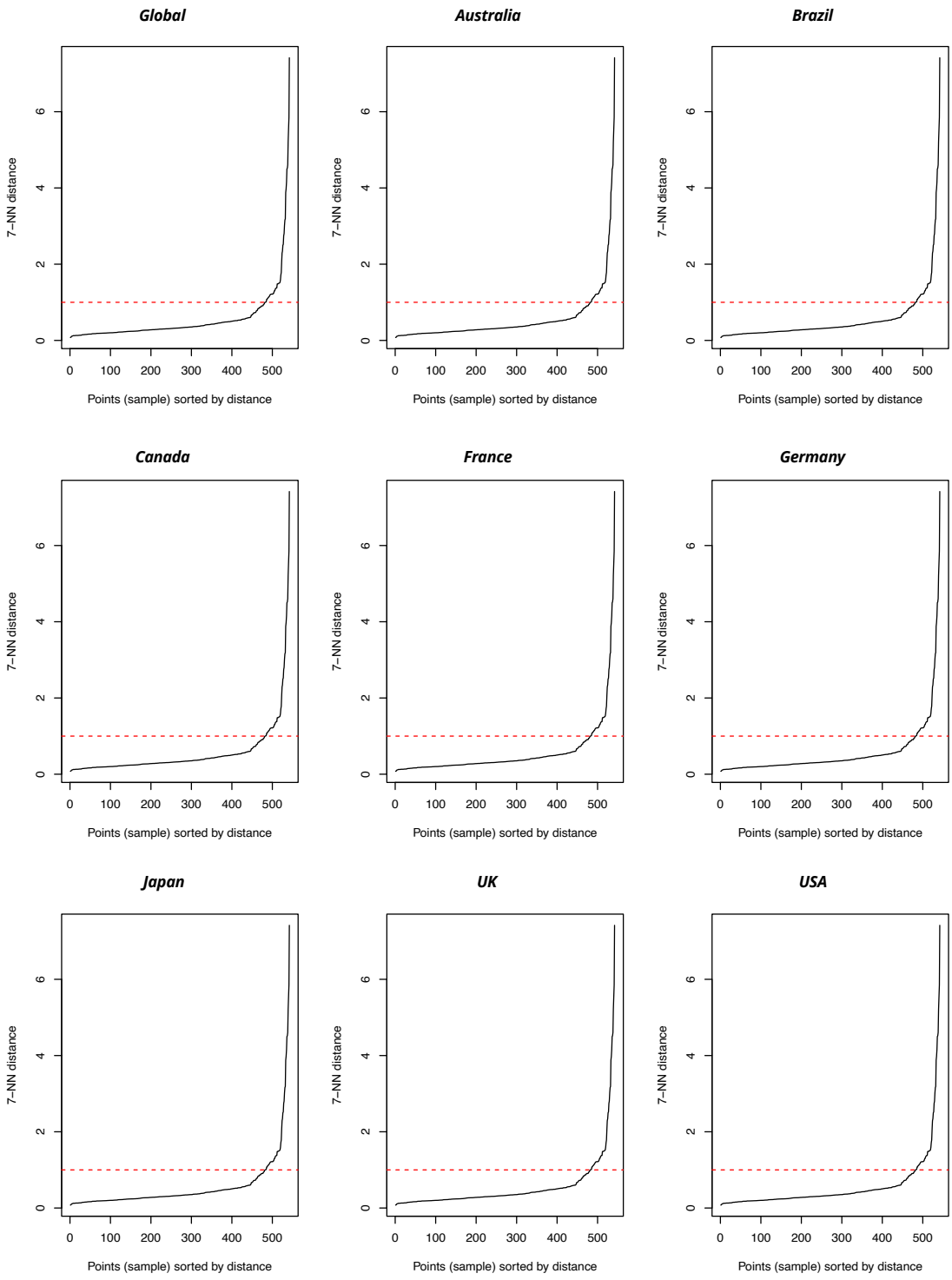

Figure 7: 7-NN distance plot for each genre network in 2017. Dashed lines represent the threshold where a major change occurs in the curve, chosen as the ϵ parameter of DBSCAN.

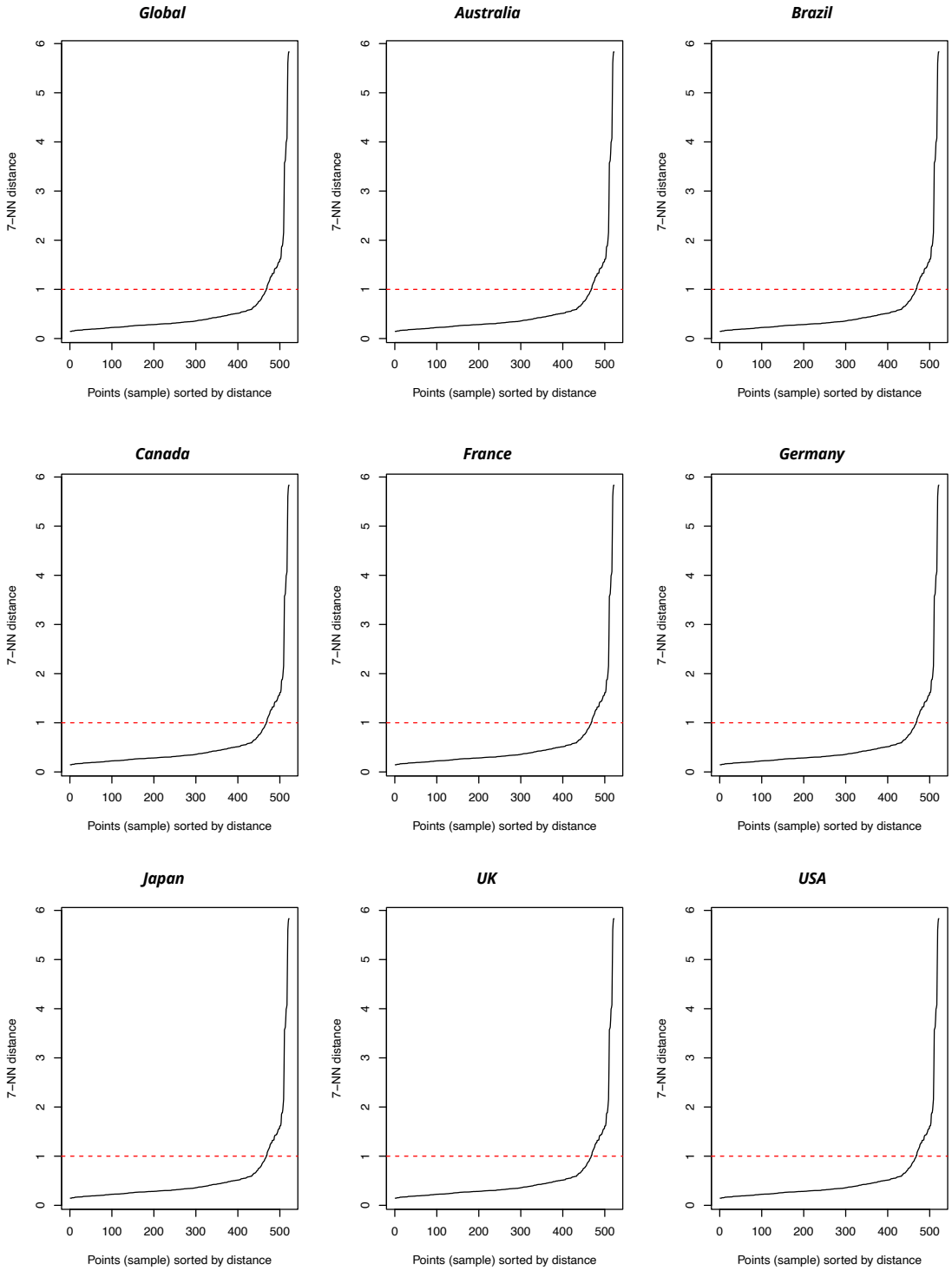

Figure 8: 7-NN distance plot for each genre network in 2018. Dashed lines represent the threshold where a major change occurs in the curve, chosen as the ϵ parameter of DBSCAN.

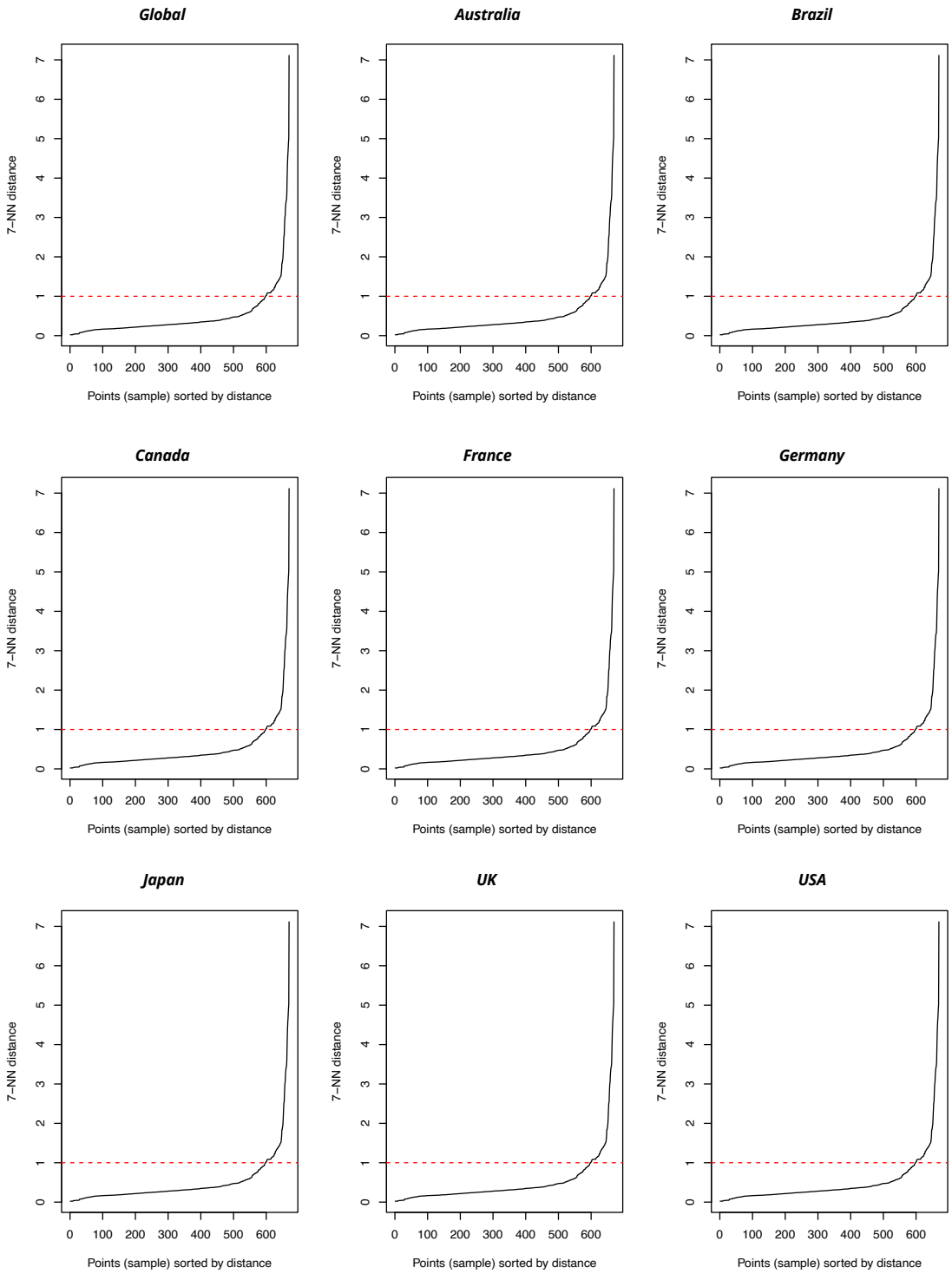

Figure 9: 7-NN distance plot for each genre network in 2019. Dashed lines represent the threshold where a major change occurs in the curve, chosen as the ϵ parameter of DBSCAN.

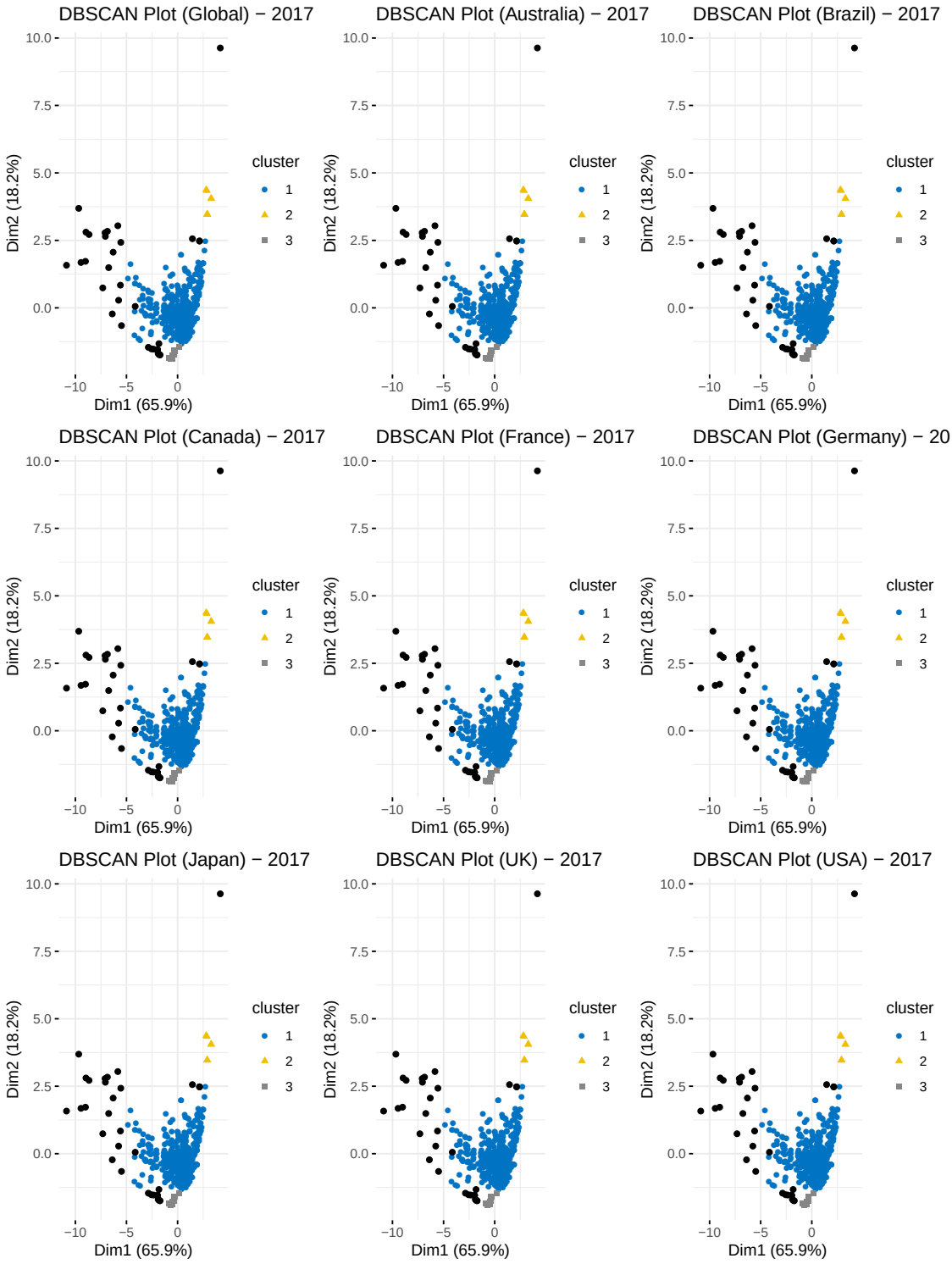

Figure 10: Clustering of genre collaboration profiles in 2017. The results are generated with DBSCAN algorithm with $MinPts = 7$ and $\epsilon = 1.0$. The clustering is based on the topological metrics. Black points correspond to outliers.

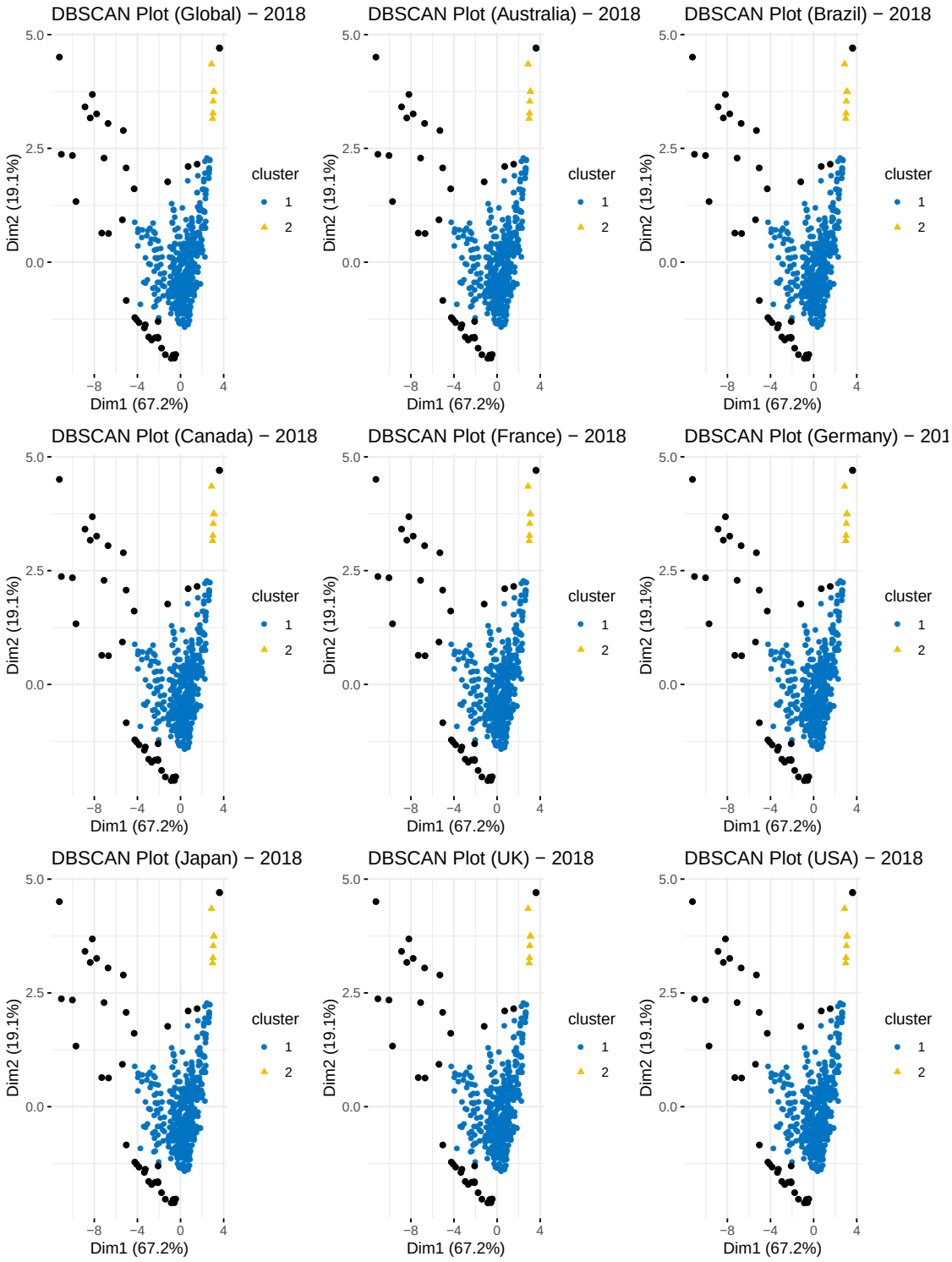

Figure 11: Clustering of genre collaboration profiles in 2018. The results are generated with DBSCAN algorithm with $MinPts = 7$ and $\epsilon = 1.0$. The clustering is based on the topological metrics. Black points correspond to outliers.

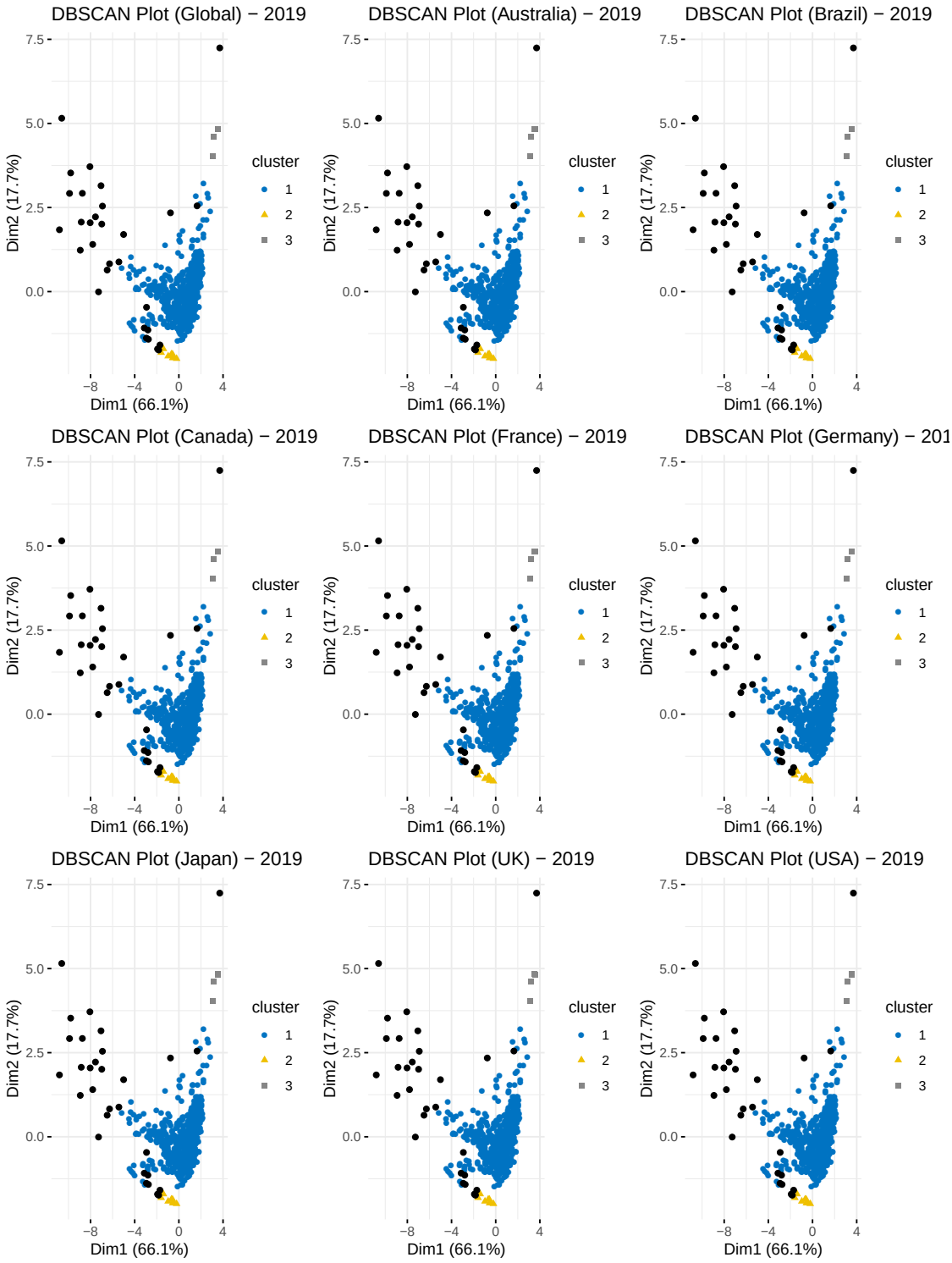

Figure 12: Clustering of genre collaboration profiles in 2019. The results are generated with DBSCAN algorithm with $MinPts = 7$ and $\epsilon = 1.0$. The clustering is based on the topological metrics. Black points correspond to outliers.

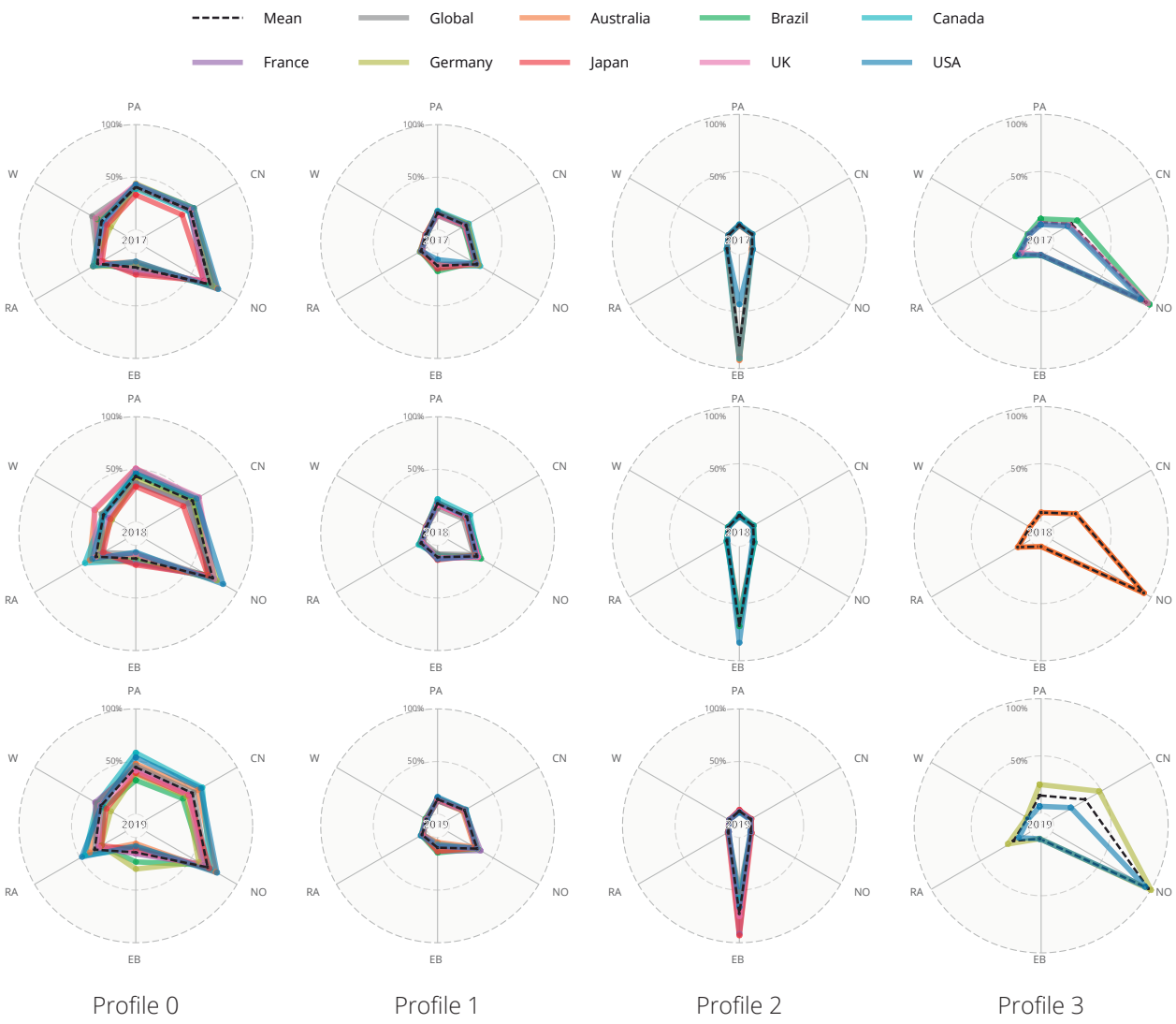

Figure 13: Radar Plots of each genre collaboration profile, divided by year.

Figure 14: Collaboration profiles for all markets (2017-19).

References

Ulrik Brandes. 2008. On variants of shortest-path betweenness centrality and their generic computation. *Soc. Networks* 30, 2 (2008), 136–145.

Raymond B Cattell. 1966. The scree test for the number of factors. *Multivariate behavioral research* 1, 2 (1966), 245–276.

Martin Ester et al. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Procs. of KDD*. 226–231.

Michael Hahsler, Matthew Piekenbrock, and Derek Doran. 2019. dbscan: Fast density-based clustering with r. *Journal of Statistical Software* 25 (2019), 409–416.

Lloyd G Humphreys and Richard G Montanelli Jr. 1975. An investigation of the parallel analysis criterion for determining the number of common factors. *Multivariate Behavioral Research* 10, 2 (1975), 193–205.

William R Revelle. 2017. psych: Procedures for personality and psychological research. (2017).