

- Master Géomatique SIGMA -
Atelier géomatique – rapport d'étude

Géovisualisation des mobilités épiscopales européennes au début du XIV^{ème} siècle



Sommaire

Introduction et présentation des objectifs	3
Organisation du projet et gestion des tâches	4
I. Mise en place de la base de données.....	5
A. Données spécifiques à la mobilité épiscopale.....	5
B. Intégration de données complémentaires à des fins d'analyse.....	6
1. Limites des Etats souverains.....	6
2. Altitude et proximité au réseau routier des diocèses	7
II. Analyse et représentation de la mobilité épiscopale	7
A. Identification des déplacements des évêques	8
B. Impact de l'accessibilité des diocèses sur les déplacements épiscopaux	9
C. Visualisation des déplacements épiscopaux de manière spatialisée et non-spatialisée : utilisation du logiciel Gephi	11
D. Etude des trajets effectués par une personne	12
E. Réalisation de la carte des taux de transferts intra-royaumes	13
a. Préparation des données	13
b. Rendu visuel des transferts intra-royaumes	14
F. Diagramme en cordes	14
G. Carte de localisation des diocèses.....	15
III. Mise en place de rendus (carto)graphiques sur un site web : outils et choix techniques employés	17
A. Visualisation de graphes : comparatif de sigma.js et d3.js	17
B. Réalisation de rendus cartographiques : combiner d3 & leaflet.....	18
C. Affichage d'une frise chronologique	24
IV. Réalisations et améliorations envisagées.....	26
A. Carte des espaces qui fonctionnent ensemble	26
B. Carte des transferts inter-royaumes	26
C. Analyse des transferts intra-provinces et inter-provinces	26
D. Mise à jour des données directement à partir de la base de données	27
Conclusion.....	28
Bibliographie	29
Liens utiles.....	29

Introduction et présentation des objectifs

Le projet présenté s'inscrit dans un cadre particulier combinant plusieurs aspects :

- Une composante historique au travers de l'utilisation de données retraçant des événements ayant eu lieu au XIV^{ème} siècle,
- Une composante géographique comme il s'agit de l'étude de déplacements,
- Une composante informatique par l'aspect traitement de données et mise en place de rendus sur le web.

Au sein de l'Eglise latine, l'évêque représente l'autorité apostolique sur le diocèse et est nommé par le pape. Au XIV^{ème}, l'évêque avait un rôle d'administrateur en plus de son rôle religieux sur son propre diocèse. Les diocèses étaient sous l'autorité directe du Saint-Siège ou étaient regroupés en province sous l'autorité d'un archevêque ou d'un métropolitain (Mayeur et al., 1990).

Les déplacements et nominations des évêques au XIV^{ème} siècle étaient influencés par de multiples facteurs. Ceux-ci cherchaient souvent à être promus à une meilleure place au sein d'un diocèse plus prospère ou à devenir archevêque. Cela encourageait leur mobilité à travers l'Europe. Certains diocèses pouvaient notamment servir de passage obligé vers un poste convoité au sein des institutions de l'Eglise Latine. D'autres diocèses étaient plus isolés et moins sujets à des changements épiscopaux, les évêques nommés restant généralement en place.

Les nominations des évêques étaient influencées par les autorités des différents royaumes catholiques mais surtout par le pape. Ce dernier nommait généralement un nombre conséquent d'évêques la première année et dixième année de son règne, étendant ainsi son influence. Les autorités régaliennes avaient aussi un impact sur les nominations des hommes d'Eglise, évêques, cardinaux ou papes.

Les déplacements épiscopaux permettent-ils de faire émerger des entités regroupant des évêchés fortement connectés ? Ces regroupements d'évêchés, ont-ils un lien avec des entités administratives ou des groupes culturels ? Dès lors, l'objectif de ce projet a été de trouver des méthodes pertinentes de représentation des évêchés et des déplacements épiscopaux à des fins d'analyse et de visualisation en ligne. Il faut donc parvenir à trouver des solutions adaptées au support web permettant de produire des représentations (carto)graphiques en adéquation avec les données et les résultats d'analyse que nous pourrions obtenir. Il est également nécessaire de prendre en compte autant la dimension spatiale que temporelle des données utilisées concernant les mobilités épiscopales.

Organisation du projet et gestion des tâches

La période du projet a été globalement scindée en trois étapes :

- Une partie de recherche et prospection, consistant à prendre en main les données à notre disposition, à trouver des données complémentaires, à découvrir et prendre en main des outils pertinents pour le projet
- Une partie consacrée à la production de données et à la réalisation de rendus visuels permettant d'analyser les déplacements épiscopaux
- Une dernière partie visant à rédiger le rapport et finaliser les rendus (site web, base de données)

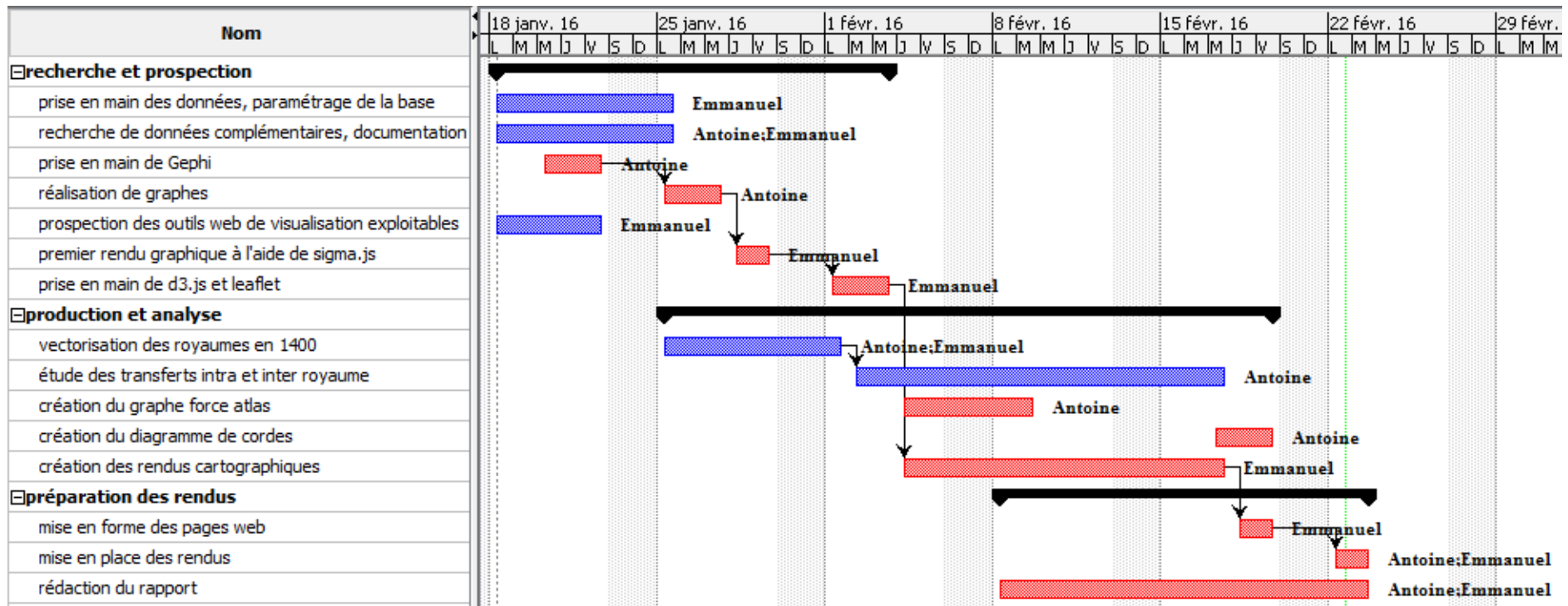


Figure 1 : diagramme de Gantt du projet
 Nb : les couleurs n'ont pas de signification particulière

Certaines tâches ont été réalisées sur une longue période car elles ont nécessité des modifications conséquentes après discussion avec le commanditaire (l'étude des transferts au sein des royaumes notamment). La réalisation des rendus cartographiques a consommé beaucoup de temps afin d'aboutir aux résultats attendus et répondre aux demandes du commanditaire. Dans la partie production, une personne se chargeait du paramétrage des données et l'autre s'occupait ensuite de la partie mise en ligne et visualisation de celles-ci. La rédaction du rapport a été entamée à mi-parcours et s'est intensifiée les derniers jours du projet.

I. Mise en place de la base de données

A. Données spécifiques à la mobilité épiscopale

Les principales données employées lors du projet concernent les diocèses et évêques du XIV^{ème} siècle. La période étudiée s'étend de 1305 à 1378. Entre ces bornes temporelles, la quasi-totalité des mobilités épiscopales ont été recensées dans une base de données en s'appuyant sur une multitude de sources, principales vaticanes.

Les données ont été fournies par le commanditaire du projet, M. Labarthe, sous la forme d'un ensemble de fichiers SQL provenant d'une BDD MySQL. Chaque fichier correspondait à une table de la BDD. Les liens entre les différentes tables étaient absents et ont été recréés.

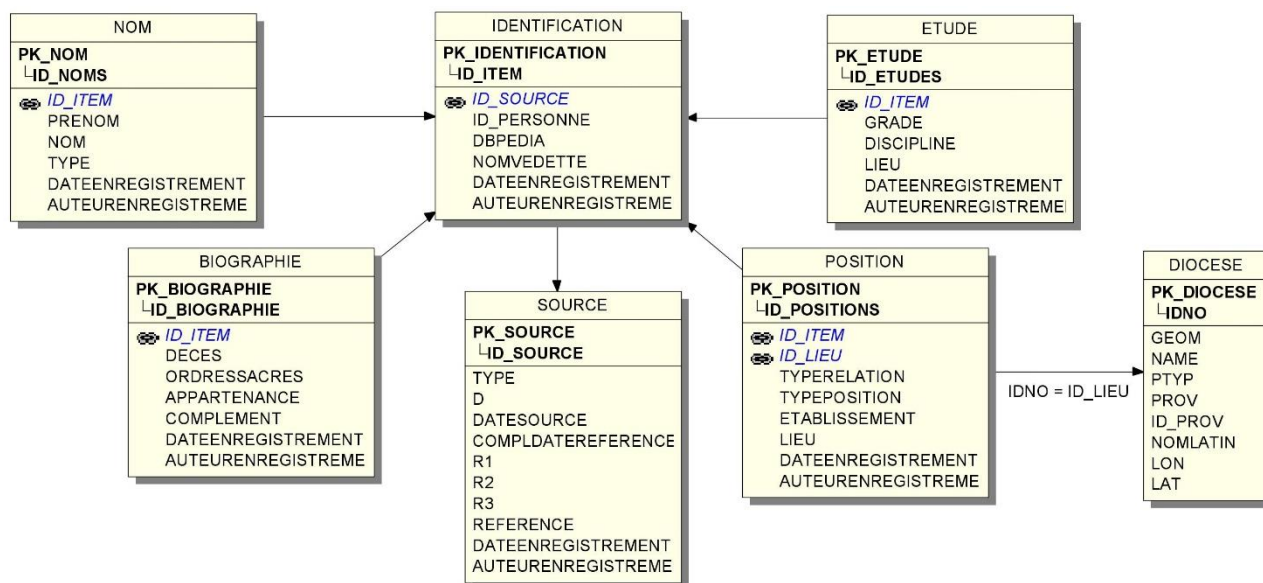


Figure 2 : Modèle Logique de Données concernant la mobilité épiscopale

Les données fournies comportent 7 tables :

- **Source** : ensemble des références employées afin d'alimenter la BDD. Cette table permet notamment de dater les sources à partir du champ *datesource* et de connaître la date de nomination des évêques.
- **Identification** : table rassemblant les identifications de personnes à partir des sources. Une personne pouvant correspondre à plusieurs entrées dans cette table, la valeur du champ *id_personne* est unique pour chaque personne distincte. Le champ *nomvedette* fournit les noms des évêques. A partir de cette table pivot, un ensemble d'informations sont

connectées concernant la biographie, le nom et les études réalisées par la personne identifiée (tables **biographie**, **nom** et **etude**).

- **Position** : ensemble des informations concernant les déplacements épiscopaux. Chaque entrée correspond à une arrivée au sein d'un diocèse (nomination, transfert) ou à un départ (résignation). Cette information est précisée au sein du champ *typeRelation*. Ainsi, chaque déplacement est indiqué par deux entrées dans la table : une permettant d'identifier le lieu de départ et une permettant d'identifier le lieu d'arrivée.
- **Diocese** : ensemble des diocèses recensés avec leur nom et leur localisation (les champs *lon* et *lat* correspondent aux coordonnées géographiques de l'évêché).

Dans un premier temps, nous avons importé les tables brutes fournies sur un serveur MySQL local. Comme la table **diocese** comporte des données géographiques, nous avons migré vers un serveur PostgreSQL local les données afin de pouvoir utiliser les fonctions PostGIS dans la suite du projet. Les données ont d'abord été exportées en CSV à partir de MySQL Workbench puis importées sur une base de données PostgreSQL/PostGIS à l'aide de QGIS.

Nous avons ensuite paramétré les clés étrangères dans la base de données afin de nous assurer de la cohérence des données. Il s'est avéré que des identifiants référencés étaient absents de la BDD. Nous avons paramétré les valeurs des références s'appuyant sur ces identifiants à NULL afin de créer des clés étrangères valides.

```
--lien entre la table source et identification
--une ligne modifiée
UPDATE identification
SET "ID_source" = NULL
WHERE "ID_source" IN (
  SELECT a."ID_source"
  FROM identification a LEFT JOIN source b ON a."ID_source" = b."ID_source"
  WHERE b."ID_source" IS NULL
);

ALTER TABLE identification
ADD FOREIGN KEY ("ID_source") REFERENCES source("ID_source");
```

Cette première correction concerne en définitive peu d'entrées :

Table comportant la clé étrangère	Table référencée	Lignes modifiées
identification	source	1
nom	identification	1
biographie	identification	1
etude	identification	0
position	identification	8
position	diocese	406

B. Intégration de données complémentaires à des fins d'analyse

Les données complémentaires ajoutées à celles concernant les déplacements épiscopaux visent à mettre en relation ces mouvements avec des données concernant les royaumes en Europe et l'accessibilité des diocèses (au travers de l'altitude et de la proximité du réseau routier).

1. Limites des Etats souverains

Les liens entre les tables étant désormais fixés, nous avons cherché à ajouter des données complémentaires concernant les provinces et les royaumes. Dans un premier temps, nous pensions

employer les données fournies par M. Labarthe sous forme de fichier SQL. Cependant, après import sur la BDD et visualisation des données, nous avons constaté que celles-ci étaient assez incomplètes et que les polygones des Etats pouvaient se superposer par endroit.

Nous ne sommes pas parvenus à trouver de données gratuites vectorielles concernant les limites territoriales au XIV^{ème} siècle. Cependant, des images raster de l'Europe à cette époque sont accessibles sur certains sites¹. Après avoir géoréférencé à l'aide de QGIS une image représentant l'Europe en 1400, nous avons créé un fichier de formes représentant les Etats souverains de l'époque.

Même si la donnée n'est pas complète, notamment en ce qui concerne le Saint Empire romain germanique, cela permet de disposer des limites des principaux royaumes de l'époque. Cette donnée va pouvoir être employée afin d'identifier les déplacements à l'intérieur des Etats.

2. Altitude et proximité au réseau routier des diocèses

Nous avons d'abord cherché à récupérer des informations concernant l'altitude des diocèses présents dans la BDD. Dans un premier temps, nous pensions employer le MNT du SRTM disponible sous forme de tuiles raster sur la quasi-totalité du globe terrestre². Cependant, ces données sont extrêmement lourdes pour un traitement à l'échelle de l'Europe. Le MNT a en effet une résolution de 30 m. Nous n'avons pas trouvé de MNT répondant à nos besoins en terme de couverture et de volume de traitement.

Une solution plus légère et adaptée à nos besoins a été trouvée. Nous avons exporté la table des diocèses comportant leurs coordonnées géographiques au format KML en connectant la BDD à QGIS et en réalisant l'export à partir de ce dernier. A l'aide d'un service disponible sur le web³, nous avons pu récupérer l'altitude des points en fournissant leurs coordonnées géographiques contenues dans le fichier KML. Ce site s'appuie sur une combinaison de plusieurs MNT mondiaux et couvre l'Amérique du Nord et la totalité de l'Europe. Le résultat du traitement est renvoyé sous forme de fichier texte avec une colonne comportant l'altitude des points. Nous importons sur QGIS le fichier de points afin de le transférer ensuite sur la BDD et d'ajouter un champ à la table des diocèses précisant l'altitude.

La seconde donnée employée en lien avec l'accessibilité des diocèses est la distance au réseau routier. Nous avons récupéré le réseau de routes romaines issu du Digital Atlas of Roman and Medieval Civilizations⁴. Aucune autre donnée exploitable concernant le réseau routier au Moyen-Age n'a pu être trouvée. Nous avons donc dû prendre comme hypothèse que les axes de déplacement présents sous l'empire romain étaient sensiblement similaires à ceux du XIV^{ème} siècle. Ces informations sont fournies sous forme d'un shapefile que nous avons importé via QGIS sur la BDD PostgreSQL.

II. Analyse et représentation de la mobilité épiscopale

L'étude de la mobilité des évêques peut être représentée au travers d'un graphe composé de nœuds (les évêchés) et de liens (les déplacements des évêques). Ces déplacements ont pour cause des nominations épiscopales ou des transferts vers d'autres diocèses. L'analyse de ces mouvements nous a conduit à la création de plusieurs rendus visibles sur ce site : <http://geotests.net/eveques/>.

¹ Voir info.euratlas.net

² Voir <http://dwtkns.com/srtm/>

³ Voir <http://www.gpsvisualizer.com/elevation>

⁴ Voir <http://darmc.harvard.edu/icb/icb.do?keyword=k40248&pageid=icb.page601659>

Cette partie du rapport vise à détailler les étapes de traitement des données ayant précédé la mise en place des rendus présentés sur le site. Les pages web réalisées lors du projet et illustrant les analyses sont les suivantes :

- http://geotests.net/eveques/localisation_diocese.html : localisation des évêchés et archevêchés avec possibilité d'affichage au survol sur un évêché des déplacements liés à ce dernier. Ce premier rendu permet en plus d'accéder à des informations de base concernant les mobilités épiscopales
- http://geotests.net/eveques/evol_temp.html : cartographie des déplacements épiscopaux sur une période délimitée
- http://geotests.net/eveques/transferts_intra.html : cartographie des transferts épiscopaux intra-royaumes
- http://geotests.net/eveques/diagramme_cordes.html : diagramme en cordes des déplacements inter-royaumes
- http://geotests.net/eveques/force_atlas.html : visualisation du résultat issu de l'algorithme Force Atlas.

Certaines symbologies sont employées de façon identique sur l'ensemble des rendus afin de faciliter leur lecture : les cercles représentent systématiquement des évêchés et les lignes correspondent aux déplacements. Lorsque l'on sélectionne un nœud sur une représentation, les déplacements pour lesquels ce nœud est le point de départ sont en bleu et les déplacements pour lesquels ce nœud est le point d'arrivée sont en rouge.

A. Identification des déplacements des évêques

La donnée concernant les nœuds est déjà présente directement dans la BDD, il s'agit de la table *diocese*. Les liens doivent néanmoins être créés au travers d'une table *mouvement*⁵.

La table *position* présente des indications en latin concernant les déplacements épiscopaux au travers du champ *typeRelation*. Comme plusieurs formulations sont employées pour désigner un seul et même type de relation, nous avons ajouté un champ à la table *position* comportant une désignation simplifiée et uniformisée fournie par M. Labarthe du type de relation. Cette table recense non seulement les cas de déplacements épiscopaux mais aussi les promotions ou nominations d'évêques dans leur diocèse d'origine. Dans ce cas, l'évêque ne se déplace pas. Nous avons donc identifié les cas correspondant véritablement à des mobilités épiscopales dans la table *position*. Deux configurations ont été retenues :

- Les cas où une personne est "résignée" (elle renonce à une charge au sein d'un diocèse) afin d'être "pourvue" ailleurs. Il s'agit alors d'une première nomination épiscopale pour cette personne.
- Les cas où une personne est "résignée" et est "transférée". Il s'agit alors d'une personne déjà évêque et changeant de diocèse.

Ces deux possibilités couvrent la quasi-totalité des déplacements. Nous avons essayé d'autres combinaisons avec "a possédé", "a siégé", "démis", "suffragant". Mais cela ne concernait que des cas isolés ou des cas déjà traités avec les configurations retenues.

⁵ Un guide des tables créées et utilisées de la BDD est disponible en annexe I.

La table *mouvement* créée au travers de la requête présentée ci-dessous rassemble l'ensemble des déplacements en précisant notamment la personne concernée, la date du déplacement, la position et le lieu de départ et d'arrivée.

```
--création de la table des mouvements des évêques
CREATE TABLE mouvement AS (
  WITH t1 AS (
    --positions liées aux lieux de départ
    SELECT "ID_personne", "nomVedette", relation_equivalente,
"setPosition", "dateSource", "ID_lieu"
    FROM source a, identification b, position c
    WHERE c."ID_item" = b."ID_item"
    AND a."ID_source" = b."ID_source"
    AND "ID_lieu" IS NOT NULL
    AND relation_equivalente IN ('résigne')
  ),
  t2 AS (
    --positions liées aux lieux d'arrivée
    SELECT "ID_personne", "nomVedette", relation_equivalente,
"setPosition", "dateSource", "ID_lieu"
    FROM source a, identification b, position c
    WHERE c."ID_item" = b."ID_item"
    AND a."ID_source" = b."ID_source"
    AND "ID_lieu" IS NOT NULL
    AND relation_equivalente IN ('pourvu', 'a été transféré')
  )
  --création de la table mouvement en rassemblant les informations sur le
  départ et l'arrivée
  SELECT DISTINCT
    t1."ID_personne",
    t1."nomVedette",
    t1.relation_equivalente AS statut_depart,
    t2.relation_equivalente AS statut_arrivee,
    t1."dateSource",
    t1."setPosition" AS position_depart,
    t2."setPosition" AS position_arrivee,
    t1."ID_lieu" AS lieu_depart,
    t2."ID_lieu" AS lieu_arrivee
  FROM t1, t2
  WHERE t1."ID_personne" = t2."ID_personne"
  AND t1."dateSource" = t2."dateSource"
  AND t1."ID_lieu" != t2."ID_lieu"
  ORDER BY "ID_personne"
);
```

Cette table nous permettra d'étudier les déplacements des évêques entre diocèses à l'aide de la théorie des graphes.

B. Impact de l'accessibilité des diocèses sur les déplacements épiscopaux

Au XIV^{ème} siècle, certains diocèses sont situés dans des espaces difficiles d'accès soit parce qu'ils se situent en zone montagneuse et/ou qu'ils sont éloignés des principaux axes de communication. Pour autant, cela influence-t-il la mobilité des évêques vers ses lieux plus isolés ?

L'altitude des diocèses est la première information que nous avons mise en relation avec le degré⁶ de chaque diocèse. Nous avons ensuite calculé la distance entre le diocèse et le réseau routier romain en considérant les axes majeurs. Les diocèses situés hors de l'influence romaine (Europe du l'Est et du Nord notamment) ont été exclus du calcul comme nous ne disposons d'aucune information sur les routes situées dans ces régions. Afin d'évaluer les diocèses à exclure, nous avons employé l'enveloppe concave du réseau routier estimée à l'aide de PostGIS :

```
CREATE TABLE enveloppe AS (
  SELECT ST_ConcaveHull(ST_Union(geom), 0.99) AS geom
  FROM roman_road
  WHERE class = 'Major Road'
);
```

Le résultat de cette requête ainsi que les diocèses pris en compte sont visualisables sur la carte de couverture des diocèses par le réseau routier⁷. Le calcul de la distance au réseau routier est obtenu à l'aide de la fonction ST_Distance de PostGIS appliquée à des coordonnées géographiques en degrés (WGS84). Le fait de préciser qu'il s'agit du type *geography* permet d'obtenir une distance en mètres et non en degrés :

```
--ajout d'une colonne par diocèse renseignant la distance minimale au
réseau routier
ALTER TABLE diocese
ADD COLUMN distance int;

UPDATE diocese
SET distance = dist
FROM
(
  SELECT idno, min(ST_Distance(a.geom::geography, c.geom::geography)) AS
dist
FROM diocese a, enveloppe b, roman_road c
WHERE ST_Intersects(a.geom, b.geom)
AND class = 'Major Road'
GROUP BY idno
) AS t1
WHERE diocese.idno = t1.idno;
```

Comme nous avons désormais pour chaque diocèse l'information concernant l'altitude et la distance au réseau routier, il est possible de calculer un indice de corrélation avec le nombre de déplacements par diocèse :

```
--corrélation entre l'altitude du diocèse et le nombre de mouvements
--corrélation entre la distance au réseau routier et le nombre de
mouvements
WITH t1 AS (
--nombre de départs par diocèse
SELECT idno, count(lieu_depart) AS nb_depart
FROM diocese LEFT JOIN mouvement ON idno = lieu_depart
GROUP BY idno
),
t2 AS (
--nombre d'arrivées par diocèse
SELECT idno, count(lieu_arrivee) AS nb_arrivee
```

⁶ Le degré d'un nœud (ici un diocèse) correspond au nombre de liens connectés à celui-ci.

⁷ Voir annexe 2 : couverture des diocèses par le réseau routier

```

FROM diocese LEFT JOIN mouvement ON idno = lieu_arrivee
GROUP BY idno
),
result AS (
--nombre de déplacements (départs + arrivées) par diocèse
SELECT t1.idno, nb_depart, nb_arrivee,
       nb_depart + nb_arrivee AS nb_mvt, altitude, distance
FROM t1 JOIN t2 ON t1.idno = t2.idno
       JOIN diocese ON t1.idno = diocese.idno
ORDER BY 4
)
--calcul de la corrélation entre le nombre de déplacements
--et l'altitude puis la corrélation entre le nombre de
--déplacements et la distance au réseau routier
SELECT corr(nb_mvt, altitude) AS correlation
FROM result
WHERE altitude > 0
UNION
SELECT corr(nb_mvt, distance) AS correlation
FROM result
WHERE distance IS NOT NULL;

```

Nous obtenons les résultats suivants :

- Corrélation entre le nombre de déplacements et l'altitude égale à -0.143.
- Corrélation entre le nombre de déplacements et la distance au réseau égale à 0.002.

Le nombre de déplacements apparaît comme étant relativement indépendant de l'altitude et de la distance aux routes d'après ces résultats. Afin de rendre ces résultats plus pertinents, il faudrait disposer du réseau routier du XIV^{ème} en Europe étant donné que nous avons ici considéré que les routes romaines étaient encore présentes sur la période étudiée. D'autres facteurs d'isolement pourraient aussi être considérés tels que la distance au centre urbain le plus proche mais peu de données concernant les villes et leur ampleur au XIV^{ème} siècle sont disponibles.

C. Visualisation des déplacements épiscopaux de manière spatialisée et non-spatialisée : utilisation du logiciel Gephi

Ces visualisations ont été réalisées par le biais de graphes composés des diocèses reliés entre eux par les déplacements des évêques. Pour réaliser des analyses à l'aide de graphes, nous avons le choix entre deux logiciels : Pajek et Gephi. Nous avons choisi d'utiliser Gephi, moins complexe à utiliser, plus « user-friendly ». Ce dernier nécessite des nœuds (correspondant à la table *diocese*) et des liens (correspondant à la table *mouvement*) sous forme de fichiers CSV. Ces deux tables sont exportées au format CSV à partir de la BDD PostgreSQL en utilisant GDAL et la commande *ogr2ogr* :

```

ogr2ogr -overwrite -f CSV mouvement.csv PG:"host=localhost user=postgres
dbname=eveque password=postgres" -sql "SELECT \"ID_personne\",
\"nomVedette\", statut_depart, statut_arrivee, \"dateSource\",
position_depart, position_arrivee, lieu_depart AS \"Source\", lieu_arrivee
AS \"Target\" FROM mouvement;"

ogr2ogr -overwrite -f CSV diocese.csv PG:"host=localhost user=postgres
dbname=eveque password=postgres" -sql "SELECT idno AS \"Id\", name AS
\"Label\", ptyp, prov, \"ID_prov\", nomlatin, lon, lat, altitude FROM
diocese WHERE lon IS NOT NULL;"

```

Les champs sont formatés avec des noms lus automatiquement par Gephi : *Id* correspond à l'identifiant du nœud, *Source* et *Target* correspondent aux identifiants de départ et d'arrivée des nœuds, *Label* est interprété comme étant le nom à afficher du nœud. Les champs *lon* et *lat* correspondent aux coordonnées des nœuds. Nous avons ignoré les diocèses non localisés.

En ayant renseigné tout cela, Gephi est à même de créer des liens entre les différents nœuds. Cependant, nous avons rencontré quelques soucis avec le logiciel, que nous allons expliquer ici, car il est impératif de les régler si l'on veut pouvoir l'utiliser correctement.

Tout d'abord, le plugin GeoLayout, qui permet d'avoir une répartition spatiale des nœuds, n'est pas compatible avec la version 0.9 de Gephi. Il est impossible de rouvrir un fichier enregistré sur lequel on travaille, il faut tout recommencer à chaque fois.

Il faut donc installer la version précédente (Gephi 0.8.2), sur laquelle GeoLayout fonctionne. Il est alors nécessaire d'utiliser une ancienne version de Java (Java 6 ou 7) car la dernière version de Java (Java 8) pose des soucis lors de l'utilisation du logiciel. Suite à ces petites modifications, le logiciel fonctionne parfaitement.

Nous avons utilisé deux modes de spatialisation :

- le GeoLayout, très intéressant puisqu'il propose une répartition géographique, avec en entrée les latitudes et les longitudes. Attention cependant, pour que le GeoLayout lise les longitudes, il faut créer une colonne portant le nom « lng » et non « lon ». On peut de ce fait avoir une représentation du réseau des mouvements des évêques au XIV^{ème} siècle, très massive par exemple au niveau de la France et de l'Italie. Nous avons utilisé le GeoLayout pour deux rendus sur le site web : la carte de localisation des diocèses, et la carte des déplacements épiscopaux selon une période délimitée.

- le Force Atlas (algorithme Force Atlas 2 dans Gephi) : ce mode de spatialisation n'est pas géographique. Il s'appuie sur une méthode itérative, c'est à l'utilisateur de le stopper lorsque le graphe lui convient. Il est basé sur un algorithme qui rapproche les nœuds très liés entre eux et éloigne les autres. Ainsi, les nœuds correspondant à des villes françaises et bourguignonnes s'entremêlent complètement, chose qui est moins visible sur une répartition géographique. D'autres espaces, comme l'Ecosse, sont complètement excentrés du modèle car les échanges se font en interne. Ces royaumes sont donc en autarcie en ce qui concerne les échanges épiscopaux, avec seulement deux villes servant de lien vers l'extérieur pour l'Ecosse.

Pour le rendu⁸, nous avons cette fois attribué une couleur aux royaumes, car la répartition n'étant pas spatialisée, il est impossible de repérer les différents royaumes en voyant le graphe. En faisant cela, on distingue aisément de grands ensembles de nœuds, qui forment de grandes taches colorées. Nous avons cependant gardé le degré pour la taille des nœuds, comme dans le graphe GeoLayout.

D. Etude des trajets effectués par une personne

Nous nous étions intéressés au nombre de trajets effectués par une personne, cependant les résultats n'ont pas été très probants. En effet, seules 127 personnes sur 4895 ont effectué plus de 2 transferts durant leur carrière, soit à peine 3 % des individus présents dans la BDD. La majorité des évêques ne sont pas transférés à un nouveau poste à la suite de leur première nomination.

```
SELECT COUNT(*) AS nbr_doublon, "ID_personne", "nomVedette"  
from position, identification  
WHERE position."ID_item" = identification."ID_item"
```

⁸ Voir http://geotests.net/eveques/force_atlas.html

```

AND "ID_personne" is NOT NULL
GROUP BY "ID_personne", "nomVedette"
HAVING COUNT(*) > 1
ORDER BY COUNT(*) DESC

```

Si l'on affichait la totalité des déplacements, cela nous donnait donc une carte avec un très grand nombre de trajets uniques, masquant complètement les trajets effectués plusieurs fois par une personne. En enlevant les trajets effectués une seule fois et en ne laissant que les trajets effectués au moins deux fois, la carte était en grande partie vide et il était assez difficile d'arriver à en tirer des explications pertinentes concernant les mouvements des évêques.

E. Réalisation de la carte des taux de transferts intra-royaumes

Initialement, nous voulions créer deux cartes : celle des taux de transferts intra-royaumes, et celle des transferts inter-royaumes. Finalement, nous nous sommes concentrés sur la carte intra-royaume, car nous ne sommes pas parvenus à trouver un mode de représentation qui nous convenait pour la seconde carte, et parce que nous avons dû faire des choix, du fait des délais.

a. Préparation des données

Il nous faut tout d'abord créer une table avec le nom du royaume en départ et en arrivée. Pour cela, nous utilisons le fichier de formes que nous avons créé ('europe_1400_intra.shp'), qui contient les noms des royaumes. Nous faisons ensuite une jointure spatiale, entre la table 'diocese' (ou 'diocese_final' si l'on veut utiliser notre sélection) et ce shape. Puis nous créons une nouvelle table qui contient les diocèses en départ et arrivée et les royaumes où ils se situent : il s'agit de la table 'mouvement_transfert'.

Nous pouvons ensuite calculer pour chaque royaume le nombre de transferts vers un autre royaume, et le nombre de transferts intra. Pour le nombre de transfert intra-royaume par exemple :

```

SELECT souverain, count(souverain)
FROM mouvement_transfert, diocese_final
WHERE (mouvement_transfert.lieu_depart = diocese_final.id_1)
AND (depart_souverain = arrivee_souverain)
AND ("dateSource" between '1305-00-00' and '1378-12-31')
GROUP BY souverain;

```

En faisant un GROUP BY « souverain », nous obtenons le nombre de transferts intra-royaumes par royaume. Nous faisons la même manipulation pour le nombre de transferts inter-royaume, puis nous pouvons faire une requête pour obtenir le taux de transferts intra, après avoir créé une colonne pouvant l'accueillir :

```

UPDATE mouvement_transfert
SET taux_intra = (intra_count*100)/(intra_count+inter_count);

```

Nous pouvons ensuite créer une table avec les taux intra-royaumes :

```

CREATE TABLE taux_transfert_intra_royaume AS (
  SELECT depart_souverain, taux_intra
  FROM mouvement_transfert
  WHERE taux_intra > 0
  GROUP BY depart_souverain, taux_intra
);

```

La précision « WHERE transfert_total > 0 » permet d'exclure du calcul les royaumes qui n'ont aucun transfert intra-royaume. Une fois que nous avons cette table, nous pouvons l'intégrer au shape via QGIS, puis exporter les données au format geojson pour être utilisable par d3.js.

b. Rendu visuel des transferts intra-royaumes

Le rendu réalisé⁹ permet de visualiser les royaumes de l'Europe occidentale (les frontières employées étant celles de 1400). Pour les principaux royaumes (France, Castille, Aragon, Angleterre, Ecosse, Autriche, Bohême, Danemark, Naples), le taux de déplacements à l'intérieur du royaume est représenté par un aplat de couleur. Nous avons fait le choix de ne pas afficher cette information pour les autres entités car celles-ci présentent parfois des délimitations discutables (notamment dans le Saint Empire Romain Germanique et à l'est de l'Europe).

Nous avons également ajouté une option de visualisation des déplacements intra-royaumes au passage du curseur sur les entités. On constate que les mouvements sont effectués majoritairement à l'intérieur des royaumes en Ecosse (80%), France (75%), Angleterre (74%), Castille (72%) et en Hongrie (71%). Seuls la Bohême (46%) et la monarchie de Habsbourg (5%) – qui appartiennent au Saint Empire Romain Germanique – ont des transferts majoritairement avec l'extérieur.

F. Diagramme en cordes

Afin de percevoir les liens entre les évêchés d'une manière différente, nous avons opté pour la représentation sous forme d'un diagramme en cordes¹⁰. Ce type de graphe permet de voir quels sont les espaces qui fonctionnent ensemble. C'est une autre manière de représenter les échanges de façon non spatialisée. Pour plus de lisibilité et pour que ce graphe soit intéressant, nous nous sommes uniquement penchés sur les royaumes qui ont le plus d'échanges au cours du XIV^{ème} siècle.

Concrètement, la réalisation de ce graphe nécessite en entrée un fichier .csv contenant une matrice « départ – arrivée », avec en ligne les départs, en colonnes les arrivées, et dans les enregistrements la somme des départs d'un royaume vers un autre.

Pour créer cette matrice, nous nous basons sur la table *mouvement*, puis nous créons une nouvelle table avec les royaumes de départ, et les royaumes d'arrivée. Une fois que nous avons cette table, nous faisons une sélection des royaumes qui nous intéressent vraiment :

```
CREATE TABLE diagramme_cordes_reduit AS (  
SELECT * FROM diagramme_cordes  
WHERE depart_souverain IN ('Royaume de France', 'Royaume de Naples', 'Etats  
Pontificaux', 'Castille',  
'Royaume d'Angleterre', 'Couronne d'Aragon',  
'Bourgogne', 'République de Venise', 'Duché de Milan', 'Comté de Provence',  
'Royaume de Hongrie', 'Royaume du Portugal',  
'Sardaigne', 'Principauté d'Achaïe', 'Padoue',  
'Royaume de Sicile', 'Florence')  
AND arrivee_souverain IN ('Royaume de France', 'Royaume de Naples', 'Etats  
Pontificaux', 'Castille',  
'Royaume d'Angleterre', 'Couronne d'Aragon',  
'Bourgogne', 'République de Venise', 'Duché de Milan', 'Comté de Provence',  
'Royaume de Hongrie', 'Royaume du Portugal',  
'Sardaigne', 'Principauté d'Achaïe', 'Padoue', 'Royaume de Sicile',  
'Florence')  
);
```

⁹ Voir http://geotests.net/eveques/transferts_intra.html

¹⁰ Voir http://geotests.net/eveques/diagramme_cordes.html

Nous pouvons maintenant créer notre table contenant la matrice dont nous nous avons besoin :

```
CREATE TABLE mouvement_matrix AS (  
SELECT start AS départ,  
SUM((dest = 'République de Venise')::INT) AS "République de Venise",  
SUM((dest = 'Duché de Milan')::INT) AS "Duché de Milan",  
SUM((dest = 'Padoue')::INT) AS "Padoue",  
SUM((dest = 'Royaume de France')::INT) AS "Royaume de France",  
SUM((dest = 'Royaume de Naples')::INT) AS "Royaume de Naples",  
SUM((dest = 'Couronne d'Aragon')::INT) AS "Couronne d'Aragon",  
SUM((dest = 'Castille')::INT) AS "Castille",  
SUM((dest = 'Royaume du Portugal')::INT) AS "Royaume du Portugal",  
SUM((dest = 'Royaume de Sicile')::INT) AS "Royaume de Sicile",  
SUM((dest = 'Royaume de Hongrie')::INT) AS "Royaume de Hongrie",  
SUM((dest = 'Florence')::INT) AS "Florence",  
SUM((dest = 'Royaume d'Angleterre')::INT) AS "Royaume d'Angleterre",  
SUM((dest = 'Etats Pontificaux')::INT) AS "Etats Pontificaux",  
SUM((dest = 'Comté de Provence')::INT) AS "Comté de Provence",  
SUM((dest = 'Bourgogne')::INT) AS "Bourgogne",  
SUM((dest = 'Principauté d'Achaïe')::INT) AS "Principauté d'Achaïe",  
SUM((dest = 'Sardaigne')::INT) AS "Sardaigne"--,  
FROM (  
SELECT diagramme_cordes_reduit.depart_souverain AS start,  
diagramme_cordes_reduit.arrivee_souverain AS dest  
FROM diagramme_cordes_reduit  
UNION ALL  
SELECT diagramme_cordes_reduit.arrivee_souverain AS start,  
diagramme_cordes_reduit.depart_souverain AS dest  
FROM diagramme_cordes_reduit) AS s  
GROUP BY start  
);
```

Il ne reste plus qu'à intégrer le fichier .csv à un script d3 permettant de réaliser le diagramme en cordes.

G. Carte de localisation des diocèses

Il s'agit de la carte représentant les diocèses et les archidiocèses¹¹. Les archidiocèses n'étant pas vraiment renseignés dans la base de données que nous avons reçue, nous avons établi une méthode pour les reconnaître :

```
CREATE TABLE archeveche AS (  
SELECT idno, "name"  
FROM diocese, position  
WHERE (diocese.idno = position."ID_lieu")  
AND position."typePosition" LIKE 'archiepiscopus'  
GROUP BY idno);
```

Lors d'un mouvement d'évêque, si le type de position associé à un diocèse est 'archiepiscopus', nous en concluons que celui-ci est un archidiocèse.

¹¹ Voir http://geotests.net/eveques/localisation_diocese.html

Sur Gephi : il faut importer la table *diocese.csv* (ou *diocese_final.csv*, qui est la version la plus avancée, dans laquelle les villes indésirables ont été supprimées) en table des nœuds, et la table *mouvement.csv* pour la table des liens. Concernant le rendu, nous avons conservé la même taille pour tous les nœuds, et distingué les évêchés et les archevêchés grâce aux couleurs. Les nœuds sont quelque peu transparents, afin de créer un effet d'accumulation sur la carte. Concernant les liens, nous avons conservé des liens droits (et non courbés comme le propose Gephi), car ils rendent mieux compte de l'intensité des transferts épiscopaux.

Les statistiques générales :

Nombre de diocèses : on ne garde que les diocèses pour lesquels les liens sont 'réels' (on supprime par exemple Jérusalem, ou Alexandrie, car personne n'est physiquement placé là-bas). On garde par contre les diocèses où il n'y a pas de lien, même s'ils sont éloignés. Au total, on obtient 808 diocèses.

Nombre d'archevêchés : même méthode que pour les diocèses.

Nombre de déplacements : nous prenons en compte les déplacements entre 1305 et 1378, cadre de l'étude. Nous obtenons ainsi 1923 déplacements.

Les « couples » de trajectoires : nous prenons ici le nombre de fois où nous avons des trajectoires identiques. Pour cela nous créons une nouvelle colonne sur la table des mouvements où nous concaténons l'ID de départ et d'arrivée, puis nous cherchons le nombre de fois où nous avons des doublons. Ainsi, nous avons 113 fois des trajets identiques effectués deux fois, 9 fois des trajets identiques effectués trois fois, 2 fois des trajets identiques effectués quatre fois, et enfin 1 fois un trajet identique effectué 5 fois.

On crée d'abord une nouvelle colonne dans 'mouvement' avec les couples de transferts :

```
SELECT (CAST(@lieu_depart AS VARCHAR(20) ) || CAST(@lieu_arrivee AS
VARCHAR(20) )) AS couples_tr
FROM mouvement
WHERE "dateSource" BETWEEN '1305-00-00' AND '1378-12-31';
```

Avec cette nouvelle colonne, nous pouvons faire un COUNT sur les couples identiques.

```
SELECT couples_tr, count(couples_tr) FROM mouvement
WHERE "dateSource" BETWEEN '1305-00-00' AND '1378-12-31'
GROUP BY couples_tr
HAVING count(couples_tr) = 2;
```

Ici par exemple nous cherchons les couples de transfert qui apparaissent deux fois. Il y en a 113.

Les interconnexions : Nous entendons par là le nombre de fois où nous avons un trajet d'un diocèse à un autre diocèse qui est effectué dans les deux sens. Cela se produit 134 fois entre 1305 et 1378. Voici la requête SQL correspondante :

```
SELECT distinct L1.lieu_depart, L1.lieu_arrivee FROM mouvement L1,
mouvement L2
WHERE L1.lieu_depart=L2.lieu_arrivee AND L1.lieu_arrivee=L2.lieu_depart AND
L1.lieu_depart>L1.lieu_arrivee
ORDER BY lieu_depart asc;
```


III. Mise en place de rendus (carto)graphiques sur un site web : outils et choix techniques employés

A. Visualisation de graphes : comparatif de sigma.js et d3.js

A partir de Gephi, il est possible d'exporter le résultat au format gexf (format d'export standard des graphes) ou json (format plus fréquemment lu par des bibliothèques javascript) afin de le visualiser sur une page web. Deux bibliothèques ont été testées au cours du projet afin de paramétrer un rendu graphique à partir des résultats obtenus sur Gephi : sigma.js et d3.js. Cependant, Gephi n'offre pas de possibilité d'export au format GeoJSON qui aurait été adapté pour des bibliothèques de visualisation cartographique comme Leaflet.

Nous avons dans un premier temps utilisé sigma.js qui nous a semblé adapté à une mise en place rapide d'un rendu à partir d'un fichier gexf. Cette bibliothèque permet en effet avec un minimum de code d'aboutir à un rendu basique avec des effets facilement paramétrables en s'appuyant sur les exemples présentés sur le site <http://sigmajs.org/>. Le survol du curseur sur les nœuds du graphe et la visualisation des éléments connectés entre eux au clic ont été mis en place rapidement.

Néanmoins, sigma.js présente de nombreuses limites d'utilisation. Certains paramétrages ne sont pas prévus dans la bibliothèque : il n'est par exemple pas possible de changer la taille des pointes de flèches à l'extrémité des liens sans modifier directement celle-ci.

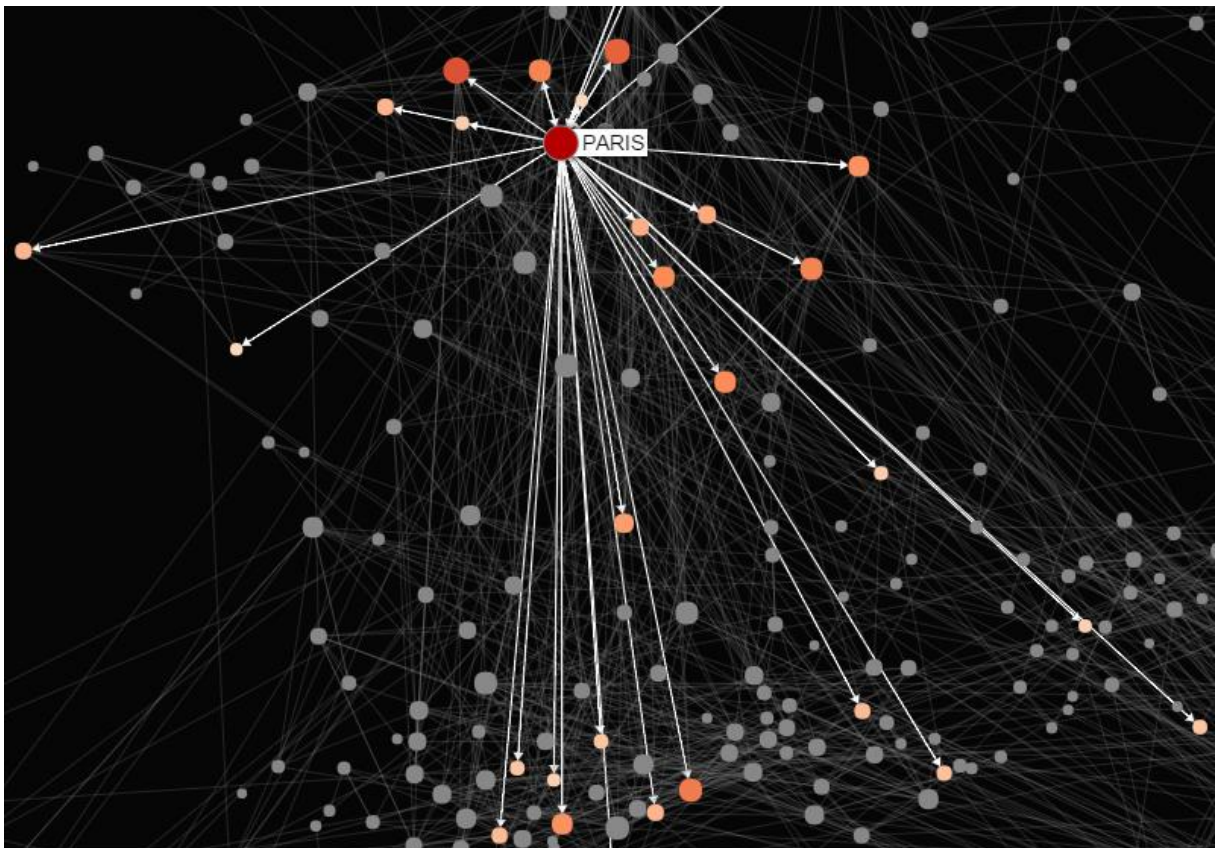


Figure 3 : première visualisation réalisée à partir de sigma.js (déplacements épiscopaux à partir et vers Paris)

De plus, la gestion des données géographiques est très réduite. La bibliothèque se limite à la visualisation de nœuds et de liens. Il n'est pas possible d'ajouter des lignes ou des polygones notamment. Par ailleurs, peu d'exemples élaborés s'appuyant sur sigma.js sont disponibles. Nous

n'avons pas trouvé de solutions de combinaison de cette bibliothèque avec une autre (Leaflet, OpenLayers par exemple) permettant de rajouter un fond cartographique.

C'est pourquoi nous avons décidé d'employer par la suite une seconde bibliothèque actuellement très utilisée : d3.js. Par rapport à sigma.js, d3 est plus difficile à prendre en main et demande plus de temps de paramétrage mais offre des possibilités conséquentes en terme de personnalisation du rendu et de méthodes de visualisation. La galerie d'exemples sur le site de la bibliothèque donne un aperçu des options disponibles¹². On y trouve notamment des solutions d'intégration de données géographiques.

	Sigma.js	D3.js
Dépendances	aucune	aucune
Version employée	1.0.3	3.5.14
Lecture des gexf	+++	++
Lecture des json	+++	+++
Facilité de prise en main	+++	+
Possibilités de paramétrage du rendu	++	+++
Documentation et exemples	++	+++
Diversité d'utilisation	+	+++
Gestion de données géographiques	+	++

Tableau 1 : comparatif de sigma.js et d3.js
(+ : mauvais, ++ : moyen, +++ : bon)

B. Réalisation de rendus cartographiques : combiner d3 & leaflet

Deux bibliothèques ont été combinées afin d'obtenir un rendu cartographique des graphes obtenus à l'aide de Gephi : leaflet afin de gérer le fond cartographique et l'interface et d3 permettant d'ajouter une surcouche svg sur la carte. Nous nous sommes inspirés d'exemples disponibles en ligne¹³.

Nous allons ici détailler une des visualisations réalisées lors du projet concernant la localisation des évêchés et archevêchés¹⁴. Celle-ci est paramétrée au travers de cinq fichiers :

- localisation_diocese.html (fichier de base de la page)
- js/d3_param_localisation_diocese.js (ensemble des scripts)
- css/style.css (paramétrage du style)
- data/localisation_diocese.json (données employées)
- header.html (en-tête appelé à chaque début de page)

La principale difficulté a été de configurer l'affichage d'une couche SVG à l'aide de d3 permettant de visualiser l'ensemble des évêchés et des déplacements sur la carte. Ce paramétrage se fait dans le fichier `d3_param_localisation_diocese.js`. Il est nécessaire d'initialiser la création d'éléments SVG sur la carte et d'ajouter un groupe (ici nommé "g") qui inclura l'ensemble des objets (nœuds et liens) ajoutés à l'aide de d3 :

¹² Voir <https://github.com/mbostock/d3/wiki/Gallery>

¹³ Voir les liens utiles renseignés en fin de rapport

¹⁴ Voir http://geotests.net/eveques/localisation_diocese.html

```
//initialisation de la carte
var map = L.map('map').setView([49, 8], 4);

/* initialisation de la couche SVG */
map._initPathRoot();

/* récupération de l'objet SVG de la carte */
var svg = d3.select("#map").select("svg");
svg.append("g");
```

Il faut tout d'abord paramétrer les informations concernant les nœuds (évêchés) et les liens (déplacements épiscopaux). En ce qui concerne les nœuds, il est nécessaire d'avoir les coordonnées dans le référentiel de la carte générée par leaflet. Il est possible de récupérer ces coordonnées en pixels à l'aide de la méthode *latLngToLayerPoint* :

```
// paramétrage de la donnée à afficher à l'aide de d3
d3.json("data/localisation diocese.json", function(error, json) {

  if (error) throw error;

  //paramétrage des données des noeuds
  json.nodes.forEach(function(d) { // pour chaque noeud
    d.LatLng = new L.LatLng(d.attributes.lat, d.attributes.lon);
    d.fixed = true;
    d.x = map.latLngToLayerPoint(d.LatLng).x;
    d.y = map.latLngToLayerPoint(d.LatLng).y;
  })
})
```

En ce qui concerne les liens, il faut connaître les objets javascript correspondant aux nœuds source et cible ainsi que le statut. Cette dernière information va permettre de faire le tri entre les premières nominations en tant qu'évêque et les transferts.

```
//paramétrage des données des liens
var edges = [];
json.edges.forEach(function(e) {
  var sourceNode = json.nodes.filter(function(n) {
    return n.id === e.source;
  })[0],
  targetNode = json.nodes.filter(function(n) {
    return n.id === e.target;
  })[0];

  edges.push({
    source: sourceNode,
    target: targetNode,
    statut: e.attributes.statut_arrivee
  });
});
```

Ensuite, nous paramétrons les objets svg correspondant aux nœuds et liens ainsi que leur rendu. Comme nous souhaitons représenter le sens de déplacement des évêques à l'aide de flèches placées au milieu des traits symbolisant les mouvements. Nous devons définir la forme de la pointe de flèche utilisée :

```

//définition des fleches
svg.append("defs").append("marker")
  .attr("id", "marker")
  .attr("viewBox", "0 -5 10 10")
  .attr("refX", 10)
  .attr("refY", 0)
  .attr("markerWidth", 4)
  .attr("markerHeight", 4)
  .attr("orient", "auto")
.append("path")
  .attr("d", "M0,-5L10,0L0,5");

```

Nous pouvons ensuite paramétrer les objets svg représentant les nœuds. On précise par l'attribut "r" le rayon du cercle. On ajoute des attributs liés à des évènements s'activant au passage du curseur sur l'objet (attributs "onmouseover" et "onmouseout"). Les fonctions se déclenchant par ce biais sont détaillées par la suite. La fonction stylizeNode est appliquée afin de configurer le style du rendu des cercles.

```

//paramétrage du style de rendu des points
var node = svg.selectAll(".node")
  .data(json.nodes)
  .enter().append("circle")
  .attr("r", 4)
  .attr("label", function(d) {
    return d.label; //ajout d'un attribut avec le nom de la ville
  })
  .attr("onmouseover", "highlight_point(evt)")
  .attr("onmouseout", "reset_evt(evt)")
  .attr("id", function(d) { //ajout d'un attribut comportant l'identifiant
    return d.id;
  });

stylizeNode(node);

//paramétrage du style de rendu des liens
var links = svg.selectAll(".link")
  .data(edges)
  .enter().append("polyline")
  .attr("class", "link")
  .attr("marker-mid", "url(#marker)")
  .attr("id-source", function(d) {
    return d.source.id; //ajout d'un attribut avec l'id du noeud source
  })
  .attr("id-target", function(d) {
    return d.target.id; //ajout d'un attribut avec l'id du noeud cible
  })
  .attr("statut", function(d) {
    return d.statut; //attribut précisant s'il s'agit d'un transfert ou
    d'une première nomination
  });

stylizeLink(links);

```

Les fonctions définissant le style de rendu des cercles et des polygones sont déclarées en début de script comme suit :

```
//fonction paramétrant le style par défaut pour les points
function stylizeNode(n) {
  n.style("stroke-opacity", 1)
  .style("stroke-width", 1)
  .style("stroke", function(d) {
    if (d.attributes.type_dioc === "diocese") {
      return "OrangeRed";
    } else {
      return "Indigo";
    }
  })
  .style("fill", function(d) {
    if (d.attributes.type_dioc === "diocese") {
      return "OrangeRed";
    } else {
      return "Indigo";
    }
  });
  hide ? n.style("opacity", .2) : n.style("opacity", .6);
}

//fonction paramétrant le style par défaut pour les liens
function stylizeLink(l) {
  l.style("stroke-width", 2)
  .style("opacity", 0);
}
```

Les liens sont chargés mais ne sont en fait pas visibles (l'opacité est nulle). Ils ne s'afficheront qu'au passage du curseur sur un nœud. Le dernier réglage concernant l'affichage concerne l'ajustement des nœuds et liens en fonction de la carte. A chaque chargement (lors d'un zoom ou d'un déplacement), les coordonnées des nœuds et des liens sont recalculées et adaptées à la carte. A noter que la polyligne se compose de 3 points (les deux extrémités et un point au milieu). Ce point intermédiaire sert d'accroche pour placer la pointe de flèche sur la ligne.

```
//se déclenche lors d'un chargement/rechargement de la carte

map.on("viewreset", update);
update();

function update() {
  //ajustement du positionnement des points selon le zoom
  node.attr("transform", function(d) {
    if (d.fixed === true) {
      d.x = map.latLngToLayerPoint(d.LatLng).x;
      d.y = map.latLngToLayerPoint(d.LatLng).y;
      return "translate(" +
        map.latLngToLayerPoint(d.LatLng).x + "," +
        map.latLngToLayerPoint(d.LatLng).y + ")";
    }
  });
}

//paramétrage des liens à afficher entre les points
//au travers de l'attribut "points"
links.attr("points", function(d) {
```

```

        return d.source.x + "," + d.source.y + " " +
(d.source.x+d.target.x)/2 + "," + (d.source.y+d.target.y)/2 + " " +
d.target.x + "," + d.target.y
    });
}
});

```

Le rendu basique de la carte est désormais paramétré, on ajoute ensuite de l'interaction pour l'utilisateur et une infobulle. Cette dernière est paramétrée à l'aide des fonctionnalités offertes par Leaflet. Elle servira à afficher les informations concernant le nom de l'évêché survolé au curseur et le nombre de départs et d'arrivées d'évêques pour ce noeud.

```

//div d'affichage du nom de la ville intégrée à la carte
var info = L.control();

info.onAdd = function (map) {
    this._div = L.DomUtil.create('div', 'info');
    this._div.id = "info";
    this.update();
    return this._div;
}

info.update = function(name, countStart, countEnd) {
    this._div.innerHTML = "<h6 class='text-center'>" + (
        name ?
        name + "</h6><label>" + countStart + (countStart > 1 ? " départs<br>"
: " départ<br>") + countEnd + (countEnd > 1 ? " arrivées</label>" : "
arrivée</label>"):
        "Cliquez sur un évêché pour plus d'informations" + "</h6>"
    );
}

info.addTo(map);

```

Afin de visualiser les liens par évêché sans pour autant rendre la carte illisible en affichant la totalité des déplacements, on utilise une fonction se déclenchant au passage du curseur sur un nœud. L'utilisateur peut afficher les nominations de nouveaux évêques, les transferts ou la totalité des liens. On récupère tout d'abord les liens à afficher dans les variables *linkTarget* (liens arrivant au nœud sélectionné) et *linkSource* (liens partant du nœud sélectionné).

```

function highlight_point(e) {

    //récupération des liens de l'objet sélectionné
    if (document.getElementById("transfert").checked &&
document.getElementById("nomination").checked) { //si on affiche les
transferts et premières nominations
        var linkTarget = d3.selectAll("[id-target='" + e.target.id + "']");
//liens dont la cible est sélectionnée
        var linkSource = d3.selectAll("[id-source='" + e.target.id + "']");
//liens dont la source est sélectionnée

    } else if (!document.getElementById("transfert").checked &&
document.getElementById("nomination").checked) { //si on affiche que les
premières nominations
        var linkSource = d3.selectAll("[id-source='" + e.target.id +
"']").filter(function (l) {return l.statut === "pourvu";});

```

```

    var linkTarget = d3.selectAll("[id-target='" + e.target.id +
    "')").filter(function (l) {return l.statut === "pourvu";});

    } else if (document.getElementById("transfert").checked &&
    !document.getElementById("nomination").checked) { //si on affiche que les
    premières nominations
        var linkSource = d3.selectAll("[id-source='" + e.target.id +
        "')").filter(function (l) {return l.statut === "a été transféré";});
        var linkTarget = d3.selectAll("[id-target='" + e.target.id +
        "')").filter(function (l) {return l.statut === "a été transféré";});

    } else {
        //affichage du nom de l'evêche sélectionné dans l'infobulle
        info.update(e.target.attributes.label.value, 0, 0);
        return; //sortir de la fonction
    }

    //cacher les éléments
    hide = true;

    //réinitialisation des styles pour tous les points et polygones sauf
    ceux de la légende
    stylizeNode(d3.selectAll('circle:not([svgLegend="true"])'));
    stylizeLink(d3.selectAll("polyline"));

    hide = false;

```

Selon les options choisies par l'utilisateur, cette fonction va conduire à afficher soit les déplacements provenant du nœud survolé, soit les déplacements aboutissant à ce nœud, soit les deux. Les nœuds connectés sont également mis en avant avec changement de style.

```

    if (!document.getElementById("mvt_source").checked) { //si l'on affiche
    tous les déplacements ou que les déplacements arrivant au point sélectionné
    linkTarget.style("stroke", "red") //affichés en rouge
    .style("opacity", 0.6) //réduction de la transparence
    .each(function(l) { //pour chaque lien
        var attrib = d3.select(this).attr("id-source"); //récupération de
    l'id source
        d3.select("[id='" + attrib + "')").style("stroke", "yellow")
        .style("stroke-width", 2)
        .style("opacity", 0.8); //modification du style du noeud cible
    });
    }

    if (!document.getElementById("mvt_target").checked) { //si l'on affiche
    tous les déplacements ou que les déplacements à partir du point sélectionné
    linkSource.style("stroke", "blue") //affichés en bleu
    .style("opacity", 0.6) //réduction de la transparence
    .each(function(l) { //pour chaque lien
        var attrib = d3.select(this).attr("id-target"); //récupération de
    l'id cible
        d3.select("[id='" + attrib + "')").style("stroke", "yellow")
    //modification du style du noeud source
        .style("stroke-width", 2)
        .style("opacity", 0.8);
    });
    }

```

Enfin, la fonction va mettre à jour les informations concernant le point sélectionné à l'aide de la méthode *update* appliquée à la variable *info* correspondant à l'infobulle. Le style du point sélectionné est également changé.

```
//affichage du nom de l'evêche sélectionné dans l'infobulle
info.update(e.target.attributes.label.value, linkSource.size(),
linkTarget.size());

//changement de style pour le noeud sélectionné
e.target.style.fill = "yellow";
e.target.style.stroke = "yellow";
e.target.style.strokeWidth = 2;
e.target.style.opacity = 0.9;
}
```

Il faut également s'assurer que cet effet de survol disparaît lorsque le curseur sort du point. Cela est géré par la fonction *reset_evt* ci-dessous :

```
function reset_evt(e) { //fonction s'activant à la sortie d'un noeud svg
//réinitialisation des styles pour tous les points et polygones sauf
ceux de la légende
stylizeNode(d3.selectAll('circle:not([svgLegend="true"])'));
stylizeLink(d3.selectAll("polyline"));
//réinitialisation de l'infobulle
info.update();
}
```

En résumé, l'ensemble du script présenté ci-dessus permet d'afficher au-dessus d'un fond cartographique des objets provenant d'un graphe créé sur Gephi et d'ajouter de l'interactivité pour l'utilisateur afin de lui apporter des informations supplémentaires. La méthodologie suivie ici a été employée de façon analogue pour tous les rendus cartographiques produits.

C. Affichage d'une frise chronologique

Un des éléments à considérer afin d'analyser les déplacements épiscopaux est la dimension temporelle des données. Cette évolution temporelle des données est illustrée au travers de la cartographie des déplacements épiscopaux selon une période délimitée¹⁵

Pour réaliser ce type de représentation temporelle, il nous paraît nécessaire de pouvoir sélectionner des déplacements épiscopaux compris entre deux bornes temporelles, chaque mouvement étant fixé à une date unique dans la BDD (sans prise en compte du temps de parcours). Nous souhaitons offrir à l'utilisateur deux approches afin de sélectionner une période :

- Une sélection libre où l'on peut indiquer deux dates manuellement ou par saisie sur un calendrier
- Une sélection suggérée à partir de périodes clés du XIV^{ème} siècle comme les différents règnes des papes ou la période de la peste noire.

Le premier type de sélection nécessite la mise en place d'un affichage des dates à l'aide d'un calendrier. Comme nous utilisons Bootstrap pour structurer le site, nous avons employé la bibliothèque Bootstrap 3 Datetimepicker qui permet de mettre en place une zone de saisie d'une date complétée

¹⁵ Voir : http://geotests.net/evêques/evol_temp.html

par un calendrier configurable en français. Son seul défaut est son nombre important de dépendances (jQuery, Bootstrap, moment.js et la version française de moment.js).

Pour le second type de sélection, nous avons employé la bibliothèque vis.js qui comprend une partie timeline dédiée à la création de frises chronologiques intégrées à une page web. Celle-ci s'est avérée assez simple à prendre en main à partir des nombreux exemples présents sur le site dédié à la bibliothèque. Les éléments à afficher sur la frise chronologique sont facilement modifiables car insérés dans un jeu de données :

```
//récupération de la div contenant la frise
var frise = document.getElementById('visualization');

// Création du jeu de données de la frise chronologique des papes
var items = new vis.DataSet([
  {id: 1, content: 'Benoît XI', start: '1303-09-22', end: '1304-07-07',
group: 1, className: 'white'},
  {id: 2, content: 'Clément V', start: '1305-06-05', end: '1314-04-20',
group: 1, className: 'white'},
  {id: 3, content: 'Jean XXII', start: '1316-08-07', end: '1334-12-04',
group: 1, className: 'white'},
  {id: 4, content: 'Benoît XII', start: '1334-12-20', end: '1342-04-25',
group: 1, className: 'white'},
  {id: 5, content: 'Clément VI', start: '1342-05-07', end: '1352-12-06',
group: 1, className: 'white'},
  {id: 6, content: 'Innocent VI', start: '1352-12-18', end: '1362-09-12',
group: 1, className: 'white'},
  {id: 7, content: 'Urbain V', start: '1362-09-28', end: '1370-12-19',
group: 1, className: 'white'},
  {id: 8, content: 'Grégoire XI', start: '1370-12-30', end: '1378-03-27',
group: 1, className: 'white'},
  {id: 9, content: 'Urbain VI', start: '1378-04-08', end: '1389-10-15',
group: 1, className: 'white'},
  {id: 10, content: 'Peste noire', start: '1347-01-01', end: '1352-12-31',
group: 2, className: 'red'},
  {id: 11, content: 'Guerre de Cent Ans', start: '1337-01-01', end: '1453-
12-31', group: 2, className: 'red'}
]);
```

La propriété group va permettre de séparer les règnes des papes des autres évènements :

```
//distinction des règnes papaux des autres évènements par des groupes
var groups = new vis.DataSet([
  {id: 1, content: 'papes'},
  {id: 2, content: 'évènements'}
]);
```

Il est possible de paramétrer les options d'affichage de la frise avant de la générer dans la div HTML correspondant à la variable frise :

```
// Configuration de la frise
var options = {
  groupOrder: 'id',
  start: "1300-01-01",
  end: "1400-01-01"
};

// Création de la frise
var timeline = new vis.Timeline(frise, items, groups, options);
```

L'interaction avec la frise par sélection d'un élément a ensuite été ajoutée afin de mettre à jour les dates et la carte affichée selon la période choisie.

IV. Réalisations et améliorations envisagées

A. Carte des espaces qui fonctionnent ensemble

Par manque de temps, nous n'avons malheureusement pas pu réaliser d'analyse spatialisée. Nous avons réalisé deux analyses sur les mouvements des diocèses : le Force Atlas et le diagramme en cordes, mais ceux-ci ne sont pas spatialisés.

Afin de réaliser ce type d'analyse, nous aurions pu par exemple créer une carte avec des triangles de Voronoï, permettant de passer de données ponctuelles à des données surfaciques. Cependant, il est difficile de définir les critères sur lesquels nous pourrions regrouper les diocèses ensemble. Comment définir les espaces qui fonctionnent ensemble ?

Pour cela, il existe plusieurs pistes intéressantes, comme la méthode de Louvain par exemple. C'est un algorithme adaptatif performant qui permet de détecter des communautés sur des graphes.

Par ailleurs, il peut être intéressant de se baser le diagramme en cordes ainsi que le Force Atlas pour repérer les espaces qui fonctionnent ensemble, et les représenter de manière spatialisée.

B. Carte des transferts inter-royaumes

Pour créer la table des transferts inter-royaume, c'est la même manipulation que pour la carte intra-royaumes. Il suffit de modifier (partie en grisé) la requête SQL :

```
SELECT souverain, count(souverain)
FROM mouvement_taux_transfert, diocese_final
WHERE (mouvement_taux_transfert.target = diocese_final.id_1)
AND (depart_souverain != arrivee_souverain)
AND ("dateSource" between '1305-00-00' and '1378-12-31')
GROUP BY diocese_final.souverain;
```

Plusieurs idées de représentation sont possibles, comme l'utilisation des centroïdes des polygones pour ensuite les relier entre eux, et créer ainsi une carte de flux. Les transferts inter-royaumes pourraient être représentés sous forme de flèches, avec une épaisseur de trait variant en fonction du nombre de transferts vers un autre royaume. Les flèches apparaîtraient à la sélection d'un royaume source en indiquant les destinations des évêques provenant du royaume sélectionné.

C. Analyse des transferts intra-provinces et inter-provinces

De la même manière que les transferts intra-royaumes et inter-royaumes, nous pensions au début du projet réaliser une carte des transferts intra-provinces et inter-provinces. Cependant, étant donné la difficulté que nous avons eue à réaliser un fichier de formes (.shp) des royaumes européens au cours du XIV^{ème} siècle, nous ne nous sommes pas lancés dans la réalisation plus poussée d'un shapefile contenant les provinces. De plus, le contour et la légitimité des royaumes au cours du XIV^{ème} siècle étant déjà assez discutables, nous n'avons pas voulu nous lancer dans les contours des provinces, qui sont encore plus discutables de la part d'historiens notamment.

Cependant, en faisant des recherches plus avancées et en ayant un peu plus de temps, il serait possible et intéressant de compléter le shapefile que nous avons créé pour y ajouter les provinces, et ainsi pouvoir faire des analyses plus précises des mouvements des évêques au cours du XIV^{ème} siècle.

Il serait également intéressant de créer un shapefile spécifique pour le Saint Empire Romain Germanique, qui est une entité à part entière.

D. Mise à jour des données directement à partir de la base de données

Lors de ce projet, nous avons travaillé sur une BDD qui n'a jamais été reliée directement au site mis en place. Toutes les données chargées sur ce dernier sont au format json ou gexf et proviennent de logiciels (Gephi, QGIS). Si les données devaient être mises à jour, il faudrait refaire le cheminement déjà réalisé et remplacer les fichiers de données utilisés. Afin d'éviter ce problème, il faudrait générer directement à partir des données hébergées sur un serveur PostgreSQL ou MySQL les résultats obtenus notamment au travers de Gephi.

Cela serait possible à l'aide d'un script PHP écrivant des variables javascript venant remplacer les fichiers json ou gexf nécessaires aux différentes visualisations. A chaque chargement d'une page, les infographies seraient ainsi formatées et actualisées en cas de changement intervenant sur la base de données.

Conclusion

Le projet que nous avons réalisé comportait plusieurs objectifs : tout d'abord proposer un moyen adapté pour visualiser les déplacements épiscopaux au cours du XIV^{ème} siècle. Ces données étant localisées (transferts d'évêques d'un diocèse à un autre), il y avait une forte composante géographique. Un travail conséquent était nécessaire en amont sur la base de données avant de pouvoir l'exploiter. L'objectif du projet consistait également à proposer une visualisation de ces données sur un site web, permettant à tous d'avoir accès aux résultats des recherches réalisées. De ce fait, il était primordial de proposer un site web agréable et qui soit ergonomique, facile à utiliser. Enfin, le dernier objectif essentiel était de produire des analyses pertinentes et nouvelles, permettant d'ouvrir de nouvelles perspectives.

Pour répondre à tous ces objectifs, nous avons mis en place un site web au moyen de divers outils (Bootstrap, bibliothèques `sigma.js`, `d3.js`, Leaflet notamment), qui facilitent la création d'un rendu agréable. Nous sommes parvenus à créer des cartes dynamiques et interactives en JavaScript (réponses aux actions de l'utilisateur avec la souris : au clic, au passage de la souris, utilisation de la frise chronologique, etc.).

Concernant les données, la mise en place d'une base de données sur PostgreSQL nous a permis de structurer les données pour ensuite travailler efficacement sur la réalisation des rendus sur Gephi notamment. Afin de produire des analyses nouvelles, nous nous sommes intéressés à des visualisations sous forme de graphe, qui permettent d'avoir une vision inédite des données : certaines, comme le diagramme en cordes ou le Force Atlas ne sont pas spatialisées et font ressortir des tendances peu perceptibles à l'aide d'une représentation géographique traditionnelle.

Notre projet comporte cependant quelques limites. Tout d'abord, il n'est pas évident de mettre à jour les rendus présents sur le site en cas de modification ou d'ajout des données. La réalisation du shapefile des frontières européennes de l'époque, de la part de non-historiens, était assez difficile. En effet, nous manquons de connaissances sur cette période qui est déjà en soi assez méconnue.

Partant de notre rendu, il est possible de réaliser un certain nombre de nouvelles analyses, comme les transferts inter-royaumes par exemple. Nous avons une analyse inter-royaumes avec le diagramme de cordes, mais qui n'est pas spatialisée. Pour créer une représentation géographique, il serait possible de réaliser une carte de flux partant des centroïdes des royaumes pour se diriger vers les autres centroïdes, avec Gephi par exemple. Il serait également intéressant de se pencher sur la méthode de Louvain pour faire ressortir des groupes de diocèses qui fonctionnent ensemble, et ainsi définir une nouvelle carte se superposant aux frontières officielles. Enfin, il serait vraiment bénéfique de migrer vers un site qui communiquerait directement avec la base de données, afin que les mises à jour se fassent instantanément, sans avoir à refaire toutes les étapes nécessaires actuellement.

Bibliographie

Mayeur J.-M., Pietri Ch., Vauchez A., Venard M., 1990. *Histoire du Christianisme, tome 6 : Un temps d'épreuves (1274 – 1449)*. Desclée, 945 p. (P.27 à 60) chapitre : Les cadres et les institutions de l'église latine

Liens utiles

Sources de données complémentaires

Carte de l'Europe en 1400 : http://info.euratlas.net/history/en_europe_1400_640.html

Limites côtières : <http://www.naturalearthdata.com/downloads/50m-physical-vectors/>

SRTM : <http://dwtkns.com/srtm/>

Routes romaines :

<http://darmc.harvard.edu/icb/icb.do?keyword=k40248&pageid=icb.page601659>

Outils de traitement

Récupération de l'altitude en connaissant les coordonnées géographiques de points :

<http://www.gpsvisualizer.com/elevation>

Bibliothèques web employées

Calendrier :

Bootstrap 3 Datepicker (version 4.17.37) : <https://eonasdan.github.io/bootstrap-datetimepicker/>

Cartographie :

Leaflet (version 0.7.7) : <http://leafletjs.com/>

Infographie :

D3 (version 3.5.14) : <https://d3js.org/>

Sigma (version 1.0.3) : <http://sigmajs.org/>

vis (version 4.14.0) : <http://visjs.org/>

Structuration de pages web :

Bootstrap (version 3.3.6) : <http://getbootstrap.com/>

Exemples d'utilisation des bibliothèques employées

Combinaison de d3 et leaflet : <https://ryanmatthewhorne.wordpress.com/2015/11/11/2-of-n-gephi-d3-js-and-maps-success/>

Création d'une carte choroplèthe avec leaflet : <http://leafletjs.com/examples/choropleth.html>

Création d'un diagramme de cordes : <http://coscimo.net/cordes.html>

Construire un diagramme de cordes avec D3 : <http://bl.ocks.org/mbostock/4062006>

Notice de la BDD et des fichiers produits

BDD :

Pour importer la base :

Créer une nouvelle base de données sur Postgresql → ajouter une extension postgis (Requête SQL : 'CREATE EXTENSION postgis ;') → via PGSQL Console, entrer :

\i 'D:/dossier/projet eveques/sql/dump_base eveques.sql' (par exemple). La base est créée.

- **archeveche** : table contenant la liste des archevêchés (nom des villes), et leur ID (issu de la table diocese).

- **biographie** : table d'origine

- **coordonnees_manquantes** : table contenant toutes les villes qui n'étaient pas géolocalisées. Nous avons ajouté un champs « editeurs », sur lequel nous avons précisé que nous avons ajoutées les coordonnées nous-même, afin de ne pas abîmer la table d'origine (diocese).

- **diagramme_cordes** : table contenant les royaumes en départ, et les royaumes en arrivée. Elle est le point de départ pour créer la matrice qui permet de réaliser le diagramme de cordes

- **diagramme_cordes_reduit** : table issue de 'diagramme_cordes', de laquelle nous avons supprimé un certain nombre d'enregistrements pour ne garder finalement que les royaumes qui ont le plus de transferts. Ce sont les suivants :

Royaume de France, Royaume de Naples, Etats Pontificaux, Castille, Royaume d'Angleterre, Couronne d'Aragon, Bourgogne, République de Venise, Duché de Milan, Comté de Provence, Royaume de Hongrie, Royaume du Portugal, Sardaigne, Principauté d'Achaïe, Padoue, Royaume de Sicile, Florence.

- **diocese** : table d'origine, à laquelle nous avons ajouté les champs 'editeurs', que nous avons évoqué plus tôt, et 'a_exclure', ayant une mention « yes », ou vide. Si « yes » est mentionné, la ville ne sera pas retenue pour la suite des traitements.

- **diocese_final** : avec champs « editeurs » (soit 'antoine et emmanuel', soit vide), qui permet de savoir que l'enregistrement a été modifié par nous. Cela permet de garder la table de base - 'diocese' - vierge. Ainsi, nous avons pu géolocaliser des diocèses qui ne l'étaient pas sans abîmer la table originale. De la même façon, nous avons créé un champ « a_exclure », qui contient une liste de villes à exclure :

Tripoli, Alexandrie, Jérusalem, Césarée, Nazareth, Tibérias, Sarafand, Saida, Tarsus, Antakya, Soltaniyeh, Naxcivan, Caffa, Tbilissi, Trabzon, Sivas, Istanbul, Rodos, Kos, Izmir, Bayt Ras + 5 villes dont nous n'avons pas les coordonnées géographiques (Tamatarkha, Navarzan, Neretva, Syia, Lavacensis).

- **equivalence** : table créée par M. Labarthe. Permet de nettoyer la table 'position' (le champ typeRelation). En réalisant une jointure entre ces deux tables, nous obtenons une nouvelle table beaucoup plus lisible avec un nouveau champ « relation_equivalente ». Cette table recense les différents types de transferts qui existent.

- **etude** : table d'origine.

- **identification** : table d'origine.

- **mouvement** : table vraiment centrale. Elle comprend tous les mouvements effectifs, et a été créée d'après une requête qui est détaillée dans le rapport. Cette table comprend l'ID de la personne, son nom, son statut en départ et son statut en arrivée, la date de la source, la position (évêque, archevêque, etc.) en départ et en arrivée, le lieu de départ et le lieu d'arrivée ainsi que le code « couple », qui permet ensuite de recenser les parcours identiques. Cette table est le point de départ de tous nos traitements sur Gephi ou autres.

- **mouvement_matrix** : table issue de la table 'diagramme_cordes_reduit'. Elle contient un tableau sous forme de matrice « départ – arrivée ». C'est la matrice nécessaire à la création du diagramme de cordes. Attention cependant : pour obtenir un diagramme de cordes sans les transferts intra-royaumes, il faudra changer des données, et mettre la valeur '0' sur l'enregistrement correspondant à 'Duché de Milan' vers 'Duché de Milan' (au lieu de '44' actuellement sur la table).

- **mouvement_transfert** : Table de référence pour créer la carte des transferts intra-royaumes (et inter-royaumes). Elle contient le royaume de départ, le royaume d'arrivée, le nombre total de transferts pour chaque royaume, le nombre de transferts intra-royaume pour chaque royaume, le nombre de transferts inter-royaumes pour chaque royaume, et enfin le taux de transfert intra-royaumes pour chaque royaume.

- **nom** : table d'origine.

- **position** : table d'origine.

- **source** : table d'origine.

- **taux_transfert_intra_royaume** : Table contenant - pour chaque royaume ayant des transferts intra-royaume – le taux de transferts intra-royaume par rapport au nombre total de transferts. Cette table permet par la suite, après avoir fait une jointure sur le shape des royaumes, de faire un export en .geojson afin d'exploiter ces données avec d3 et Leaflet.

Fichiers :

- Dossier 'carte_europe_1400' :

- **en_europe_1400.tif** : fichier .tif de référence pour la création de notre shapefile.

- **europe_1400_intra.shp** (+ autres extensions) : shapefile que nous avons créé en nous basant sur la carte en .tif. Ce shapefile contient le nom des royaumes, leur type (entité mineure/majeure), ainsi que le taux de transfert intra-royaumes pour chaque royaume.

- Dossier 'creation_base_de_donnees' :

- **dump_base_eveques.sql** : fichier permettant d'importer toute la base de données évêque que nous avons complétée.

Couverture des diocèses par le réseau routier

