The Stata Journal

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go "beyond the Stata manual" in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

The *Stata Journal* is indexed and abstracted by *CompuMath Citation Index*, *Current Contents/Social and Behavioral Sciences*, *RePEc: Research Papers in Economics*, *Science Citation Index Expanded* (also known as *SciSearch*), *Scopus*, and *Social Sciences Citation Index*.

For more information on the *Stata Journal*, including information for authors, see the webpage

http://www.stata-journal.com

Subscriptions are available from StataCorp, 4905 Lakeway Drive, College Station, Texas 77845, telephone 979-696-4600 or 800-782-8272, fax 979-696-4601, or online at

http://www.stata.com/bookstore/sj.html

Subscription rates listed below include both a printed and an electronic copy unless otherwise mentioned.

U.S. and Canada		Elsewhere	
Printed & electronic		**Printed & electronic**	
1-year subscription	$124	1-year subscription	$154
2-year subscription	$224	2-year subscription	$284
3-year subscription	$310	3-year subscription	$400
1-year student subscription	$ 89	1-year student subscription	$119
1-year institutional subscription	$375	1-year institutional subscription	$405
2-year institutional subscription	$679	2-year institutional subscription	$739
3-year institutional subscription	$935	3-year institutional subscription	$1,025
Electronic only		**Electronic only**	
1-year subscription	$ 89	1-year subscription	$ 89
2-year subscription	$162	2-year subscription	$162
3-year subscription	$229	3-year subscription	$229
1-year student subscription	$ 62	1-year student subscription	$ 62

Back issues of the *Stata Journal* may be ordered online at

http://www.stata.com/bookstore/sjj.html

Individual articles three or more years old may be accessed online without charge. More recent articles may be ordered online.

http://www.stata-journal.com/archives.html

The *Stata Journal* is published quarterly by the Stata Press, College Station, Texas, USA.

Address changes should be sent to the *Stata Journal*, StataCorp, 4905 Lakeway Drive, College Station, TX 77845, USA, or emailed to sj@stata.com.

The *Stata Journal* (ISSN 1536-867X) is a publication of Stata Press. Stata, Stata, Stata Press, Mata, Mata, and NetCourse are registered trademarks of StataCorp LLC.

The Stata Journal (2018)
18, Number 3, pp. 564–584

flowbca: A flow-based cluster algorithm in Stata

Jordy Meekes
Utrecht University
Utrecht University School of Economics
Utrecht, The Netherlands,
University of Melbourne
Melbourne Institute of Applied Economic and Social Research
Melbourne, Australia,
and IZA, Bonn, Germany
jordy.meekes@unimelb.edu.au

Wolter H. J. Hassink
Utrecht University
Utrecht University School of Economics
Utrecht, The Netherlands
and IZA, Bonn, Germany
W.H.J.Hassink@uu.nl

Abstract. In this article, we introduce the Stata implementation of a flow-based cluster algorithm, `flowbca`, written in Mata. The main purpose of `flowbca` is to identify clusters based on relational data of flows. We illustrate the command by providing multiple examples of applications from the research fields of economic geography, industrial input–output analysis, and social network analysis.

Keywords: st0535, flowbca, clusters, aggregation, flows, regions, industries, economic geography, input–output analysis, social network analysis

1 Introduction

In this article, we introduce the Stata implementation of a flow-based cluster algorithm, `flowbca`, written in Mata.[1] A flow variable registers the total change of the variable from one entity to another entity during a specific period of time. The entity can be a region, firm, or person. During the process of clustering, two entities will be grouped according to the size of the bilateral flows. Currently, flow-based cluster algorithms available in Stata focus on visualizing social networks (for example, Corten [2011] and Miura [2012]). However, these algorithms lack the ability to flexibly aggregate units into clusters based on relational data of flows. The main motivation to write `flowbca` is that there is a need in many statistical applications—and in research fields other than social network analysis—for an algorithm to flexibly aggregate nonoverlapping units into clusters. This is specifically because it provides a choice of how to operate clusters in empirical analyses and allows a researcher to compare alternative sets of clusters.

1. This article is based on a chapter in the dissertation of Meekes (2019).

Given the increasing availability and use of relational data of various types of flows, `flowbca` can be used in a variety of research fields. For example, the field of economic geography uses flows to cluster regional units into regional clusters of economic activity (Coombes, Green, and Openshaw 1986; Brezzi et al. 2012).[2] Alternatively, industrial input–output analysis is based on trade linkages that register the flows of goods that are produced in one production chain and used as input in another production chain (Leontief 1986; Timmer et al. 2015). Finally, social network analysis detects communities (Fortunato 2010) and defines flow networks (Ford and Fulkerson 1962; Beguerisse-Díaz et al. 2014) in graphs as connected groups based on the strength of flows between nodes.

`flowbca` is the implementation in Stata of a so-called agglomerative hierarchical clustering algorithm (Fortunato 2010) to define clusters based on relational data of flows, which has been used in the aforementioned research fields. The key difference between `flowbca` and other agglomerative hierarchical clustering algorithms currently available in Stata is the focus on flow-based clustering instead of distance-based clustering.[3] In general terms, the flow-based cluster algorithm behind `flowbca` can be described as follows: It starts from a set of K disjoint units. The algorithm aggregates two units into one, while considering the bilateral flows between the units. Clusters are defined by iteratively repeating this procedure. In each iteration of the algorithm, the decision criterion for aggregating two units into one is based on an optimization function selecting the maximum flow out of all bilateral flows. The source unit from which the largest flow starts is aggregated to the destination unit.

The algorithm is flexible in various aspects. First, the optimization function can be based on two definitions of flows—directed and undirected. The former refers to the maximum of the single directed flows that are flowing from one unit to another; the latter denotes the maximum of the sum of two bilateral flows. Second, the optimization function can be based on absolute flows and relative flows, which are computed by taking each absolute flow relative to the unit-specific total of outgoing flows. Third, the algorithm allows for flexibility in the stopping criterion by allowing for five optional ex ante user choices different from the default one. Using different options, the researcher can thus create different sets of clusters.

After the algorithm has been terminated, the researcher could evaluate the choice of optimization function and stopping criterion by analyzing the level of self-containment of the set of clusters. The level of self-containment is approximated by the average of the internal relative flows. A higher average of the internal relative flows means there is a stronger connectivity within each cluster and a weaker connectivity to outside clusters.

2. Global regional clusters of economic activity are defined using trade flows (Smith and White 1992), foreign direct investment flows (Bathelt and Li 2014), or multinational firms relocation flows (Chen and Moore 2010). Within-country regional clusters such as local labor markets or local housing markets can be defined using commuting flows from place of residence to place of work, job-to-job turnover flows, household migration flows, or job search flows (for example, Duranton [2015]).
3. Distance-based clustering uses the distance between a pair of units as a measure of similarity. Similar units are grouped into the same cluster, and dissimilar units are grouped into separate clusters.

2 The flow-based cluster algorithm

2.1 The algorithm

The main inputs of the algorithm are a K-dimensional square matrix that contains the absolute flows between K different units. Effectively, each row represents a different source unit, and each column represents a different destination unit.

The algorithm consists of the following five steps, which are provided given the default options of the algorithm that will be described in Section 3.

Step 1: the absolute flows between K different units are rewritten as a K-dimensional square matrix (that is, an adjacency matrix in graph theory) of absolute flows $\mathbf{F}^{(K)}$,

$$\mathbf{F}^{(K)} = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1K} \\ f_{21} & f_{22} & \cdots & f_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ f_{K1} & f_{K2} & \cdots & f_{KK} \end{bmatrix}$$

where f_{ij} $(i \neq j)$ represents the flow in absolute term from source unit i to destination unit j. Flows f_{ii} are defined as the internal absolute flows.

Step 2: the matrix $\mathbf{F}^{(K)}$ is rewritten in terms of relative flows as a K-dimensional square matrix $\mathbf{G}^{(K)}$. Relative flows are computed by taking each absolute flow relative to the unit-specific total of outgoing flows. $\mathbf{G}^{(K)}$ can be expressed as

$$\mathbf{G}^{(K)} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1K} \\ g_{21} & g_{22} & \cdots & g_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ g_{K1} & g_{K2} & \cdots & g_{KK} \end{bmatrix}$$

where

$$g_{ij} = f_{ij} / \sum_{t=1}^{K} f_{it} \qquad i, j = 1, 2, \ldots, K$$

Note that the matrix is row-normalized.

Step 3: the optimization function selects the arguments of the maximum directed relative flow from one unit to another, for $i \neq j$, of all $K \times (K-1)$ pairs of i and j,

$$(r, s) = \underset{\substack{i,j \\ i \neq j}}{\arg\max}\, g_{ij}$$

where units r and s are defined as the source unit and destination unit, respectively. If $g_{rs} = 0$ or $K = 1$, the default stopping criterion of the procedure is met, and the algorithm is terminated.

Step 4: source unit r will be aggregated to destination unit s. The core of the cluster is defined as the receiving unit, that is, the destination unit s. To be able to adjust

matrix $\mathbf{F}^{(K)}$, the algorithm specifies a $K \times (K-1)$-dimensional matrix $\mathbf{C}^{(K)}$, which can be expressed as

$$\mathbf{C}^{(K)} = (e_1, e_2, \ldots, e_{r-1}, e_r + e_s, e_{r+1}, \ldots, e_{s-1}, e_{s+1}, \ldots, e_K)$$

where e_i represents the ith unit column vector. For the sake of convenience in the exposition of this algorithm, matrix $\mathbf{C}^{(K)}$ is based on the assumption that the identifier value of unit r is larger than the identifier value of unit s.

Step 5: the absolute flows to and from units r and s will be added. The new matrix $\mathbf{F}^{(K-1)}$ can be expressed as

$$\mathbf{F}^{(K-1)} = (\mathbf{C}^{(K)})^T \mathbf{F}^{(K)} \mathbf{C}^{(K)}$$

where T refers to the transpose operator. $\mathbf{F}^{(K-1)}$ is now a square matrix of dimension $(K-1)$. The algorithm continues with step 1, starting with $\mathbf{F}^{(K-1)}$ as an input.

After the stopping criterion of step 3 has been met, the algorithm is terminated and K^* clusters are returned. The matrix of absolute flows between the K^* clusters, $\mathbf{F}^{(K^*)}$, equals

$$\mathbf{F}^{(K^*)} = \mathbf{C}^T \mathbf{F}^{(K)} \mathbf{C}$$

where matrix $\mathbf{C}$ is a matrix product of the matrices $\mathbf{C}^{(K)} \ldots \mathbf{C}^{(K^*)}$, which can be expressed as $\mathbf{C} = \mathbf{C}^{(K)} \mathbf{C}^{(K-1)} \mathbf{C}^{(K-2)} \ldots \mathbf{C}^{(K^*)}$.

2.2 Stopping criteria

In step 3, the stopping criterion of the algorithm is defined as $g_{rs} = 0$ or $K = 1$. The algorithm allows for five alternative stopping criteria, and each of them is a modification of the stopping criterion mentioned in step 3.

First, the researcher could specify a flow threshold, q, that represents the minimum level of interaction at which a source unit is aggregated to a destination unit. The algorithm is terminated in step 3 if $g_{rs} < q$.

Second, the researcher could specify a minimum number of clusters, k. The algorithm is terminated in step 3 if the number of units have reduced to this minimum, that is, if $k = K^*$.

Third, the researcher could specify a minimum average of the internal relative flows, l_a. The average of the internal relative flows, L_a, is defined as equal to the sum of the internal relative flows, g_{ii}, relative to the number of clusters, K^*:

$$L_a = \frac{1}{K^*} \sum_{i=1}^{K^*} g_{ii} \tag{1}$$

The algorithm is terminated in step 3 if $l_a \leq L_a$.

Fourth, the researcher could specify a minimum weighted average of the internal relative flows, l_w. The weighted average of the internal relative flows, L_w, is defined as equal to the sum of the internal absolute flows relative to the sum of all absolute flows,

$$L_w = \frac{1}{N} \sum_{i=1}^{K^*} f_{ii} \tag{2}$$

for which the sum of all absolute flows equals $N = \sum_{i=1}^{K^*} \sum_{j=1}^{K^*} f_{ij}$. The algorithm is terminated in step 3 if $l_w \leq L_w$.

Finally, the researcher could impose a minimum internal relative flow l_m that all the clusters must satisfy. The minimum of the internal relative flows L_m is defined by

$$L_m = \min_i g_{ii} \tag{3}$$

The algorithm is terminated in step 3 if $l_m \leq L_m$.

2.3 An alternative optimization function

The algorithm provides two different optimization functions, which are based either on the directed or on the undirected flows approach. The optimization function based on the directed flows selects arg max g_{ij}, considering the maximum of the directed flows from one unit to another. In contrast, the optimization function based on the undirected flows approach selects arg max $g_{ij} + g_{ji}$, considering the maximum of the sum of two bilateral flows, which can be expressed as

$$(r, s) = \arg\max_{\substack{i,j \\ i \neq j}} g_{ij} + g_{ji}$$

Using the undirected flows approach, the algorithm is terminated if $g_{rs} + g_{sr} = 0$ or $K = 1$. Otherwise, the procedure will continue with step 4 of the algorithm. Note that the new cluster gets the identification number of the unit with the largest incoming flow, which represents the core of the new cluster.

2.4 Some caveats

In step 3, $\arg\max g_{ij}$ might hold for multiple pairs of units i, j. Consequently, the pair r, s will not be unique. The following rules are imposed to close the algorithm:

1. If there are two or more source units r, for example, r_1 and r_2, that both have the maximum flow to the same destination unit s, r_1 is aggregated to s if r_1 has the highest incoming flow from the other source unit or units r.

2. If source unit r has identical flows to two or more units s, for example, s_1 and s_2, r is aggregated to s_1 if s_1 has the highest incoming flow from the other destination unit or units s.

3. If both r and s are not unique, for example, there are two pairs r_1,s_1 and r_2,s_2, the algorithm aggregates r_1 to s_1 if s_1 has the highest incoming flow from the other destination unit or units s.

4. For the iterations where a unique pair is still not defined, the algorithm picks one pair r, s of all pairs that correspond to the maximum flow.

3 The flowbca command

3.1 Syntax

`flowbca` *varname varlist* [`, q(#) k(#) la(`*fraction*`) lw(`*fraction*`) lm(`*fraction*`) opt_f(#) save_k`]

3.2 Description

`flowbca` implements the algorithm that is discussed in section 2 and performs it in Mata. The main inputs for `flowbca` are the variables listed in *varname* and *varlist*. *varname* contains one variable representing the source unit identifier. This variable should be numerical because string variables are ignored by the `flowbca` command. *varlist* contains a set of variables with one variable for each distinct destination unit that represents the absolute flows from the source units to the destination unit.

Effectively, the destination unit variables represent the columns of a K-dimensional square matrix of flows between the K units. For example, the value of the first observation of a destination unit variable represents the absolute flow from the first source unit to the corresponding destination unit. The source and destination units should be numbered such that if they are sorted and ordered in a sequential order, the diagonal elements of the K-dimensional square matrix represent the internal absolute flows. If the flow data of the researcher are available only in a $K \times 3$-dimensional matrix in which there are three columns that represent the source unit identifier, destination unit identifier, and absolute flows between the units, respectively, the data should be reshaped by the user into a K-dimensional square matrix.

3.3 Options

`q(#)` sets the flow threshold. To set a relative flow threshold, specify # as a fraction. To set an absolute threshold, specify # as an integer number. If the threshold is higher than the maximum of all flows, the stopping criterion of the procedure has been met, and the algorithm is terminated. The default is `q(0)`.

`k(#)` specifies the number of distinct clusters the algorithm should define. The default is `k(1)`.

`la(`*fraction*`)` specifies the minimum average of the internal relative flows. If the fraction is lower than or equal to the average of the internal relative flows, the stopping criterion of the procedure is met. The default is no minimum average of the internal relative flows.

`lw(`*fraction*`)` specifies the minimum weighted average of the internal relative flows. If the fraction is lower than or equal to the weighted average of the internal relative flows, the stopping criterion of the procedure is met. The default is no minimum weighted average of the internal relative flows.

`lm(`*fraction*`)` specifies the minimum internal relative flow. If the fraction is smaller than or equal to the minimum value of the internal relative flows, the stopping criterion of the procedure is met. The default is no minimum internal relative flow.

`opt_f(#)` specifies the optimization function. Four optimization functions can be chosen. The default is `opt_f(1)`, which implements the directed relative flows approach. The other functions are `opt_f(2)`, which implements the undirected relative flows approach; `opt_f(3)`, which implements the directed absolute flows approach; and `opt_f(4)`, which implements the undirected absolute flows approach.

`save_k` saves the `cluster_setk` datasets (see below). For each k, the dataset contains the absolute flows between the remaining k units (that is, the matrix $\mathbf{F}^{(K)}$ for each k). To save these datasets, specify `save_k`.

3.4 Output

`flowbca` saves three datasets.[4]

1. `cluster_set` contains variables that characterize the defined clusters, including variables that represent the cluster identifier (`clusterid`), the cluster-specific internal relative flow (`internal`), the average of the internal relative flows (`La`), the weighted average of the internal relative flows (`Lw`), the minimum of the internal relative flows (`Lm`), the cluster-specific total value of outgoing flows (`rowflows`), the total value of all flows (`N`), and a set of variables that represents the flows among all clusters (`destinationunit`).

2. `unit_set` contains variables that provide information on each starting unit, including variables that represent the source unit identifier (`sourceunit`), the cluster to which the unit is aggregated (`clusterid`), the relative flow at which the source unit is aggregated to a destination unit (`g`), the number of distinct clusters that are remaining after aggregating the source unit (`round`), and a 0/1 indicator variable that equals 1 if the unit is the core of a cluster and 0 otherwise (`core`).

3. `cluster_setk` contains the source unit identifier variable (`sourceunit`) and one variable for each destination unit representing the absolute flow (`destinationunit`). If the researcher uses the option `save_k`, then the `cluster_setk` datasets will be saved.

4. We suggest that the researcher creates a dataset that consists of the variable `sourceunit` and a variable that represents the source unit labels. This dataset could be merged to the datasets `cluster_set` and `unit_set`.

4 Examples

4.1 Example 1: Within-country regional clusters based on commuting flows

In the first example of a statistical application, a researcher uses `flowbca` to construct regional clusters based on individuals' commuting flows from municipality of residence to municipality of work. The researcher aims to compare the levels of self-containment of 40 NUTS-3 (nomenclature territorial units for statistics; in French, nomenclature des unités territoriales statistiques) areas and 12 provinces to the levels of self-containment of 40 and 12 clusters defined using `flowbca`, respectively. Note that a higher level of self-containment means there is a stronger connectivity within each regional cluster and a weaker connectivity to outside regional clusters. That is, clusters that are relatively self-contained are characterized by relatively many individuals who both live and work in the identical cluster. The Dutch NUTS-3 areas offer an interesting point of comparison, because they were defined in 1971 based on journey-to-work and place-of-work statistics that reflected the employment outcomes and commuting behavior of the Dutch population. Moreover, in research on European countries, NUTS-3 areas are often used as the regional classification to operate regional clusters (for example, Ciccone [2002]).

For this example, aggregate data on 7,131,000 commuting flows in 2014 are used at the municipality level, retrieved from the CBS StatLine open databank of Statistics Netherlands (CBS Statline 2018).[5] The algorithm starts from a set of 398 municipalities (K).[6] Note that this example uses option `k()` of `flowbca` because the researcher aims to define a specific number of clusters. The optimization function is based on the directed relative flows approach. The directed flows approach is used because commuting flows are naturally directed in that they flow from one unit to another. Relative flows are preferred to absolute flows because relative flows function as weights to account for the relative importance of a unit that allows smaller source units to be able to aggregate to bigger destination units. To visualize the defined clusters, we use the Stata commands `mergepoly` (Picard and Stepner 2012) and `spmap` (Pisati 2007).

Before we discuss the results, we illustrate the main steps of the specific code used to create figures 1a and 1b.

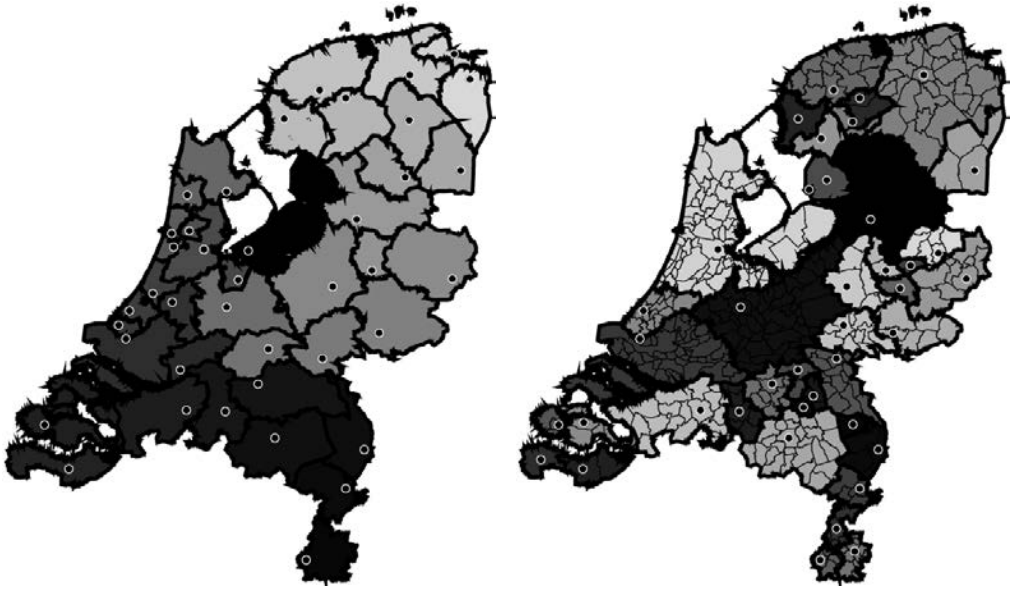
```
/* A loop is used to define 40 and 12 regional clusters
(fig. 1b and 2b, respectively) */
local numbers 40 12
local a=1
foreach numbs of local numbers {

/* Open the dataset that contains commuting flows across the Dutch
municipalities, retrieved from Statistics Netherlands CBS Statline. */
use cbs_comm_flow, clear
```

5. Note that the researcher could also exploit microdata to construct clusters. For instance, subgroup-specific clusters could be defined using subgroup-specific flows (Farmer and Fotheringham 2011).
6. Note that we remove five municipalities that represent small Wadden islands in the northern part of the Netherlands because they would be defined as small self-contained clusters that artificially increase the average of the internal relative flows (L_a).

```
/* Drop Wadden Islands that are isolated municipalities */
drop if homemun==13
drop if homemun==294
drop if homemun==322
drop if homemun==323
drop if homemun==353
drop workmun13 workmun294 workmun322 workmun323 workmun353

/* Apply flowbca. "homemun" is the source unit identifier;
"workmun*" are the destination unit identifiers */
flowbca homemun workmun*, k(`numbs´)

/* Get the summary statistics of the La, Lw, Lm variables */
display as text "Values of La, Lw, Lm for figure `a´b"
summarize La Lw Lm

/* The following syntax lines are specified to merge the cluster labels
(names of the regions) to the cluster_set dataset and the unit_set dataset
(see footnote 4 in the paper) */
rename clusterid homemun
merge 1:1 homemun using ex1label
keep if _merge==3
drop _merge
order label
compress
rename homemun clusterid
save cluster_set_fig`a´b, replace

use unit_set, clear
rename sourceunit homemun
merge 1:1 homemun using ex1label
keep if _merge==3
drop _merge
order label
compress
rename homemun sourceunit
save unit_set_fig`a´b, replace

/* Open the shape boundary database file "ex1nldb", which was generated using
the ESRI shapefile of the Netherlands */
use ex1nldb, clear

/* Drop nonexisting regional units referred to by "GM9999", and remove water
referred to by WATER=="JA" */
drop if GM_CODE=="GM9999"
drop if WATER=="JA"
rename GM_NAAM label

/* Create an identifier */
sort label
generate ID=_n
order ID
```

```
/* Merge "ex1nldb" to the codings dataset "ex1id" */
merge 1:1 ID using ex1id
drop if _merge!=3
drop _merge
rename homemun sourceunit

/* Merge the dataset to "unit_set_fig`a´b" that includes the cluster
identifier */
merge 1:1 sourceunit using "unit_set_fig`a´b"

/* Note: The Wadden Islands are deleted, because they were not included in
the clustering process */
drop if _merge!=3
drop _merge
save ex1part1, replace

/* Generate a dataset "ex1pointcoord" that contains the point coordinates of
the cores of the clusters that are returned by the algorithm */
use ex1part1, clear
keep if core==1
keep clusterid x_centroid y_centroid
save ex1pointcoord, replace

/* mergepoly: The mergepoly command is used to merge adjacent polygons from a
shape boundary file; see http://fmwww.bc.edu/RePEc/bocode/m/mergepoly.html */
/* We use the coordinate file "ex1nlcoord" to merge the polygons of the units
that are in the same cluster (given by the variable "clusterid") */
use ex1part1, clear
mergepoly id_shape using ex1nlcoord, coordinates(ex1nlcoord2) replace     ///
by(clusterid)

/* The output is the "ex1nlcoord2" dataset, which contains the coordinates
of each cluster. This dataset will be used to draw the thick border of the
cluster in the map below */

/* Now draw the map with spmap */
use ex1part1, clear
spmap clusterid using ex1nlcoord, id(id_shape) clmethod(unique)           ///
osize(thin) fcolor(Greys2) legenda(off) polygon(data("ex1nlcoord2")       ///
osize(thick ..) by(_ID)) point(data("ex1pointcoord")                      ///
x(x_centroid) y(y_centroid) by(clusterid) size(medium) ocolor(white ..))
graph export "figure`a´b_clusters.eps", replace
graph export "figure`a´b_clusters.png", replace width(5000)
local a=`a´+1
}
```

Figure 1 shows that the weighted average of the internal relative flows (L_w) of the 40 defined regional clusters (see figure 1b) is higher than the weighted average of the NUTS-3 areas (see figure 1a). This means there are more individuals who live and work in their defined regional cluster (about 80.5%) than in their NUTS-3 area (about 74.3%).

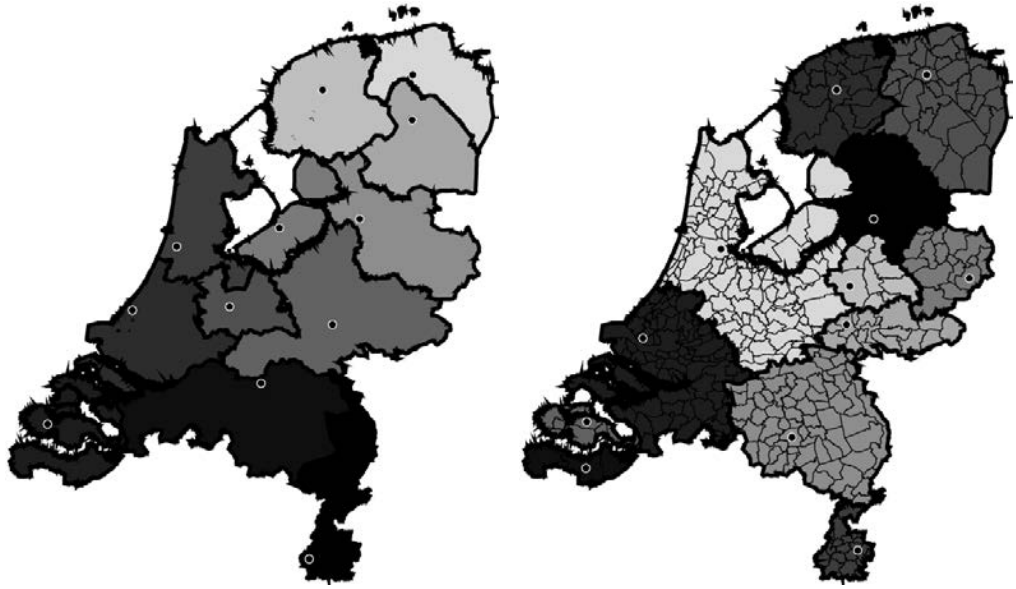


(a) NUTS-3 areas. $L_a = 0.699$; $L_w = 0.743$; $L_m = 0.462$; $K^* = 40$; $K = 40$

(b) 40 clusters. $L_a = 0.695$; $L_w = 0.805$; $L_m = 0.385$; $K^* = 40$; $K = 398$

Figure 1. NUTS-3 areas and 40 defined clusters. Notes: A commuting flow registers the number of workers who commute from a municipality of residence to a municipality of work. The NUTS-3 cores (black dots with a white circle) are defined as the municipality with the highest number of residents. The algorithm returns the cores of the defined regional clusters. Each distinct cluster is surrounded by a thick border and highlighted by a different shade of gray. Note that the shade of a cluster does not provide any further information. The algorithm returns L_a, L_w, and L_m, which refer to the average of the internal relative flows (1), the population-weighted average of the internal relative flows (2), and the minimum of the internal relative flows (3), respectively. K^* and K refer to the number of defined regional clusters and the number of distinct starting units, respectively.

Figure 2 shows that the average of the internal relative flows (L_a), the weighted average of the internal relative flows (L_w), and the minimum of the internal relative flows (L_m) are higher in the case of the 12 defined clusters than of the 12 predefined administrative provincial areas. All in all, this example shows that `flowbca` can be used to define meaningful regional clusters that are characterized by a relatively high level of self-containment.



(a) Provincial areas. $L_a = 0.833$; $L_w = 0.866$; $L_m = 0.570$; $K^* = 12$; $K = 12$

(b) Twelve clusters. $L_a = 0.869$; $L_w = 0.901$; $L_m = 0.776$; $K^* = 12$; $K = 398$

Figure 2. Provinces and 12 defined clusters. Notes: A commuting flow registers the number of workers who commute from a municipality of residence to a municipality of work. The provincial cores (black dots with a white circle) are the capital cities. The algorithm returns the cores of the defined regional clusters. Each distinct cluster is surrounded by a thick border and highlighted by a different shade of gray. Note that the shade of a cluster does not provide any further information. The algorithm returns L_a, L_w, and L_m, which refer to the average of the internal relative flows (1), the population-weighted average of the internal relative flows (2), and the minimum of the internal relative flows (3), respectively. K^* and K refer to the number of defined regional clusters and the number of distinct starting units, respectively.

4.2 Example 2: Global regional clusters based on national trade flows

In the second example, a researcher uses `flowbca` to construct global regional clusters based on trade flows that are defined as the size of the annual trade from an exporting to an importing country. The researcher aims to examine how interrelated countries are in terms of trade and whether this relatedness changed over time. The researcher defines global clusters of economic activity for the years 1995 and 2011, given a minimum flow threshold equal to 5%.

For this example, we use the World Input–Output Database (WIOD) (Dietzenbacher et al. 2013). This dataset consists of data on the trade flows between 40 countries in the period 1995–2011. The algorithm starts from a set of 40 countries (K). Note that this exercise uses the option `q()` of `flowbca`. The optimization function is based on the directed relative flows approach.

We compare the set of clusters in 1995 and 2011 to examine the change in global trade clusters over time. Figure 3 shows, given the flow threshold of 5%, that the number of distinct global clusters (K^*) decreases from 19 to 10 over the period 1995–2011. Consistent with globalization, the decrease in the number of defined clusters suggests that the trade flows within countries decreased relative to the trade flows between countries. Another observation is that the size of the two clusters in which China and Germany is the core, respectively, became larger over time.

(a) Global trade clusters in 1995. $L_a = 0.920$; $L_w = 0.943$; $L_m = 0.854$; $K^* = 19$; $K = 40$

(b) Global trade clusters in 2011. $L_a = 0.938$; $L_w = 0.947$; $L_m = 0.898$; $K^* = 10$; $K = 40$

Figure 3. Global clusters based on trade flows. Notes: A trade flow registers the size of annual trade from an exporting country to an importing country. The algorithm returns the cores of the defined clusters (black dots with a white circle). Each distinct cluster is highlighted by a different shade of gray. Note that the shade of a cluster does not provide any further information. Global clusters were defined based on trade flows expressed in millions of dollars between countries from the 1995 and 2011 WIOD data. The flow threshold `q` was equal to 5%. Trade flows between countries were computed by aggregating all within-country flows. Nine countries with zero or negative flows were removed. The algorithm returns L_a, L_w, and L_m, which refer to the average of the internal relative flows (1), the population-weighted average of the internal relative flows (2), and the minimum of the internal relative flows (3), respectively. K^* and K refer to the number of defined global clusters and the number of starting countries, respectively.

4.3 Example 3: A social network based on friendship ties

In the third example, a researcher uses `flowbca` to detect groups of prison inmates based on friendship ties. A friendship tie could be considered as a binary flow variable from one inmate to another, which is one in case of a friendship. If two inmates indicate a friendship with each other, the ties will flow in both directions. The researcher aims to detect groups of inmates in which each group should have a minimum internal relative flow of at least 50%. The minimum internal relative flow of 50% means that, in each group, at least 50% of the inmates' friendship ties should be with inmates in their own group.

For this example, the Gagnon and MacRae prison friendship dataset was used (MacRae 1960). The level of interaction between inmates is approximated by multiple 0/1 indicator variables that represent friendship ties, which equal 1 if a given "source" inmate indicates a friendship with a given "destination" inmate and 0 otherwise. The algorithm starts from a set of 67 inmates (K). All inmates could indicate as few or as many friendship ties as desired. Inmates could not indicate a friendship with themselves. Note that this example uses the option `lm()` of `flowbca`. Relative flows were used to have the relative importance of each tie.

Before we discuss the results, we illustrate the main steps of the specific code used to construct table 1.

```
/* Directed relative flows approach */
use ex3_prison, clear

/* Apply flowbca */
flowbca sourceunit destinationunit*, lm(.5) opt_f(1)

/* The option lm(fraction) is used to specify the minimum internal relative
flow. If the fraction is smaller than or equal to the minimum value, the
algorithm is terminated */
/* The option opt_f() is specified to use the directed relative flows
approach */

/* Drop the inmate (number 35) who is isolated */
drop if internal==.
drop destinationunit35

/* Get the summary statistics of the La, Lw, Lm variables */
summarize La Lw Lm
save cluster_set_lm05f1_table1, replace

use unit_set, clear

/* Generate the number of inmates in each cluster */
bysort clusterid: gen n=_N
tabulate n if core==1 & n!=1

/* Undirected relative flows approach */
use ex3_prison, clear

/* Apply flowbca */
flowbca sourceunit destinationunit*, lm(.5) opt_f(2)

/* The option lm(fraction) is used to specify the minimum internal relative
flow. If the fraction is smaller than or equal to the minimum value, the
algorithm is terminated */
/* The option opt_f() is specified to use the undirected relative flows
approach */

/* Drop the inmates (numbers 19, 25, 26, and 35) who are isolated */
drop if internal==.
drop destinationunit19 destinationunit25 destinationunit26 destinationunit35

/* Get the summary statistics of the La, Lw, Lm variables */
summarize La Lw Lm
save cluster_set_lm05f2_table1, replace

use unit_set, clear

/* Generate the number of inmates in each cluster */
bysort clusterid: gen n=_N
tabulate n if core==1 & n!=1
```

Table 1 provides information about the groups of inmates that were detected using `flowbca` for both the directed and undirected flows approach, respectively. The results show that the directed flows approach leads to fewer and bigger groups of inmates compared with the undirected flows approach. Another observation is that there are more isolated inmates if the optimization function is based on the undirected flows approach. Effectively, the undirected flows approach puts more weight on the situation where two inmates indicate each other as friend and leads to more sparse groups.

Table 1. Number and size of the detected groups of inmates

Directed flows approach		Undirected flows approach	
Number of groups for a given size	Size (in # of inmates)	Number of groups for a given size	Size (in # of inmates)
1	38	1	11
1	12	1	9
1	7	1	7
1	5	1	6
1	4	3	5
		2	4
		1	3
		2	2
$L_a = 0.847$		$L_a = 0.689$	
$L_w = 0.846$		$L_w = 0.676$	
$L_m = 0.758$		$L_m = 0.5$	
$K^* = 5$		$K^* = 12$	
$K = 67$		$K = 67$	

NOTE: The connected groups of inmates are based on friendship ties between inmates from the Gagnon and MacRae prison dataset. A friendship tie registers a friendship as a flow from one inmate to another. The minimum internal relative flow, `lm`, which each group should satisfy, was set equal to 50%. The raw prison dataset contains 67 inmates. No inmate was disconnected (that is, the situation where an inmate did not specify or was specified as a friend by another inmate). However, one inmate and four inmates were detected as isolated using the directed and undirected flows approach, respectively. The isolated inmates were removed. The algorithm returns L_a, L_w, and L_m, which refer to the average of the internal relative flows (1), the population-weighted average of the internal relative flows (2), and the minimum of the internal relative flows (3), respectively. K^* and K refer to the number of defined regional clusters and the number of distinct starting units, respectively.

4.4 Example 4: Industrial clusters based on input–output flows

The final example is a case where a researcher uses `flowbca` to define five U.S. industrial clusters based on input–output flows of goods between U.S. industries. A flow of goods registers the size of the goods delivered by the industry of input to the industry of output. The researcher examines whether the relatedness between industries changed over time by comparing the set of industrial clusters in 1995 with the set of clusters in 2011. The WIOD was used to define U.S. industrial clusters in the years 1995 and 2011. The algorithm starts from a set of 35 industries (K). Note that this exercise

uses option `k()` of `flowbca`. The optimization function is based on the directed relative flows approach.

Table 2 presents the results of example 4. The sectors "Construction" and "Public Administration and Defence; Compulsory Social Security" are the largest industrial clusters in 1995 and 2011, respectively. Hence, in 2011, the industry "Public Administration and Defence; Compulsory Social Security" uses relatively more goods that are produced in other production chains than the industry "Construction". Interestingly, `flowbca` defines the industries "Food, Beverages and Tobacco", "Textiles and Textile Products", and "Transport Equipment" as the core of a cluster in both 1995 and 2011, which suggests that these clusters have been relatively self-contained over time.

Table 2. Industrial clusters based on U.S. input–output flows

1995		2011	
Core of cluster	Size (in # of units)	Core of cluster	Size (in # of units)
Construction	28	Public Administration and Defence; Compulsory Social Security	30
Food, Beverages and Tobacco	3	Food, Beverages and Tobacco	2
Textiles and Textile Products	2	Chemicals and Chemical Products	1
Chemicals and Chemical Products	1	Basic Metals and Fabricated Metal	1
Transport Equipment	1	Transport Equipment	1
$L_a = 0.602$		$L_a = 0.567$	
$L_w = 0.828$		$L_w = 0.837$	
$L_m = 0.335$		$L_m = 0.287$	
$K^* = 5$		$K^* = 5$	
$K = 35$		$K = 35$	

NOTE: U.S. industrial clusters based on U.S. input–output flows of goods expressed in millions of dollars between 35 ISIC industries from the WIOD data. The minimum number of clusters `k()` was set equal to five. The algorithm returns L_a, L_w, and L_m, which refer to the average of the internal relative flows (1), the population-weighted average of the internal relative flows (2), and the minimum of the internal relative flows (3), respectively. K^* and K refer to the number of defined regional clusters and the number of distinct starting units, respectively.

As table 2 shows, the largest cluster is composed of many units and the other clusters are composed of few units. Example 4 highlights the main limitation of `flowbca`, which is that the algorithm defines one relatively big cluster that is composed of many units, if the network is not sparse enough and thus not composed of multiple subnetworks. For example, consider the case that a researcher aims to define clusters using random flows between units. It is likely that in each iteration, a source unit is aggregated to the identical destination unit; the destination unit represents a relatively big cluster because of the aggregations in the earlier iterations.

All in all, to define meaningful clusters, the network should be sparse enough, for example, by distance (in the case of within-country regional clusters), input–output flows of goods (in the case of industrial clusters), or social interaction (in the case of social network analysis). Otherwise, the algorithm will define one relatively big cluster and several distinct clusters of few units. Note that the use of more disaggregated units in the starting set of units would improve the accuracy of the cluster algorithm because more detailed flow data are used.

5 Concluding remarks

In this article, we have introduced and illustrated the `flowbca` command, which can be used to define clusters based on relational data of flows between disjoint units. We provided four examples of statistical applications in a wide range of research fields to illustrate `flowbca` cluster identification capabilities. Given the increasing availability of relational data of various types of flows, `flowbca` can be used in a variety of research fields. `flowbca` is flexible because it allows for various optimization functions and stopping criteria. The command is accessible for the researcher, which will hopefully lead to a further development of the algorithm. Overall, the command is robust, user friendly, and well able to define clusters that are characterized by a high level of self-containment.

6 Acknowledgments

We thank an anonymous reviewer whose valuable comments improved the quality of the paper. In addition, we are grateful for the comments of seminar participants at the Utrecht University School of Economics. We also thank Rense Corten, Elena Fumagalli, and Bastian Westbrock for insightful comments.

7 References

Bathelt, H., and P.-F. Li. 2014. Global cluster networks—foreign direct investment flows from Canada to China. *Journal of Economic Geography* 14: 45–71.

Beguerisse-Díaz, M., G. Garduño-Hernández, B. Vangelov, S. N. Yaliraki, and M. Barahona. 2014. Interest communities and flow roles in directed networks: The Twitter network of the UK riots. *Journal of the Royal Society Interface* 11. http://rsif.royalsocietypublishing.org/content/11/101/20140940.

Brezzi, M., M. Piacentini, K. Rosina, and D. Sanchez-Serra. 2012. Redefining urban areas in OECD countries. In *Redefining "Urban": A New Way to Measure Metropolitan Areas*, 19–58. Paris: OECD Publishing.

CBS Statline. 2018. CBS Open Data StatLine. https://opendata.cbs.nl/statline/.

Chen, M. X., and M. O. Moore. 2010. Location decision of heterogeneous multinational firms. *Journal of International Economics* 80: 188–199.

Ciccone, A. 2002. Agglomeration effects in Europe. *European Economic Review* 46: 213–227.

Coombes, M. G., A. E. Green, and S. Openshaw. 1986. An efficient algorithm to generate official statistical reporting areas: The case of the 1984 travel-to-work areas revision in Britain. *Journal of the Operational Research Society* 37: 943–953.

Corten, R. 2011. Visualization of social networks in Stata using multidimensional scaling. *Stata Journal* 11: 52–63.

Dietzenbacher, E., B. Los, R. Stehrer, M. Timmer, and G. J. de Vries. 2013. The construction of world input–output tables in the WIOD project. *Economic Systems Research* 25: 71–98.

Duranton, G. 2015. Delineating metropolitan areas: Measuring spatial labour market networks through commuting patterns. In *The Economics of Interfirm Networks*, ed. T. Watanabe, I. Uesugi, and A. Ono, 107–133. Japan: Springer.

Farmer, C. J. Q., and A. S. Fotheringham. 2011. Network-based functional regions. *Environment and Planning A: Economy and Space* 43: 2723–2741.

Ford, L. R., Jr., and D. R. Fulkerson. 1962. *Flows in Networks*. Princeton, NJ: Princeton University Press.

Fortunato, S. 2010. Community detection in graphs. *Physics Reports* 486: 75–174.

Leontief, W. 1986. *Input–Output Economics*. 2nd ed. New York: Oxford University Press.

MacRae, D., Jr. 1960. Direct factor analysis of sociometric data. *Sociometry* 23: 360–371.

Meekes, J. 2019. Local labour markets, job displacement and agglomeration economies. PhD thesis, Utrecht University, Utrecht, Netherlands.

Miura, H. 2012. Stata graph library for network analysis. *Stata Journal* 12: 94–129.

Picard, R., and M. Stepner. 2012. mergepoly: Stata module to merge adjacent polygons from a shapefile. Statistical Software Components S457574, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s457574.html.

Pisati, M. 2007. spmap: Stata module to visualize spatial data. Statistical Software Components S456812, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s456812.html.

Smith, D. A., and D. R. White. 1992. Structure and dynamics of the global economy: Network analysis of international trade 1965–1980. *Social Forces* 70: 857–893.

Timmer, M. P., E. Dietzenbacher, B. Los, R. Stehrer, and G. J. de Vries. 2015. An illustrated user guide to the world input–output database: The case of global automotive production. *Review of International Economics* 23: 575–605.

About the authors

Jordy Meekes was a doctoral candidate in the Department of Economics at Utrecht University, Utrecht, the Netherlands. He is a research fellow of the Melbourne Institute of Applied Economic and Social Research at the University of Melbourne, Melbourne, Australia, and a research affiliate of the IZA, Bonn, Germany.

Wolter H. J. Hassink is a professor of applied econometrics at Utrecht University, Utrecht University School of Economics, Utrecht, the Netherlands, and a research fellow of the IZA, Bonn, Germany.