# Multi Source Data Linkage & Clustering

We agree for MSL there are 4 parts of the process (blocking, feature generation/comparison metrics, model building, clustering). We agreed that the less-attended parts are blocking and clustering, hence focus.

Shelley Gammon, Patrick Ball, Matt Barnes, Christos Chatzoglou, Mark Eliot, Paul Groom, Mauricio Sadinle, Beka Steorts, Dinusha Vatsalan, Bill Winkler, Cong Chen, Willie Neiswanger, Duncan Smith

?Yan-Xia Lin, James Boyd

## Blocking

### Literature repository

Mark & Duncan have a solid start

And for a lightweight perspective: PB blog

Michelson and Knoblock

Bilenko, M.

Dinusha’s paper on blocking “scalable block scheduling for efficient multi-db record linkage”

Kenig and Gal (non-field based blocking approach based on maximal frequent itemsets selection): <http://www.sciencedirect.com/science/article/pii/S0306437912001500>

### Needs

* Do you or do you not have training data; or how would you define it, how to sample to get the best training data
* Generalizability, don’t overfit (how to test?)
* What are blocking features or approaches? Field/feature based, or unsupervised (give us a word, Beka)
* Multistage?
* Performance runtime and computational load
* Send list of what you need to achieve w blocking (e.g., maximum pairs completeness, ~ 99.99%; RR ~ 1000x)

### Output – Bill & Shelley

* Good reading list
* Summary w recommendations based on balance of needs

## Clustering

Shelley is rapporteur

Literature repository

Concept review

* Graph theoretic ideas about connected components, edge-betweenness, modularity, cluster diameter
* Basic methods include transitive closure, hierarchical agglomerative clustering (HAC)

Simple case: 2 lists, no duplication w/in lists, maximum cluster size is 2

* K lists, no dup win lists, max cluster size is k
* K lists, N recs total, duplication win lists, max cluster size is N, max number of clusters is also N

Actions

* Generate data (Beka and Dinusha), data means a blocked similarity matrix, min six lists, 100-400K total recs, E(overlap) = ? , and true cluster labels for recs
* Each participant will apply approach and report estimated labels + runtime + CODE
* Who will benchmark runtimes? i.e., run in controlled environment to calc comparable runtimes + memory + diskspace

Participant ideas

* PB’s minimum conn comp+HAC (avg)
* Christos’s graph approach using graph theoretic metrics
* Matt: moment matching
* Mark & Duncan: transitive closure + fix the number of clusters (HAC?) + kmeans to optimize structure of clusters
* Beka & Mauricio: Bayesian approaches
* Dinusha incremental clustering: dedupe each list, then Hungarian Alg across lists or at final
* James

Output