Sequence Clustering Tools

[Internal Report]

Saliya Ekanayake
School of Informatics and Computing
Indiana University
sekanaya@cs.indiana.edu

1. Introduction

The sequence clustering work carried out by SALSA group in Indiana University assists biologists by identifying similarities present in sequences and classifying them accordingly. At a higher level, the breakdown of tasks involves *pairwise sequence alignment → pairwise clustering → multidimensional scaling*, where the latter two may be done in parallel to each other. Pairwise sequence alignment computes a dissimilarity value for each pair of sequences based on their alignment. Pairwise clustering performs classification of sequences based on these dissimilarities producing a mapping of sequences into groups. Multidimensional scaling works on the same dissimilarity values and maps each sequence into a point in three dimensions such that the Euclidean distance between any two points corresponds to the dissimilarity between particular two sequences. Each step being data and compute intensive requires parallel algorithms to produce results in reasonable amounts of time when run on multiple computers. We also present a tool, which helps with job creation, submission, and monitoring.

2. Algorithms

2.1 Pairwise Sequence Alignment

Table 1 classifies our collection of sequence alignment implementations.

Table 1. Collection of sequence alignment implementations

Name	Algorithms	Alignment Type	Language	Library	Parallelization	Target Environment
SALSA-SWG	Smith-Waterman (Gotoh)	Local	C#	None	Message Passing with MPI.NET	Windows HPC cluster
SALSA-SWG-MBF	Smith-Waterman (Gotoh)	Local	C#	.NET Bio (formerly MBF) [1]	Message Passing with MPI.NET	Windows HPC cluster
SALSA-NW-MBF	Needleman-Wunsch (Gotoh)	Global	C#	.NET Bio (formerly MBF) [1]	Message Passing with MPI.NET	Windows HPC cluster
SALSA-SWG-MBF2Java	Smith-Waterman (Gotoh)	Local	Java	None	Map Reduce with Twister [2]	Cloud / Linux cluster
SALSA-NW-BioJava	Needleman-Wunsch (Gotoh)	Global	Java	BioJava [3]	Map Reduce with Twister [2]	Cloud / Linux cluster

SALSA-SWG is the earliest in-house implementation used for SALSA group's sequence alignment projects. Later we adopted the open source .NET Bio (formerly Microsoft Biology Foundation – MBF) library for the core implementation of Smith-Waterman (SW) and Needleman-Wunsch (NW) algorithms for convenience and performance reasons.

2.1.1 Optimizations

The Java implementation of local alignment is a product of SALSA group, which is efficient than .NET Bio version, yet it closely adheres to the implementation from .NET Bio. In particular, we have made the following optimizations in the Java version resulting performance improvement by factor of 2.

- Avoid sequence validation when aligning
  - Sequence validation means to check if sequences conform to the alphabet in the given scoring matrix. We perform this ahead of time for the given N sequences, thus avoiding the N^2 validations inside all-pairs alignment computation.
- Avoid alphabet guessing
  - .NET Bio loops through a known set of alphabets to guess the corresponding alphabet to a given sequence. We omit this as we know the alphabet for the sequences at start.
- Avoid nested data structures

- The .NET Bio implementation wraps the result of an alignment in a nested data structure for generalization purposes. Performance and memory profiling with dotTrace [4] revealed it as a performance "hot-spot", hence we package results in a simple data structure.

SALSA-NW-BioJava implements global alignment with Java and depends on the open source BioJava library. However, we have made the following changes to BioJava module.

- Produce consistent alignment results compared to C# versions based on .NET Bio.
  - The inconsistencies arise due to correct yet alternative decisions that could be taken when aligning two sequences. For example deciding whether to align the i^{th} base in one sequence with the j^{th} base of the other, insert a gap, or delete a gap depends on the score of each option. It is possible for one or more of these alternatives to yield the same score yet produce different alignments depending on the choice made by the implementation.
- Improve substitution matrix access time
  - Substitution matrix contains a score for each possible pair of bases in the alphabet of sequences. BioJava stores it as a two dimensional array and keeps a mapping table from base to array index. This two-step mapping hinders performance specially when aligning long sequences. Therefore, we change it to store the substitution matrix indexed by the character value of bases to get an *O(1)* lookup.

The C# versions use message passing with MPI.NET [5] to parallelize the all-to-all sequence alignment whereas the Java versions use MapReduce with Twister framework. Irrespective of the technology, the parallelization follows Single Program Multiple Data (SPMD) style where a process/task computes the all-to-all alignments for a block of sequences at a time. The resulting blocks are merged to form the full result at the end. Also, we assume that alignment is independent of the order of two sequences, which simplifies the problem space by half.

2.2 Deterministic Annealing Pairwise Clustering (DA-PWC)

DA-PWC adopts concept of deterministic annealing for clustering [6] and implements a scalable version suitable for large scale data mining, which runs in *O(NlogN)* time compared to existing $O(N^2)$ implementations [7]. Given *N* data points, DA-PWC accepts an *N x N* pairwise distance matrix to perform clustering. The end result is a mapping from points to group numbers they belong, where the number of groups assumed known ahead of time. Visual aid from multidimensional scaling could be used to refine the number of clusters as discussed in [section 2.3] . DA-PWC produces cluster centers as well based on the smallest mean distances, i.e. the point with the smallest mean distance to all other points in a given cluster. If provided with a coordinate mapping for each point it could also produce centers based on smallest mean Euclidean distance and Euclidean center. The implementation is based on C# language and MPI.NET is used for parallelization targeting Windows HPC cluster environments.

2.3 Multidimensional Scaling

The idea of multidimensional scaling is to map points in higher dimensions to lower dimensions like 3 or 2 while preserving inter-point distances. We have three implementations as in [table] which operate on pairwise distances between points and produce a three dimensional coordinate mapping for them.

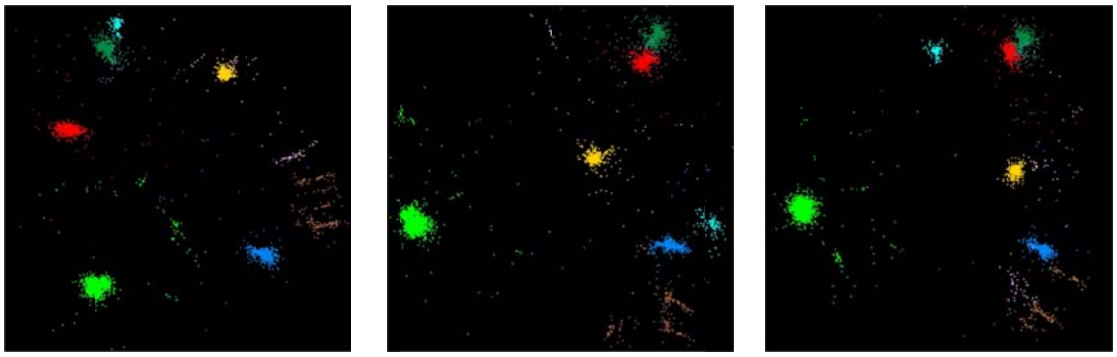
Name	Optimizes	Optimization Method	Language	Parallelization	Target Environment
MDSasChisq	General MDS with arbitrary weights and missing distances and fixed positions	Levenberg–Marquardt algorithm	C#	Message Passing with MPI.NET	Windows HPC cluster
DA-SMACOF	$\sigma(X) = \sum_{i<j\leq N} w_{ij}\,(d_{ij}(X) - \delta_{ij})^2$	Deterministic annealing	C#	Message Passing with MPI.NET	Windows HPC cluster
Twister DA-SMACOF	$\sigma(X) = \sum_{i<j\leq N} w_{ij}\,(d_{ij}(X) - \delta_{ij})^2$	Deterministic annealing	Java	Map Reduce with Twister	Cloud / Linux cluster

In addition to performing dimensional scaling, MDSasChisq also supports a complementary set of functions as given below.

- Transform input distances

- Input distances in generally lie on higher dimensions and it is often useful to transform them into a lower dimension prior to dimensional scaling to yield an accurate mapping of points into coordinates.

- Fixed point runs
  - If an already computed lower dimensional mapping is given for a subset of points then MDSasChisq will preserve it and will map the rest of the points around them.
- Result rotation
  - The coordinate mapping of dimensional scaling may not be identical even for the same dataset since the goal is to preserve inter-point distances but not the positions. However, it is useful to "align" results of same or similar datasets for the sake of side-by-side comparison. Result alignment may include rotation, inversion, or both. The center image of Figure 1 shows a mapping of a reference dataset. On its left is the mapping of the same data based on a different distance measure. The right most one is the "aligned" version of left with center, which appears similar to the reference than the "unaligned" left one.



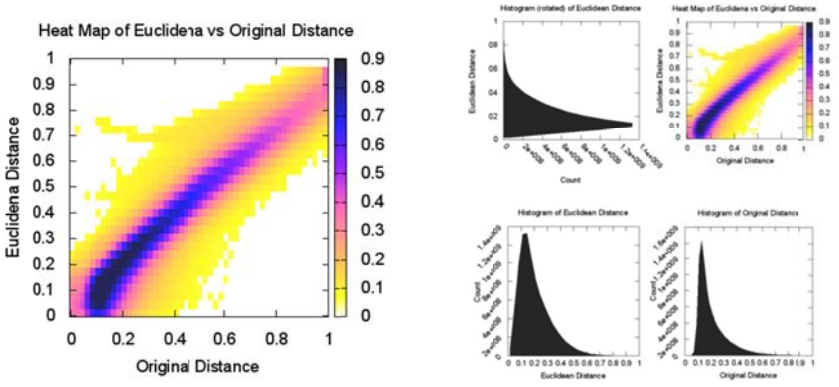
(a) Mapping of (b) with difference distance measure

(b) Reference mapping

(c) Rotation of (a) to "align" with (b)

Figure 1. Result rotation of MDSasChisq

- Heat map and histogram generation
  - The correlation between mapped Euclidean distances and input distances is a measure to determine to the quality of dimensional scaling. Scatter plots are a good way to infer the correlation, but as the plot area gets saturated with points they fail to represent correlation clearly. Heat maps are a better alternative in this regard which present densities of saturated areas making it easier to identify any correlation if present. Figure 2 shows a sample heat map on the left and the four-way view of heat map and distance histograms on the right.



(a) Sample heat map

(b) Four-way heat map and histograms

Figure 2. Heat map and histograms of MDSasChisq

3. Tryout Tool

Tryout is a .NET based desktop application designed to create, submit, and monitor DA-PWC and MDSasChisq jobs in Windows HPC cluster environments. Figure 3 show the main interface of it.

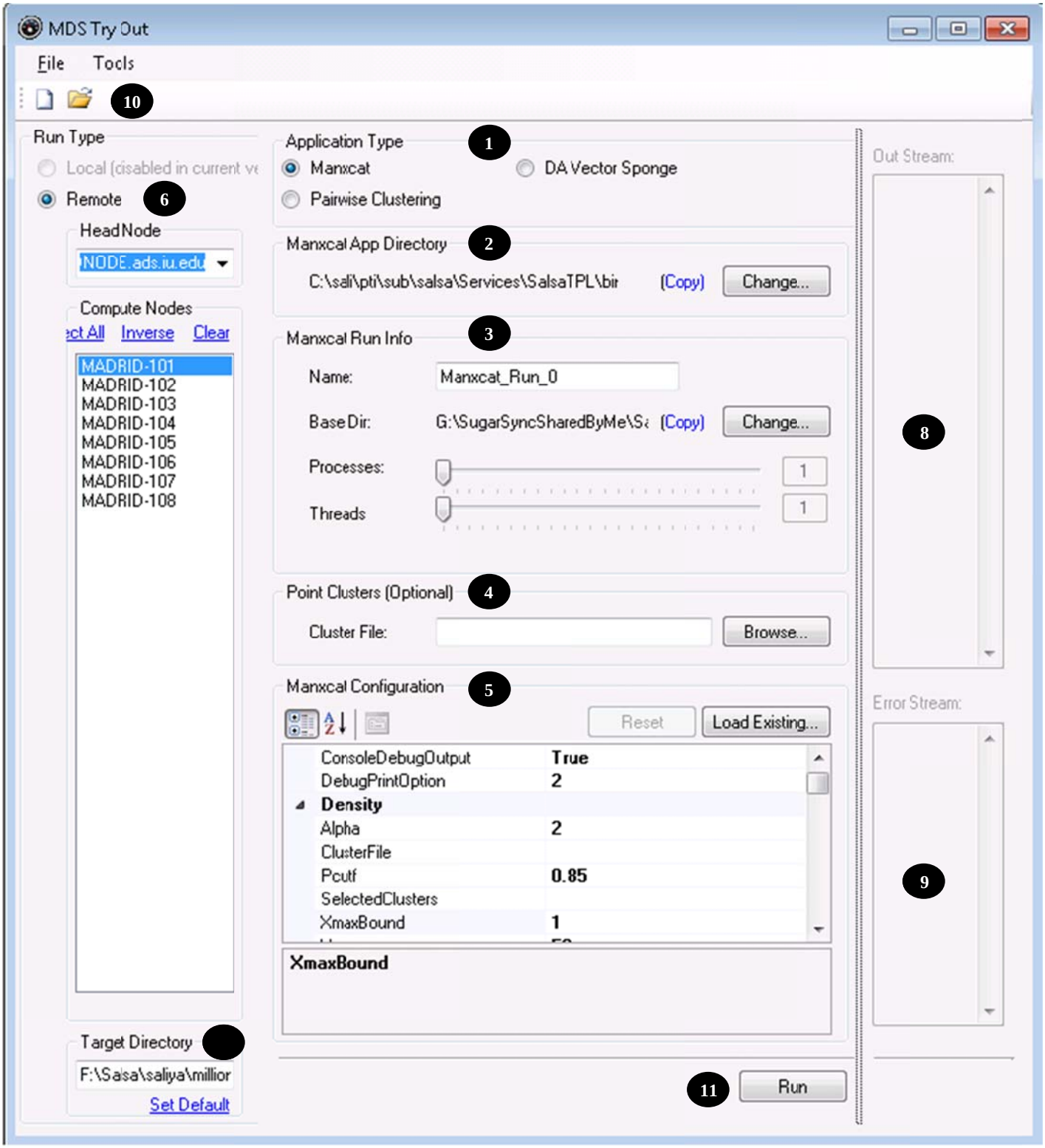


Figure 3. Main interface of Tryout tool

The description of elements in Figure 3 is as follows.

1. Application Type
    - This gives the option to select the type of application as MDSasChisq or DA-PWC. Also in the figure is the option for another application, i.e. DA Vector Sponge, we are currently developing.
2. App Directory
    - This is the directory where executables of the particular application type could be found. It is possible to set a default application directory for each application type.
3. Run Info
    - This section lets the user to specify name for the run, a local directory to create it, and the level of parallelism.
4. Optional File
    - MDSasChisq results could be combined with clustering information to visualize using PlotViz, which is a 3D data visualization tool we have developed [8]. Similarly DA-PWC results could be combined with a coordinate mapping from MDSasChisq (or other dimensional reduction program) to use with PlotViz. Therefore, depending on the application type the user can specify a corresponding file to supplement visualization.
5. Configuration
    - Each application is configurable via an XML configuration file. This section presents it in an easy-to-read tabular style. It is also possible to load an existing configuration from file using the "Load Existing" button. Description of parameters for MDSasChisq and DA-PWC is given in.
6. Cluster Selection
    - This shows the available Microsoft HPC clusters and the corresponding set of compute nodes. The user can select the desired number of nodes for the particular run.
7. Target Directory
    - A directory for the specific run is created under this in the remote cluster.
8. Output Stream
    - Once the job is submitted, any information written to standard output displayed here.
9. Error Stream
    - Similar to the output stream, errors are reported in this area.
10. New/Open Menus
    - New run option will clear the user interface and set it to default state letting the user to choose correct parameters of the desired run. Once a run is submitted to the cluster its information is saved in an XML file making it possible to later open it using the tool.
11. Run Submission
    - Once the desired parameters are set for the particular run the user can submit it using the "Run" button. If a run exists with the same name in the same directory the tool will prompt for confirmation of resubmission to avoid overwriting existing results.

In addition to the above, the following are also available in Tryout.

- Abort Run
    - Once a run is submitted the user has the option to abort it if necessary
- Show in PlotViz
    - The results of MDSasChisq can be directly visualized using PlotViz and Tryout has built-in support to invoke PlotViz with the results of particular dimensional scaling run. Also, for DA-PWC runs if a coordinate file is given in configuration it could be visualized in PlotViz with clusters colored according to the results of clustering run.
- Known Files
    - Input files required for a run are copied from local computer to the cluster by default. However, if the file is already in the cluster the user can add a mapping to that by giving a name and location in the cluster. The Tryout tool reads in these mappings from a special file at startup and the mapped resources can be referred using the given name prefixed by dollar ($) sign.

4. Summary

We assist biologists in determining similar groups of sequences and visualizing their relationship in 3D with the use of new algorithms for clustering and multi-dimensional scaling. The tool-chain we have developed includes multiple implementations of pairwise sequence alignment, clustering and dimension reduction algorithms. We also have supporting software to handle job submission to high performance clusters, and data visualization.

5. References

[1] Benge, R. *.NET Bio*. City.

[2] Ekanayake, J., Li, H., Zhang, B., Gunarathne, T., Bae, S.-H., Qiu, J. and Fox, G. Twister: a runtime for iterative MapReduce. In *Proceedings of the Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing* (Chicago, Illinois, 2010). ACM, [insert City of Publication],[insert 2010 of Publication].

[3] Prlic, A., Yates, A., Bliven, S. E., Rose, P. W., Jacobsen, J., Troshin, P. V., Chapman, M., Gao, J., Koh, C. H., Foisy, S., Holland, R., Rimsa, G., Heuer, M. L., Brandstatter-Muller, H., Bourne, P. E. and Willis, S. BioJava: an open-source framework for bioinformatics in 2012. *Bioinformatics*, 28, 20 (Oct 15 2012), 2693-2695.

[4] JetBRAINS dotTrace.

[5] Gregor, D. and Lumsdaine, A. Design and implementation of a high-performance MPI for C\# and the common language infrastructure. In *Proceedings of the Proceedings of the 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming* (Salt Lake City, UT, USA, 2008). ACM, [insert City of Publication],[insert 2008 of Publication].

[6] Rose, K., Gurewwitz, E. and Fox, G. A deterministic annealing approach to clustering. *Pattern Recogn. Lett.*, 11, 9 1990), 589-594.

[7] Fox, G. C. Deterministic annealing and robust scalable data mining for the data deluge. In *Proceedings of the Proceedings of the 2nd international workshop on Petascal data analytics: challenges and opportunities* (Seattle, Washington, USA, 2011). ACM, [insert City of Publication],[insert 2011 of Publication].

[8] Choi, J. Y., Bae, S.-H., Qiu, J., Fox, G., Chen, B. and Wild, D. Browsing large scale cheminformatics data with dimension reduction. In *Proceedings of the Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing* (Chicago, Illinois, 2010). ACM, [insert City of Publication],[insert 2010 of Publication].