

Structuring Genomic Data in GEO Using Agglomerative Clustering

Tim Vasil
6.873 Final Project
December 2, 2009

Abstract

The majority of genomic data in the Gene Expression Omnibus (GEO) is not structured, making it difficult for researchers to use this data. We offer an adapted agglomerative clustering technique based on sample names to give structure to this data, and name the resultant clusters based on both itemset mining and acronym expansion. To visualize results, we use a proxy server implementation to “inject” the structured results into live GEO website results. We find these clusters look intuitively useful upon visual inspection, yet when compared with pre-computed clusters where available, we find the Rand index is typically below 0.5, indicating relative disagreement.

1. Background

Researchers interested in submitting papers to biomedical journals must first submit the relevant genomic data to a public repository, such as the Gene Expression Omnibus (GEO) [1], maintained by the National Institutes of Health under the U.S. Department of Health and Human Services. The intent of this policy is to make it easier for colleagues to access and utilize this data. While GEO intentionally does not impose tight restrictions regarding the structure of this data so as to be responsive to developing trends [2], the unfortunate result is unstructured data that works against the stated goal of the repository. That is, the lack of structure makes it *harder* for others to find and utilize data of interest to them.

GEO staff has taken steps to address this problem by manually translating researcher-submitted data “GEO Series” into “GEO DataSets.” Once in DataSet format, additional tools become available on the website, such as cluster diagrams and gene expression profile charts. These tools give much-needed structure to the data, however the majority of Series are not available in DataSet format. There are two reasons for this: 1) GEO staff is running a backlog on Series conversions, and 2) not all Series can be translated into DataSets [2].

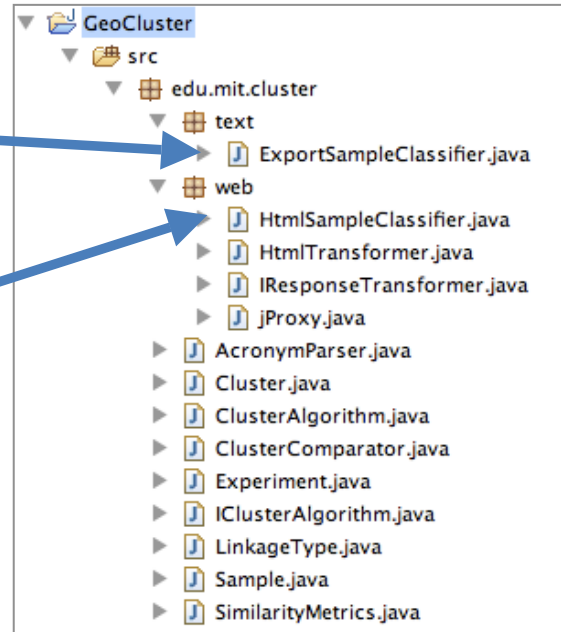
To address the lack of structure to Series submissions without corresponding DataSet translations (>85% of submissions), we devise a technique whereby structure can be placed on Series data without any manual effort. Specifically, we use agglomerative hierarchical clustering to group samples in a Series based on the sample names alone, entitle these clusters with intuitive names using itemset techniques and an understanding of acronym definitions, and mesh the results on-the-fly with the GEO website user interface using an HTTP proxy server. The resulting website gives GEO users the best of both worlds: DataSet data where available, as well as automatically-generated clusters across all Samples, increasing utility to researchers and collaborators through improved structure across the GEO repository.

2. Materials and Methods

Our agglomerative clustering algorithm, with several enhancements described below, is implemented as a Java application we called *GeoCluster*. Source code is included in the appendix and also available online [3]. We selected version 1.6 of the Java Virtual Machine (JVM) runtime and Eclipse version 1.4.2 as our integrated development environment (IDE).

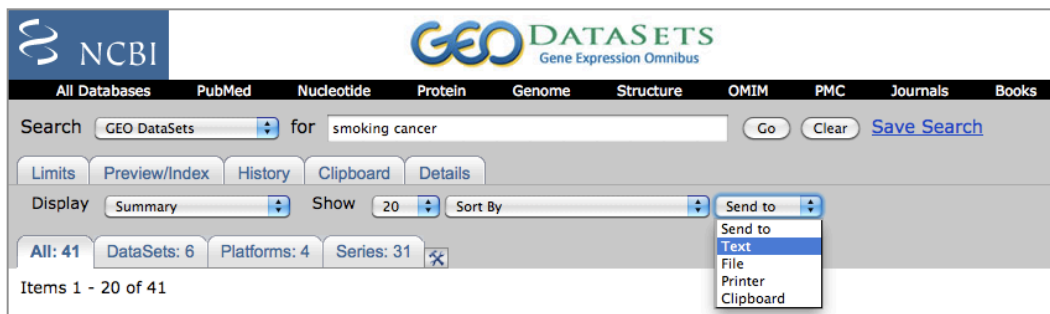
The application provides *two* entrypoint classes (i.e. two classes with static main methods):

1. **ExportSampleClassifier** (in `edu.mit.cluster.text`):
Processes plaintext data in stand-alone diagnostic mode; output is directed to *stdout* as plaintext.
2. **HtmlSampleClassifier** (in `edu.mit.cluster.web`):
Processes streaming HTML data in HTTP proxy server mode; output is embedded directly in the HTTP payload as HTML-formatted text.



2.1. Exported Materials

The first entrypoint, `ExportSampleClassifier`, reads data from a file on disk that can be obtained by exporting Samples or DataSets from the GEO website with either the *Text* or *File* format:



Selected export files `gds_result.txt`, `gds_result2.txt`, etc. are included in the Eclipse project with source code [3]. Each file contains one or more Series (identified by GSE prefixes) and/or DataSets (identified by GDS prefixes) obtained through arbitrary searching on the GEO website, where each Sample or DataSet includes both 1) a summary, and 2) zero or more samples (identified by GSM prefixes). `ExportSampleClassifier` parses the summary and sample sections for its use, but discards the other sections.

The following is a representative Series in an export file:

9: GSE6102 record: Dietary exposure to soy or whey proteins alters colonic global gene expression profiles during rat colon tumorigenesis [Rattus norvegicus]

Summary: (Submitter supplied) We previously reported that lifetime consumption of soy proteins or whey proteins reduced the incidence of azoxymethane (AOM)-induced colon tumors in rats. To obtain insights into these effects, global gene expression profiles of colons from rats with lifetime ingestion of casein (CAS, control diet), soy protein isolate (SPI), and whey protein hydrolysate (WPH) diets were determined. We identified 31 induced and 49 repressed genes in the proximal colons of the SPI-fed group and 44 induced and 119 repressed genes in the proximal colons of the WPH-fed group, relative to CAS. Hierarchical clustering identified the co-induction or co-repression of multiple genes by SPI and WPH. The differential expression of I-FABP (2.92-, 3.97-fold down-regulated in SPI and WPH fed rats; $P = 0.023$, $P = 0.01$, respectively), cyclin D1 (1.61-, 2.42-fold down-regulated in SPI and WPH fed rats; $P = 0.033$, $P = 0.001$, respectively), and the c-neu proto-oncogene (2.46-, 4.10-fold down-regulated in SPI and WPH fed rats; $P < 0.001$, $P < 0.001$, respectively) mRNAs were confirmed by real-time quantitative RT-PCR. SPI and WPH affected colonic neuro-endocrine gene expression: peptide YY (PYY) and glucagon mRNAs were down-regulated in WPH fed rats, whereas somatostatin mRNA and corresponding circulating protein levels, were enhanced by SPI and WPH. The identification of transcripts co- or differentially-regulated by SPI and WPH diets suggests common as well as unique anti-tumorigenesis mechanisms of action which may involve growth factor, neuroendocrine and immune system genes. SPI and WPH induction of somatostatin, a known anti-proliferative agent for colon cancer cells, would inhibit tumorigenesis Keywords: Comparative genomic hybridization

1 related DataSet

1 related Platform

Type: Expression profiling by array

Supplementary Files: CEL EXP

Samples: 9

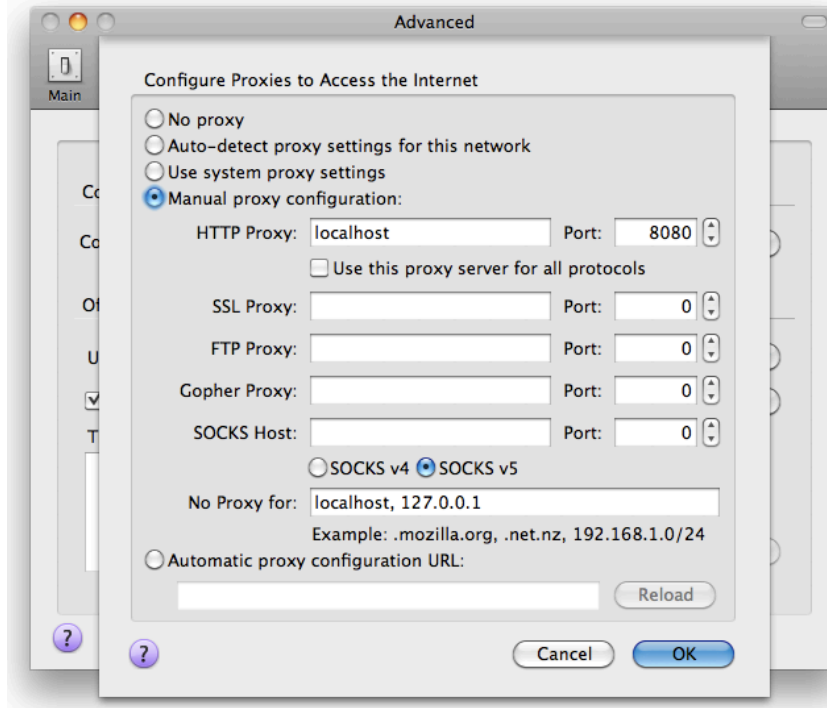
GSM141507: Proximal colon whey 1
GSM141501: Proximal colon casein 1
GSM141504: Proximal colon soy 1
GSM141502: Proximal colon casein 2
GSM141505: Proximal colon soy 2
GSM141508: Proximal colon whey 2
GSM141503: Proximal colon casein 3
GSM141506: Proximal colon soy 3

Summary

Samples

2.2. HTML Materials

The second endpoint, `HtmlSampleClassifier`, acts as a proxy server running on port 8080. It reads Series and DataSet data from an HTTP response stream and injects HTML-formatted results into that stream in real time. The proxy server is utilized when a web browser running on the sample machine is configured to use it. For example, the following configuration settings would be appropriate to use the proxy server with Firefox 3.5.5:



The proxy server injects HTML results into two types of GEO pages:

1. Search results listing Series and DataSets:

NCBI GEO DATASETS Gene Expression Omnibus

All Databases PubMed Nucleotide Protein Genome Structure OMIM PMC Journals Books

Search GEO DataSets for colon cancer intracellular folate Go Clear Save Search

Limits Preview/Index History Clipboard Details

Display Summary Show 20 Sort By Send to

All: 2 DataSets: 1 Platforms: 0 Series: 1

Items 1 - 2 of 2 One page.

1: GDS1424 record: Folic acid deficiency effect on colon cancer cells [Homo sapiens] GEO Profiles, Links

Summary: Analysis of HT-29 colon cancer cells cultured in pteroylglutamic acid or methyl-tetrahydrofolate at concentrations of 10 or 100 ng/ml each. Studies suggest that folate deficiency can have an inhibitory effect on the progression of established colonic tumor cells.
Parent Platform: GPL2721
Reference Series: GSE3099

Type: Expression profiling by array, transformed count

Subsets: 2 agent, 2 dose sets.

Samples: 22
GSM69219: 10PGA3B
GSM69220: 10PGA3A
GSM69221: 10PGA2B
GSM69222: 10PGA1B
GSM69223: 10PGA1A
GSM69207: 100PGA3B

Clustered samples: mthf (11)
a (5)
GSM69218: 100MTHF1A
10 (2)
GSM69228: 10MTHF1A

Injected results

2. Detail pages on a specific Series:

NCBI > GEO > Accession Display [?](#) Not logged in | [Login](#) [?](#)

Scope: Format: Amount: GEO accession:

Series GSE4115 [Query DataSets for GSE4115](#)

Status Public on Feb 27, 2006
Title Airway Epithelial Gene Expression Diagnostic for the Evaluation of Smokers with Suspect Lung Cancer
Organism(s) [Homo sapiens](#)
Experiment type Expression profiling by array
Summary RNA was obtained from histologically normal bronchial epithelium of smokers during time of clinical bronchoscopy from relatively accessible airway tissue. Gene expression data from smokers with lung cancer was compared with samples from smokers without lung cancer. This allowed us to generate a diagnostic gene expression profile that could distinguish the two classes. This profile could provide additional clinical benefit in diagnosing cancer amongst smokers with suspect lung cancer.
Keywords: Disease state analysis

Platforms (1) [GPL96 \[HG-U133A\] Affymetrix Human Genome U133A Array](#)

Samples (192) [GSM93997](#) Smoker NOT diagnosed with cancer Sample 283
[More...](#)
[GSM94019](#) Smoker diagnosed with cancer Sample 57
[GSM94020](#) Smoker diagnosed with cancer Sample 62

Clustered samples:

- diagnosed (187)
 - [GSM94032](#): Smoker diagnosed with cancer Sample 114
 - [GSM94033](#): Smoker diagnosed with cancer Sample 115
 - [GSM94038](#): Smoker diagnosed with cancer Sample 117
 - [GSM94024](#): Smoker diagnosed with cancer Sample 122
 - [GSM94029](#): Smoker diagnosed with cancer Sample 123
 - [GSM94028](#): Smoker diagnosed with cancer Sample 126
 - [GSM94021](#): Smoker diagnosed with cancer Sample 130
 - [GSM94027](#): Smoker diagnosed with cancer Sample 133
 - [GSM94034](#): Smoker diagnosed with cancer Sample 135
 - [GSM94025](#): Smoker diagnosed with cancer Sample 136
 - [GSM94035](#): Smoker diagnosed with cancer Sample 137

Injected results

We based our proxy server implementation on the open-source HTTP proxy server *jProxy* [4], which we needed to enhance to perform MIME-type detection, request header tweaking, and chunk encoding translation to perform HTML injection.

2.3. Method

Once GeoCluster parses plaintext or HTML input to extract an experiment's summary text and associated samples, it runs a data-structuring algorithm. The algorithm operates in three steps:

1. Tokenization
2. Clustering
3. Cluster naming

2.3.1. Tokenization

In the tokenization step, we tokenize the descriptions of samples using the following procedure:

1. Remove strings delimited by parentheses. Each string within parentheses becomes a token.
2. Split the remaining string into tokens based on whitespace and underscore (_) characters.
3. If each sample description in an experiment yields only 1 token, perform a second-pass tokenization by splitting the single token at the boundaries of letters and non-letters.
4. Normalize each token by trimming whitespace from both ends and converting all letters to lowercase. Note that we preserve pre-normalized token names in a hash table so the normalized forms can be converted back to their familiar representations for later display to the user.
5. Discard tokens of length 0.

| Example Sample Name | Resulting Tokens |
|---|--|
| Control GFPi-122 siRNA vs Reference (Replicate 1) | control; gfpi-122; reference; replicate 1; sirna; vs |
| 1 hr KLF4 Time Course Induced | 1; course; hr; induced; klf4; time |
| 100MTHF1A | 1; 100; a; mthf |
| MC38 tumor CpG 1826 treated_183 | 1826; 183; CpG; MC38; treated; tumor |

2.3.2. Clustering

Once we tokenize the descriptions of samples, we cluster the samples using agglomerative hierarchical clustering. Our application provides a parameter for linkage type; this allows us to cluster based on average linkage, complete linkage (minimum similarity), or single linkage (maximum similarity). By default, we use complete linkage. In practice, we have found the final clusters of most experiments do not depend the type of linkage we select.

For a similarly measure when performing linkage calculations, we use cosine similarity. Roughly, this indicates the proportion of tokens two samples have in common. In our Java code, we calculate cosine similarity in a manner consistent with S. Chapman [5]:

$$\text{xyCommonTermCount} / (\text{Math.sqrt}(x.\text{getTerms}().\text{size}()) * \text{Math.sqrt}(y.\text{getTerms}().\text{size}()))$$

Resulting values range from 0 (dissimilar) to 1 (identical).

While `xyCommonTermCount` would typically represents the number of tokens the two clusters have in common, we have adapted the similarity calculation slightly such that instead of a count of common tokens, `xyCommonTermCount` represents the *weight* of those common tokens by type. For tokens that contain alphabetic characters only, this weight is 1. For other tokens, such as tokens that contain a mix of alphabetic and numeric characters, this weight is 0.5. Given the extensive use of numerical values in sample descriptions to provide sequence numbers and identifiers that do not provide good indicators of similarity and may in fact throw off the clustering

algorithm, de-emphasizing these tokens improves the results. For example, if two clusters shared the tokens "lung carinoma t1 l4" the value of `xyCommonTermCount` would be 2.5, not 3.

Finally, we employ a "flattening" technique on the resultant cluster hierarchy so that a cluster may directly parent more than two subclusters. We do this for two reasons: 1) the distance among clusters is not significant for our purposes, only the clusters themselves, and 2) we believe usability is improved through consolidation. The flattening algorithm works as follows:

1. Select the root cluster.
2. With each child cluster of the cluster selected:
 - a. Perform step 2 recursively on the child cluster.
 - b. If the set of tokens common to all samples within the cluster, either directly or through descendents, of the child cluster is identical to that of the selected cluster, merge the child cluster and its parents. The children of the child cluster become the children of the selected cluster.

The following figure illustrates the effects of clustering. Here, the 9 samples are organized into a binomial tree of clusters as per the agglomerative clustering algorithm. This tree is not very satisfying, however, as the *casein*, *soy*, and *whye* samples do not appear neatly into 3 distinct clusters. After flattening, however, the clusters appear well-organized into their 3 logical groups.

```
Cluster [colon; Proximal]
-> Cluster [colon; Proximal]: ""
  -> Cluster [casein; colon; Proximal]
    -> Sample [GSM141503]: Proximal colon casein 3
    -> Cluster [casein; colon; Proximal]
      -> Sample [GSM141501]: Proximal colon casein 1
      -> Sample [GSM141502]: Proximal colon casein 2
    -> Cluster [colon; Proximal; whey]
      -> Sample [GSM141507]: Proximal colon whey 1
      -> Sample [GSM141508]: Proximal colon whey 2
  -> Cluster [colon; Proximal; soy]
    -> Sample [GSM141506]: Proximal colon soy 3
    -> Cluster [colon; Proximal; soy]: ""
      -> Sample [GSM141504]: Proximal colon soy 1
      -> Sample [GSM141505]: Proximal colon soy 2
```

Before
flattening

```
Cluster [colon; Proximal]: "colon; Proximal"
-> Cluster [casein; colon; Proximal]: "casein"
  -> Sample [GSM141501]: Proximal colon casein 1
  -> Sample [GSM141502]: Proximal colon casein 2
  -> Sample [GSM141503]: Proximal colon casein 3
-> Cluster [colon; Proximal; soy]: "soy"
  -> Sample [GSM141504]: Proximal colon soy 1
  -> Sample [GSM141505]: Proximal colon soy 2
  -> Sample [GSM141506]: Proximal colon soy 3
-> Cluster [colon; Proximal; whey]: "whey"
  -> Sample [GSM141507]: Proximal colon whey 1
  -> Sample [GSM141508]: Proximal colon whey 2
```

After
flattening

2.3.3. Naming

We name clusters based on insights from itemset mining techniques, specifically cover and support, as follows:

1. Find the set $T = (S_1 \cap S_2 \cap \dots \cap S_n) \cap (I - P)$, where $S_1 \dots S_n$ are sets of tokens of sample descriptions for all samples within the cluster, P is the same for the cluster's parent, and I is the set of all tokens. This, effectively, is the largest itemset with 100% support over all subclusters $S_1 - S_n$ and 0% support over parent cluster P .
2. Find tokens in T that are acronyms defined in the summary of the experiment, and replace these tokens with full acronym definitions. To discover these definitions, we adapt the detection algorithm written by A. Schwartz & M. Hearst [6].
3. Denormalize tokens in T by reversing the normalization process described in 2.3.1
4. Sort items in T .
5. Convert T to a string of semicolon-separated items. This becomes the cluster name.

The following diagram illustrates the results of this naming technique. Cluster names are highlighted, and to the left of each name is brackets is a representation of the largest set of tokens with 100% support over all subclusters. Notice that the "A; Argryin" cluster preserves the case of the original sample descriptions, due to denormalization, and does not contain the token "treated," as this token is covered by its parent cluster.

```
Cluster []: ""
-> Cluster [siRNA]: "siRNA"
  -> Cluster [12h; siRNA]: "12h"
    -> Sample [GSM212662]: siRNA 12h_1
    -> Sample [GSM297805]: siRNA 12h_2
  -> Cluster [24h; siRNA]: "24h"
    -> Sample [GSM212663]: siRNA 24h_1
    -> Sample [GSM297807]: siRNA 24h_2
-> Cluster [treated]: "treated"
  -> Cluster [A; Argryin; treated]: "A; Argryin"
    -> Sample [GSM212658]: Argryin A treated 14h
    -> Sample [GSM212660]: Argryin A treated 48h
  -> Cluster [12h; A; Argryin; treated]: "12h"
    -> Sample [GSM297801]: Argryin A treated 12h_1
    -> Sample [GSM297804]: Argryin A treated 12h_2
  -> Cluster [24h; A; Argryin; treated]: "24h"
    -> Sample [GSM212659]: Argryin A treated 24h
    -> Sample [GSM297802]: Argryin A treated 24h_1
    -> Sample [GSM297806]: Argryin A treated 24h_2
  -> Cluster [bortezomib; treated]: "bortezomib"
    -> Sample [GSM212655]: bortezomib treated 14h
    -> Sample [GSM212656]: bortezomib treated 24h
    -> Sample [GSM212657]: bortezomib treated 48h
-> Cluster [untreated]: "untreated"
  -> Sample [GSM212654]: untreated
  -> Sample [GSM212661]: untreated_1
  -> Sample [GSM297803]: untreated_2
```

Where feasible, token names are replaced with acronym definitions. For example, a cluster with $T = \{ \text{IPF, biopsy} \}$ will be not be named simply "IPF; biopsy" but rather "IPF (idiopathic pulmonary fibrosis); biopsy" so long as the summary of the experiment containing the sample provides this definition.

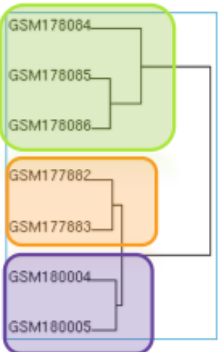
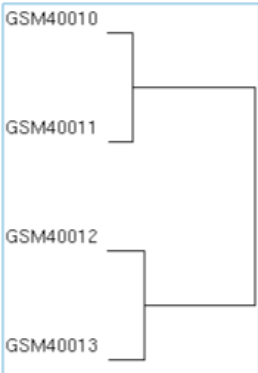
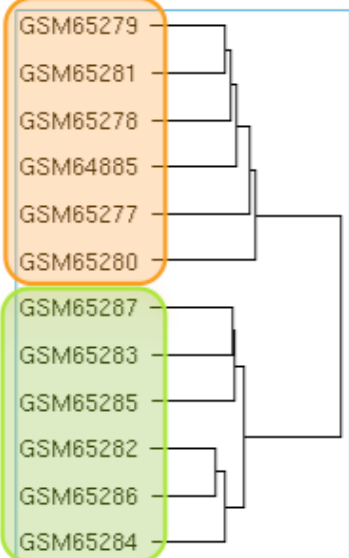
3. Results

We consider the results of *GeoCluster* both qualitatively and quantitatively. First, qualitatively, the clusters appear intuitively sensible and useful, especially in cases where there are more than 10 samples and the default sort on sample descriptions does not offer logical grouping. For example, for the following DataSet experiment, the original sample list itself appears disorganized, whereas the clustered samples appear nicely grouped into *asbestos; exposed* and *control* clusters, with further subdivisions from there.

The screenshot shows a GEO record for '6: GDS2604 record: Asbestos effect on epithelial and mesothelial lung cell lines: time course [Homo sapiens]'. The record includes a summary, type, subsets, supplementary files, and samples. The 'Samples' section lists 27 individual samples, and the 'Clustered samples' section shows a hierarchical grouping of these samples into clusters based on exposure conditions.

| Category | Sample ID | Description |
|-------------------|------------------------|------------------------------|
| Samples | GSM139646 | A549 control 0h |
| | GSM139660 | Beas2B control 0h |
| | GSM139640 | A549 asbestos exposed 1h |
| | GSM139647 | A549 control 1h |
| | GSM139654 | Beas2B asbestos exposed 1h |
| | GSM139661 | Beas2B control 1h |
| Clustered samples | asbestos; exposed (14) | |
| | GSM139760 | Met5A asbestos exposed 1h |
| | GSM139659 | Beas2B asbestos exposed 48h |
| | GSM139642 | A549 asbestos exposed 7 days |
| | 1h (2) | |
| | GSM139640 | A549 asbestos exposed 1h |
| | GSM139654 | Beas2B asbestos exposed 1h |
| | 24h (3) | |
| | GSM139643 | A549 asbestos exposed 24h |
| | Beas2B (2) | |
| | GSM139656 | Beas2B asbestos exposed 24h |

For a quantitative analysis, we examine a subset of GEO Series with corresponding DataSets. DataSets, through manual effort, have been normalized and their samples clustered using a variety of linkage types and similarity measures. In the following table, we compare our clustered results to these pre-computed results by computing Rand indices [8][9], a scale of 0 to 1 where 0 indicates no agreement and 1 indicates complete agreement. For these comparisons, we used complete linkage and compatible similarity computations. We found that changing the similarity metric or linkage did not change any rand index by more than 0.1, so we exclude those variations from our analysis here.

| GEO DataSet Clusters | Our Clusters | Rand Index |
|---|---|------------|
|  | <p>C57/129; WT (2)</p> <ul style="list-style-type: none"> GSM180005: Newborn Mouse Ovary_C57/129 WT_2 GSM180004: Newborn Mouse Ovary_C57/129 WT_3 <p>Null (5)</p> <ul style="list-style-type: none"> LHX8 (3) <ul style="list-style-type: none"> GSM178084: Newborn Mouse Ovary_LHX8 Null_1 GSM178085: Newborn Mouse Ovary_LHX8 Null_2 GSM178086: Newborn Mouse Ovary_LHX8 Null_3 Nobox (2) <ul style="list-style-type: none"> GSM177882: Newborn Mouse Ovary_Nobox Null_2 GSM177883: Newborn Mouse Ovary_Nobox Null_3 | 0.90 |
|  | <p>GSM40010: AFS084-1-291004</p> <p>GSM40011: AFS084-2-291004</p> <p>GSM40012: AFS084-4-291004</p> <p>GSM40013: AFS084-6-291004</p> | 0.33 |
|  | <ul style="list-style-type: none"> • endometrioid; stage; carcinomas (12) <ul style="list-style-type: none"> ○ E2; oestrogen (4) <ul style="list-style-type: none"> ■ I (2) <ul style="list-style-type: none"> GSM65278: EECs from stage I endometrioid carcinomas oestrogen(E2) rep 1 GSM65279: EECs from stage I endometrioid carcinomas oestrogen(E2) rep 2 ■ II (2) <ul style="list-style-type: none"> GSM65284: EECs from stage II endometrioid carcinomas oestrogen(E2) rep 1 GSM65285: EECs from stage II endometrioid carcinomas oestrogen(E2) rep 2 ○ control (4) <ul style="list-style-type: none"> ■ I (2) <ul style="list-style-type: none"> GSM64885: EECs from stage I endometrioid carcinomas control rep 1 GSM65277: EECs from stage I endometrioid carcinomas control rep 2 ■ II (2) <ul style="list-style-type: none"> GSM65282: EECs from stage II endometrioid carcinomas control rep 1 GSM65283: EECs from stage II endometrioid carcinomas control rep 2 ○ tamoxifen; TAM (4) <ul style="list-style-type: none"> ■ I (2) <ul style="list-style-type: none"> GSM65280: EECs from stage I endometrioid carcinomas tamoxifen(TAM) rep 1 GSM65281: EECs from stage I endometrioid carcinomas tamoxifen(TAM) rep 2 ■ II (2) | 0.19 |

4. Discussion

In the first quantitative comparison in section 3 (DataSet 3254), we find very high agreement (rand index = 0.90) between GEO clusters and our clusters. This is not surprising, given that the sample descriptions provide excellent cues regarding how the samples should be organized with the tokens *C57/129*, *WT*, *LHX8*, and *Null*. Our algorithm prefers to group by tokens such as *WT* and *Null* over 1, 2, and 3—which are also identifying tokens—because of the special weighting we apply, as previously described in 2.3.2.

In the second quantitative comparison, the sample descriptions do not have any identifying tokens that aid in clustering, so our algorithm places all samples in the same cluster. The corresponding GEO cluster, on the other hand, shows two subclusters a significant distance apart, indicating that GEO’s own clustering is leveraging information not present in the samples. The two results do not agree (rand index = 0.33). This is a clear limitation of our method—we are structuring data based on sample descriptions alone; if these descriptions do not provide sufficient cues as to the logical groupings of samples, our method will not produce good results.

In the third quantitative comparison, our method yields clusters along the descriptive dimensions of *oestrogen*, *control*, and *tamoxifen*, with subclusters along stage dimensions of *stage I* and *stage II*. This disagrees with GEO’s cluster (rand index = 0.19), where the *stage I* and *stage II* divisions appear to be the key differentiator between the two main clusters. This is a second limitation of our method—with multiple tokens upon which to cluster samples, the method is unable to determine the most relevant tokens. In this case, the method cannot make an informed choice between the “stage” tokens and the “descriptive” tokens. It happens to choose the descriptive tokens since these will have greater cosine similarity.

Interestingly, despite the strong disagreements with GEO clustering, this third example hints at the value of our method. We notice that our method was able to cluster *all* 16 samples in the experiment, whereas GEO clustering is only able to cluster the 12 samples backed by a manual normalization process. As our technique does not rely on manual normalization, we are able to cluster samples for all GEO Series, not just the 15% of those series that have corresponding DataSets.

Further, as demonstrated by our HTTP proxy, our approach offers an intuitive user interface extension to the GEO website, namely a hierarchical sample browser on search results pages as well as GEO Series detail pages. This provides users with ability to immediately see samples in clustered form, without having to navigate several levels into the cluster viewer—that is, for the small subset of experiments where this is even available. We propose enhancing the GEO web server to format results in a format consistent with our HTML injections—with options to display samples alphabetically, by clusters using GEO’s computations (where available), and by clusters using our computations.

At this point, we would be remiss not to note that disagreement with the GEO clustering does not imply that our results are necessary *wrong*, just different. The GEO clusters are not gold standards, so our clustering provides a differing, albeit less informed, view of the samples. Offering our results side-by-side with GEO clusters (where available) offers researchers additional utility, ultimately enhancing the mission of the GEO repository.

Our work is certainly not complete. In addition to GEO website integration, we imagine future work along the following dimensions:

1. *Code tuning*: There are several methods in the code that are not yet optimized
2. *Tokenizer tuning*: Better clusters may be obtained by tokenizing in smarter ways, such as eliminating noise tokens like sequence numbers (e.g. "#2") that may lead to clustering along inappropriate dimensions
3. *Deep tokenization*: Include among a sample's tokens not only its description but key aspects on the sample details page, such as *source name*, *organism*, and *characteristics*.
4. *Cluster tuning*: Tweaking the weights of various tokens, perhaps in specific contexts, perhaps giving acronyms or drug names more weight and stop words little weight
5. *Enhanced acronym detection*: Detect acronym definitions within nested parentheses, or even build an acronyms database to be used across experiments
6. *Additional validation*: Solicit feedback of cluster results among GEO website users and/or contributors

5. Conclusion

Our approach to clustering samples in the Gene Expression Omnibus provides intuitive results across most GEO Series, and can appropriately supplement GEO DataSets despite "disagreeable" Rand index values that average below 0.5. We believe our clustering technique offers additional value to users of the GEO website by structuring data and making that structure readily accessible, as prototyped by injecting HTML representing real-time clustering results into the GEO website using an HTTP proxy server. As a next step, we hope to collaborate with GEO administrators to incorporate the methods described in this paper directly into the GEO website.

Acknowledgements

We thank Dr. Ronilda Lacson for her guidance. Dr. Lacson conceived of the approach to sample clustering we pursued in this paper.

References

- [1] "Gene Expression Omnibus (GEO) Main page." <http://www.ncbi.nlm.nih.gov/geo/>
- [2] "GEO FAQ." <http://www.ncbi.nlm.nih.gov/geo/info/faq.html>
- [3] <http://www.mit.edu/~timvasil/SampleClustering.zip>
- [4] J. Robichaux, "Java HTTP Proxy." <http://www.nsftools.com/tips/JavaTips.htm>
- [5] S. Chapman, "Sam's String Metrics." <http://www.dcs.shef.ac.uk/~sam/stringmetrics.html>
- [6] A. Schwartz & M. Hearst, "A Simple Algorithm for Identifying Abbreviation Definitions in Biomedical Text." <http://biotext.berkeley.edu/papers/psb03.pdf>
- [7] "GEO DataSet Browser." <http://www.ncbi.nlm.nih.gov/sites/GDSbrowser>
- [8] W. M. Rand. "Objective criteria for the evaluation of clustering methods." *Journal of the American Statistical Association*. **66**: 846–850, 1971.
- [9] E. B. Fowlkes & C. L. Mallows. "A Method for Comparing Two Hierarchical Clusterings". *Journal of the American Statistical Association*. **78** (383): 553–584, 1983.

Appendix: Source Code

The entirety of the source code for the GeoCluster Java application follows. The source code, along with an Eclipse project file and sample input files, is also available online at <http://www.mit.edu/~timvasil>.