Ohsumed Baselines

1. Data (/shared/ohsumed/data)

#download Ohsumed data

```
mkdir –p /shared/ohsumed
cd /shared/ohsumed
wget http://trec.nist.gov/data/filtering/t9.filtering.tar.gz
```

#untar Ohsumed data

```
tar xvzf t9.filtering.tar.gz --owner root --group root --no-same-owner 2>&1 >> ohsumeddata.log
```

#copy data into /shared/ohsumed/data

```
cp /shared/ohsumed/ohsu-trec/trec9-test/ohsumed.88-91 /shared/ohsumed/data
cp /shared/ohsumed/ohsu-trec/trec9-train/ohsumed.87 /shared/ohsumed/data
```

#convert data into trec format

```
cd ~/biocaddie/scripts
./ohsumed2trec.sh
```

***#documents=**348566**

Output: **/shared/ohsumed/data/trecText/ohsumed_all.txt**

Also make a copy at **/data/ohsumed/data/**

2. Indexes (/shared/ohsumed/indexes/ohsumed_all)

Index *param* file: ~/biocaddie/index/build_index.ohsumed.params

#Content

```
<parameters>
  <index>/shared/ohsumed/indexes/ohsumed_all</index>
  <indexType>indri</indexType>
  <corpus>
    <path>/shared/ohsumed/data/trecText/ohsumed_all.txt</path>
    <class>trectext</class>
  </corpus>
</parameters>
```

#Build index

```
mkdir -p /shared/ohsumed/indexes/
cd ~/biocaddie
IndriBuildIndex index/build_index.ohsumed.params
```

Output is saved at **/shared/ohsumed/indexes/ohsumed_all**

Also make a copy at **/data/ohsumed/indexes/ohsumed_all**

3. Queries

#copy topics to /shared/ohsumed/queries folder

```
cp /shared/ohsumed/ohsu-trec/trec9-train/query.ohsu.1-63 /shared/ohsumed/queries
cp /shared/ohsumed/ohsu-trec/pre-test/query.ohsu.test.1-43 /shared/ohsumed/queries
```

There are two queries - for pre-test and for training (and test) sets

We create **queries.combined.orig** (total 106) including all the queries and **queries.combined.short** (total 63) for the queries used for training and test sets (not include pre-test queries)

#convert query into trec format *(use ohsumedtopics2trec.sh to create queries.combined.orig and ohsumedtopics2trec_v2.sh to create queries.combined.short)*

```
cd ~/biocaddie
scripts/ohsumedtopics2trec.sh
```

Output is saved at **/shared/ohsumed/queries**

Also make a copy of the query at **/data/ohsumed/queries**

4. Qrels

#copy qrels to /shared/ohsumed/qrels folder

```
cp /shared/ohsumed/ohsu-trec/trec9-train/qrels.ohsu.batch.87 /shared/ohsumed/qrels
cp /shared/ohsumed/ohsu-trec/pre-test/qrels.ohsu.test.87 /shared/ohsumed/qrels
cp /shared/ohsumed/ohsu-trec/trec9-test/qrels.ohsu.88-91 /shared/ohsumed/qrels
```

Similar to queries, the 3 qrels files include relevant judgements for pre-test, training and test sets.

In case of queries.combined.orig, all qrels files are used - **qrels.all**

In case of queries.combined.short, only qrels for training and test sets are used (qrels.ohsu.batch.87 and qrels.ohsu.88-91) - **qrels.notest**

However, the downloaded qrels are missing one column for trec_eval to process, we have to add the missing column before using.

#convert qrels into correct format for trec_eval (add in 0 in second column)

```
cat qrels.ohsu.* | sed 's/\t/\t0\t/1' > qrels.all
```

Output is saved at **/shared/ohsumed/qrels**

Also make a copy of the qrels at **/data/ohsumed/qrels**

5. IndriRunQuery - Output

```
cd ~/biocaddie/baselines/ohsumed
./<model>.sh <topic> <collection> |parallel -j 20 bash -c "{}"
```

For orig queries:

```
./jm.sh orig combined| parallel -j 20 bash -c "{}"
./dir.sh orig combined| parallel -j 20 bash -c "{}"
./dir.sh orig combined| parallel -j 20 bash -c "{}"
./two.sh orig combined| parallel -j 20 bash -c "{}"
./okapi.sh orig combined| parallel -j 20 bash -c "{}"
./rm3.sh orig combined| parallel -j 20 bash -c "{}"
```

For short queries:

```
./jm.sh short combined| parallel -j 20 bash -c "{}"
./dir.sh short combined| parallel -j 20 bash -c "{}"
./dir.sh short combined| parallel -j 20 bash -c "{}"
./two.sh short combined| parallel -j 20 bash -c "{}"
./okapi.sh short combined| parallel -j 20 bash -c "{}"
./rm3.sh short combined| parallel -j 20 bash -c "{}"
```

IndriRunQuery outputs for different baselines are stored at:

`/data/ohsumed/output/tfidf/combined/orig`

`/data/ohsumed/output/dir/combined/orig`

`/data/ohsumed/output/okapi/combined/orig`

`/data/ohsumed/output/jm/combined/orig`

`/data/ohsumed/output/two/combined/orig`

`/data/ohsumed/output/rm3/combined/orig`

`---`

`/data/ohsumed/output/tfidf/combined/short`

`/data/ohsumed/output/dir/combined/short`

`/data/ohsumed/output/okapi/combined/short`

`/data/ohsumed/output/jm/combined/short`

`/data/ohsumed/output/two/combined/short`

`/data/ohsumed/output/rm3/combined/short`

6. Cross-validation

cd ~/biocaddie

For orig queries which use qrels.all

```
scripts/mkeval_ohsumed.sh <model> <topics> <collection>
```

`Eg: scripts/mkeval_ohsumed.sh tfidf orig combined`

For short queries which use qrels.notest

```
scripts/mkeval_ohsumed_v2.sh <model> <topics> <collection>
```

`Eg: scripts/mkeval_ohsumed_v2.sh tfidf short combined`

7. Compare models

```
cd ~/biocaddie
Rscript scripts/compare_ohsumed.R <collection> <from model> <to model> <topic>
```

Results (compared to tfidf baseline)

Using orig queries (pre-test queries included)

Model	MAP	NDCG	P@20	NDCG@20	P@100	NDCG@100	Notes	Date
tfidf	0.2204	0.4538	0.2995	0.2904	0.1735	0.3376	Sweep b and k1	06/07/17
Okapi	0.2218	0.4557	0.2819-	0.3035	0.1717	0.3386	Sweep b, k1, k3	06/07/17
QL (JM)	0.1876-	0.4212-	0.2505-	0.2773	0.1403-	0.295-	Sweep lambda	06/07/17
QL (Dir)	0.2032-	0.4359-	0.2713-	0.2927	0.1633-	0.3304	Sweep mu	06/07/17

QL (TS)	0.2101-	0.4415-	0.2761-	0.3029	0.1638-	0.3277	Sweep mu and lambda	06/07/17
RM3	0.2618+	0.4592	0.3277+	0.2965	0.1913+	0.3662+	Sweep mu, fbDocs, fbTerms, and lambda	06/08/17

```
root@integration-1:~/biocaddie# Rscript scripts/compare_ohsumed.R combined tfidf dir orig
[1] "map 0.2204 0.2032 p= 0.9988"
[1] "ndcg 0.4538 0.4359 p= 0.9987"
[1] "P_20 0.2995 0.2713 p= 0.9985"
[1] "ndcg_cut_20 0.2904 0.2927 p= 0.417"
[1] "P_100 0.1735 0.1633 p= 0.9945"
[1] "ndcg_cut_100 0.3376 0.3304 p= 0.7764"
root@integration-1:~/biocaddie# Rscript scripts/compare_ohsumed.R combined tfidf jm orig
[1] "map 0.2204 0.1876 p= 0.9966"
[1] "ndcg 0.4538 0.4212 p= 0.9992"
[1] "P_20 0.2995 0.2505 p= 0.9999"
[1] "ndcg_cut_20 0.2904 0.2773 p= 0.8572"
[1] "P_100 0.1735 0.1403 p= 1"
[1] "ndcg_cut_100 0.3376 0.295 p= 0.9996"
root@integration-1:~/biocaddie# Rscript scripts/compare_ohsumed.R combined tfidf two orig
[1] "map 0.2204 0.2101 p= 0.972"
[1] "ndcg 0.4538 0.4415 p= 0.9859"
[1] "P_20 0.2995 0.2761 p= 0.9954"
[1] "ndcg_cut_20 0.2904 0.3029 p= 0.1072"
[1] "P_100 0.1735 0.1638 p= 0.9992"
[1] "ndcg_cut_100 0.3376 0.3277 p= 0.857"
root@integration-1:~/biocaddie# Rscript scripts/compare_ohsumed.R combined tfidf okapi orig
[1] "map 0.2204 0.2218 p= 0.4445"
[1] "ndcg 0.4538 0.4557 p= 0.414"
[1] "P_20 0.2995 0.2819 p= 0.975"
[1] "ndcg_cut_20 0.2904 0.3035 p= 0.1157"
[1] "P_100 0.1735 0.1717 p= 0.6907"
[1] "ndcg_cut_100 0.3376 0.3386 p= 0.4437"
```

Using short queries (pre-test queries not included)

Model	MAP	NDCG	P@20	NDCG@20	P@100	NDCG@100	Notes	Date
tfidf	0.3188	0.6084	0.45	0.4255	0.2657	0.4625	Sweep b and k1	06/07/17
Okapi	0.3117	0.6044	0.4408	0.4277	0.261	0.4569	Sweep b, k1, k3	06/07/17
QL (JM)	0.2545-	0.5527-	0.3908-	0.3882-	0.2135-	0.3883-	Sweep lambda	06/07/17
QL (Dir)	0.2924-	0.5866-	0.3975	0.4018-	0.2492-	0.432-	Sweep mu	06/07/17
QL (TS)	0.2934-	0.5828-	0.4092-	0.4122	0.2508-	0.4385-	Sweep mu and lambda	06/07/17
RM3	0.3717+	0.6087	0.5067+	0.4529 (p-value: 0.0541)	0.291+	0.4934+	Sweep mu, fbDocs, fbTerms, and lambda	06/08/17

```
root@integration-1:~/biocaddie# Rscript scripts/compare_ohsumed.R combined tfidf dir short
[1] "map 0.3188 0.2924 p= 0.9997"
[1] "ndcg 0.6084 0.5866 p= 0.9994"
[1] "P_20 0.45 0.3975 p= 0.9998"
[1] "ndcg_cut_20 0.4255 0.4018 p= 0.9881"
[1] "P_100 0.2657 0.2492 p= 0.9947"
[1] "ndcg_cut_100 0.4625 0.432 p= 0.9999"
root@integration-1:~/biocaddie# Rscript scripts/compare_ohsumed.R combined tfidf jm short
[1] "map 0.3188 0.2545 p= 1"
[1] "ndcg 0.6084 0.5527 p= 1"
[1] "P_20 0.45 0.3908 p= 0.9984"
[1] "ndcg_cut_20 0.4255 0.3882 p= 0.9973"
[1] "P_100 0.2657 0.2135 p= 1"
[1] "ndcg_cut_100 0.4625 0.3883 p= 1"
root@integration-1:~/biocaddie# Rscript scripts/compare_ohsumed.R combined tfidf okapi short
[1] "map 0.3188 0.3117 p= 0.7974"
[1] "ndcg 0.6084 0.6044 p= 0.6834"
[1] "P_20 0.45 0.4408 p= 0.7506"
[1] "ndcg_cut_20 0.4255 0.4277 p= 0.4236"
[1] "P_100 0.2657 0.261 p= 0.791"
[1] "ndcg_cut_100 0.4625 0.4569 p= 0.747"
root@integration-1:~/biocaddie# Rscript scripts/compare_ohsumed.R combined tfidf two short
[1] "map 0.3188 0.2934 p= 1"
[1] "ndcg 0.6084 0.5828 p= 0.9997"
[1] "P_20 0.45 0.4092 p= 0.9989"
[1] "ndcg_cut_20 0.4255 0.4122 p= 0.89"
[1] "P_100 0.2657 0.2508 p= 0.9991"
[1] "ndcg_cut_100 0.4625 0.4385 p= 0.9992"
```

8. Comments:

BioCADDIE dataset contains descriptive metadata (structured and unstructured) of more than 1.5 millions documents from biomedical datasets. There are 20 queries which are manually refined and shortened including important keywords. Relevant judgements contains 3 categories 0-"not relevant", 1-"possibly relevant" and 2-"definitely relevant".

TREC CDS dataset is a collection of 733.328 full-text biomedical literature of journal articles. 30 topics are provided, each includes topic "description" (containing a complete account of the patients' visits, including details such as their vital statistics, drug dosages, etc) and topic "summary" (a simplified versions of the narratives that contain less irrelevant information). Queries are constructed by topic summaries. Similar to bioCADDIE, relevant judgements are divided into 3 categories 0-"not relevant", 1-"possibly relevant" and 2-"definitely relevant".

The OHSUMED test collection is a set of 348,566 references/documents from MEDLINE, the on-line medical information database, consisting of titles and /or abstracts from 270 medical journals. Compared to the two above collections, Ohsumed dataset is quite small. OHSUMED topics include 2 fields - tilte (p atient description) and description (information request). Topic descriptions are selected to construct queries. Relevant judgements include 2 categories 1-"possibly relevant" and 2-"definitely relevant"

Based on the characteristics of 3 collections, TREC CDS is far different from bioCADDIE and Ohsumed as it uses full text search and its queries are patient visit record summary instead of common information queries. Ohsumed collection is closer to bioCADDIE in term of dataset similarity (non full-text). However, bioCADDIE queries are short keyword queries while OHSUMED queries are short verbose queries.

As per the baselines run results over all 3 collections, RM3 baselines generally perform well and consistent. Especially for TREC CDS and Ohsumed, RM3 gives best results for most of the metrics compared to other baselines. This was expected as RM3 based on Rocchio relevance feedback which can help to generate good query (query expansion) even we don't know the collection well.

One surprising result was that Query likelihood baselines with smoothing (such as JM, Dir and TS) did not improve the retrieval results over TFIDF for any metrics in TREC CDS and Ohsumed collections as bioCADDIE or previous studies did (http://trec.nist.gov/pubs/trec23/papers/pro-UCLA_MII_clinical.pdf). However, type of queries could be an important factor that might cause the differences in retrieval results. This was also mentioned in the study of Zhai C (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.8978) that queries with only keywords tend to perform better than more verbose queries (*** Note: Zhai also mentioned longer queries performed better than short queries)

We tried to examine the difference in using verbose queries and keyword queries on Ohsumed collection.

Using original queries (verbose queries) for OHSUMED

Model	MAP	NDCG	P@20	NDCG@20	P@100	NDCG@100	Notes	Date
tfidf	0.3188	0.6084	0.45	0.4255	0.2657	0.4625	Sweep b and k1	06/07/17
QL (JM)	0.2545-	0.5527-	0.3908-	0.3882-	0.2135-	0.3883-	Sweep lambda	06/07/17
QL (Dir)	0.2924-	0.5866-	0.3975	0.4018-	0.2492-	0.432-	Sweep mu	06/07/17
QL (TS)	0.2934-	0.5828-	0.4092-	0.4122	0.2508-	0.4385-	Sweep mu and lambda	06/07/17

Using manually refined queries (mostly keywords) for OHSUMED

Model	MAP	NDCG	P@20	NDCG@20	P@100	NDCG@100	Notes	Date
tfidf	0.315	0.5949	0.4198	0.3802	0.2614	0.4454	Sweep b and k1	06/09/17
QL (JM)	0.2587-	0.5466-	0.3817-	0.3608	0.2257-	0.3806-	Sweep lambda	06/09/17
QL (Dir)	0.3027-	0.5883	0.4087	0.379	0.261	0.4333-	Sweep mu	06/09/17
QL (TS)	0.3052-	0.5871	0.4159	0.3896	0.2627	0.4354-	Sweep mu and lambda	06/09/17

We can see that when using keyword queries, the difference in retrieval results between tfidf and QL is smaller.

Specifically, tfidf performed worse for all metrics when using keyword queries than using verbose/original queries. QL (JM) also performed worse for NDCG, P@20, NDCG@20 and NDCG@100. However, QL (Dir) and QL (TS) performed better for most of the metrics. This matched with the finding in Zhai's study that JM works worst for short keywords queries but more effective when queries are verbose while Dir works better for concise keyword queries then verbose queries.

Also number of queries used for running baselines in each collection could be considered for the difference.

Below is the cross validation results for **map** metric of tfidf, dir, jm and two baselines for all queries in Ohsumed collection (there are only 97 queries listed instead of 106 because 7 other queries contain invalid punctuations for trec_eval to process)

Compared to tfidf, dir has 26/97 queries with higher map values; jm has 19/97 queries with higher map values; and two stage has 27/97 queries with higher map values.

Hence, if we use a smaller number of queries (such as less than 25 queries), there are possibilities that QL might yield better results than tfidf.

	tfidf		dir		jm		two	
OHSU-TEST1	k1=0.7:b=0.5.eval	0	500.eval	0	lambda=0.7.eval	0	mu=250:lambda=0.6.eval	0
OHSU-TEST11	k1=0.7:b=0.5.eval	0.0013	500.eval	**0.005**	lambda=0.7.eval	**0.0059**	mu=250:lambda=0.6.eval	**0.004**
OHSU-TEST12	k1=0.7:b=0.5.eval	0.0627	500.eval	0.0138	lambda=0.7.eval	0.0084	mu=250:lambda=0.6.eval	0.0171
OHSU-TEST14	k1=0.7:b=0.5.eval	0.0425	500.eval	0.0315	lambda=0.7.eval	**0.047**	mu=250:lambda=0.6.eval	**0.0566**
OHSU-TEST15	k1=0.7:b=0.5.eval	0	500.eval	0	lambda=0.7.eval	0	mu=250:lambda=0.6.eval	0
OHSU-TEST16	k1=0.7:b=0.5.eval	0	500.eval	0	lambda=0.7.eval	0	mu=250:lambda=0.6.eval	0
OHSU-TEST17	k1=0.7:b=0.4.eval	0.0996	500.eval	0.0727	lambda=0.7.eval	0.0991	mu=250:lambda=0.6.eval	0.0857
OHSU-TEST19	k1=0.7:b=0.5.eval	0.0273	500.eval	0.0226	lambda=0.7.eval	0.0174	mu=250:lambda=0.6.eval	0.0229
OHSU-TEST2	k1=0.7:b=0.5.eval	0	500.eval	0	lambda=0.7.eval	0	mu=250:lambda=0.6.eval	0
OHSU-TEST20	k1=0.7:b=0.5.eval	0.0445	500.eval	**0.0478**	lambda=0.7.eval	**0.0529**	mu=250:lambda=0.6.eval	**0.0493**
OHSU-TEST21	k1=0.7:b=0.5.eval	0.0176	500.eval	0.0128	lambda=0.7.eval	0.0057	mu=250:lambda=0.6.eval	0.0107
OHSU-TEST22	k1=0.7:b=0.5.eval	0.0244	500.eval	0.0213	lambda=0.7.eval	**0.0278**	mu=250:lambda=0.6.eval	0.0244
OHSU-TEST23	k1=0.7:b=0.5.eval	0.0611	500.eval	0.0383	lambda=0.7.eval	0.0329	mu=250:lambda=0.6.eval	0.0382
OHSU-TEST25	k1=0.7:b=0.5.eval	0.1011	500.eval	**0.1199**	lambda=0.7.eval	0.0832	mu=250:lambda=0.6.eval	**0.1104**
OHSU-TEST26	k1=0.7:b=0.5.eval	0.018	500.eval	0.0165	lambda=0.7.eval	0.0161	mu=250:lambda=0.6.eval	0.0155
OHSU-TEST29	k1=0.7:b=0.5.eval	0.25	500.eval	0.25	lambda=0.7.eval	**0.2511**	mu=250:lambda=0.6.eval	0.25
OHSU-TEST30	k1=0.7:b=0.5.eval	0.1667	1000.eval	**0.3333**	lambda=0.7.eval	**1**	mu=250:lambda=0.7.eval	**0.5**
OHSU-TEST31	k1=0.7:b=0.5.eval	0.0131	500.eval	0.0109	lambda=0.7.eval	0.011	mu=250:lambda=0.6.eval	0.0111
OHSU-TEST32	k1=0.7:b=0.5.eval	0.021	500.eval	0.014	lambda=0.7.eval	0.0098	mu=250:lambda=0.6.eval	0.0132
OHSU-TEST33	k1=0.7:b=0.5.eval	0.0908	500.eval	0.0425	lambda=0.7.eval	0.0195	mu=250:lambda=0.6.eval	0.0782
OHSU-TEST34	k1=0.7:b=0.5.eval	0.1071	500.eval	0.0929	lambda=0.7.eval	**0.1626**	mu=250:lambda=0.6.eval	**0.1174**
OHSU-TEST35	k1=0.7:b=0.5.eval	0.0068	500.eval	0.0024	lambda=0.7.eval	0.0019	mu=250:lambda=0.6.eval	0.0029
OHSU-TEST37	k1=0.7:b=0.5.eval	0.0773	500.eval	0.0597	lambda=0.7.eval	0.0559	mu=250:lambda=0.6.eval	0.0694
OHSU-TEST38	k1=0.7:b=0.5.eval	0.0439	500.eval	0.031	lambda=0.7.eval	0.0337	mu=250:lambda=0.6.eval	0.0407
OHSU-TEST39	k1=0.7:b=0.5.eval	0	500.eval	0	lambda=0.7.eval	0	mu=250:lambda=0.6.eval	0
OHSU-TEST4	k1=0.7:b=0.5.eval	0.2	500.eval	0.1667	lambda=0.7.eval	0.1667	mu=250:lambda=0.6.eval	0.1667
OHSU-TEST40	k1=0.7:b=0.5.eval	0.0013	500.eval	**0.002**	lambda=0.7.eval	**0.0016**	mu=250:lambda=0.6.eval	**0.0021**
OHSU-TEST41	k1=0.7:b=0.5.eval	0.0023	500.eval	**0.0036**	lambda=0.7.eval	**0.0048**	mu=250:lambda=0.6.eval	**0.0034**
OHSU-TEST42	k1=0.7:b=0.5.eval	0.0074	500.eval	0.0067	lambda=0.7.eval	0.0064	mu=250:lambda=0.6.eval	0.0067
OHSU-TEST43	k1=0.7:b=0.5.eval	7.00E-04	500.eval	6.00E-04	lambda=0.7.eval	0	mu=250:lambda=0.6.eval	7.00E-04
OHSU-TEST5	k1=0.7:b=0.5.eval	0.0982	500.eval	0.0913	lambda=0.7.eval	0.0643	mu=250:lambda=0.6.eval	0.0868
OHSU-TEST7	k1=0.7:b=0.5.eval	0	500.eval	0	lambda=0.7.eval	0	mu=250:lambda=0.6.eval	0
OHSU-TEST8	k1=0.7:b=0.5.eval	0.0175	500.eval	**0.0203**	lambda=0.7.eval	0.0089	mu=250:lambda=0.6.eval	**0.021**
OHSU-TEST9	k1=0.7:b=0.5.eval	0	500.eval	0	lambda=0.7.eval	0	mu=250:lambda=0.6.eval	0
OHSU1	k1=0.7:b=0.5.eval	0.1978	500.eval	**0.2136**	lambda=0.7.eval	0.1919	mu=250:lambda=0.6.eval	**0.2157**

OHSU10	k1=0.7:b=0.5.eval	0.1683	500.eval	0.138	lambda=0.7.eval	0.1184	mu=250:lambda=0.6.eval	0.1371
OHSU11	k1=0.7:b=0.5.eval	0.2714	500.eval	0.2335	lambda=0.7.eval	0.2166	mu=250:lambda=0.6.eval	0.2162
OHSU12	k1=0.7:b=0.5.eval	0.5681	500.eval	**0.5692**	lambda=0.7.eval	**0.6**	mu=250:lambda=0.6.eval	**0.5797**
OHSU13	k1=0.7:b=0.5.eval	0.3131	500.eval	0.2962	lambda=0.7.eval	0.2169	mu=250:lambda=0.6.eval	0.2643
OHSU14	k1=0.7:b=0.5.eval	0.3825	500.eval	0.3546	lambda=0.7.eval	0.2714	mu=250:lambda=0.6.eval	0.3519
OHSU15	k1=0.7:b=0.5.eval	0.1553	500.eval	**0.2467**	lambda=0.7.eval	**0.2296**	mu=250:lambda=0.6.eval	**0.2442**
OHSU16	k1=0.7:b=0.5.eval	0.1987	500.eval	0.1674	lambda=0.7.eval	0.1554	mu=250:lambda=0.6.eval	0.1757
OHSU17	k1=0.7:b=0.5.eval	0.436	500.eval	**0.4677**	lambda=0.7.eval	0.2697	mu=250:lambda=0.6.eval	0.4261
OHSU18	k1=0.7:b=0.5.eval	0.3092	500.eval	**0.3346**	lambda=0.7.eval	0.2672	mu=250:lambda=0.6.eval	**0.3239**
OHSU19	k1=0.7:b=0.5.eval	0.7802	500.eval	**0.7808**	lambda=0.7.eval	0.7756	mu=250:lambda=0.6.eval	0.7796
OHSU2	k1=0.7:b=0.5.eval	0.38	500.eval	**0.4186**	lambda=0.7.eval	0.2564	mu=250:lambda=0.6.eval	**0.4064**
OHSU20	k1=0.7:b=0.5.eval	0.607	500.eval	**0.6242**	lambda=0.7.eval	**0.6596**	mu=250:lambda=0.6.eval	**0.6292**
OHSU21	k1=0.7:b=0.5.eval	0.4549	500.eval	**0.4568**	lambda=0.7.eval	0.3549	mu=250:lambda=0.6.eval	**0.4636**
OHSU22	k1=0.7:b=0.5.eval	0.1981	500.eval	0.1747	lambda=0.7.eval	0.16	mu=250:lambda=0.6.eval	0.1852
OHSU23	k1=0.7:b=0.5.eval	0.5543	500.eval	0.4215	lambda=0.7.eval	0.4487	mu=250:lambda=0.6.eval	0.4242
OHSU24	k1=0.7:b=0.5.eval	0.606	500.eval	0.4991	lambda=0.7.eval	0.3824	mu=250:lambda=0.6.eval	0.5521
OHSU25	k1=0.7:b=0.5.eval	0.0466	500.eval	**0.0703**	lambda=0.7.eval	**0.0569**	mu=250:lambda=0.6.eval	**0.0711**
OHSU26	k1=0.7:b=0.5.eval	0.8438	500.eval	0.829	lambda=0.7.eval	0.8408	mu=250:lambda=0.6.eval	0.8305
OHSU27	k1=0.7:b=0.5.eval	0.6065	500.eval	**0.6296**	lambda=0.7.eval	0.5555	mu=250:lambda=0.6.eval	**0.6381**
OHSU28	k1=0.7:b=0.5.eval	0.0867	500.eval	0.0803	lambda=0.7.eval	0.0681	mu=250:lambda=0.6.eval	0.0844
OHSU29	k1=0.7:b=0.5.eval	0.4462	500.eval	0.4381	lambda=0.7.eval	**0.4798**	mu=250:lambda=0.6.eval	0.4447
OHSU3	k1=0.7:b=0.5.eval	0.5177	500.eval	0.4651	lambda=0.7.eval	0.4347	mu=250:lambda=0.6.eval	0.4797
OHSU30	k1=0.7:b=0.5.eval	0.5813	500.eval	0.5806	lambda=0.7.eval	0.4513	mu=250:lambda=0.6.eval	0.5674
OHSU31	k1=0.7:b=0.5.eval	0.1089	500.eval	0.0968	lambda=0.7.eval	0.1081	mu=250:lambda=0.6.eval	0.1002
OHSU32	k1=0.7:b=0.5.eval	0.1039	500.eval	0.076	lambda=0.7.eval	0.0598	mu=250:lambda=0.6.eval	0.083
OHSU33	k1=0.7:b=0.5.eval	0.2543	500.eval	0.151	lambda=0.7.eval	0.1089	mu=250:lambda=0.6.eval	0.1654
OHSU34	k1=0.7:b=0.5.eval	0.0517	500.eval	0.0436	lambda=0.7.eval	0.0431	mu=250:lambda=0.6.eval	0.0439
OHSU35	k1=0.7:b=0.5.eval	0.8056	500.eval	0.5475	lambda=0.7.eval	0.4332	mu=250:lambda=0.6.eval	0.7356
OHSU36	k1=0.7:b=0.5.eval	0.2005	500.eval	0.164	lambda=0.7.eval	0.1623	mu=250:lambda=0.6.eval	0.1589
OHSU37	k1=0.7:b=0.5.eval	0.2567	500.eval	0.0755	lambda=0.7.eval	0.0423	mu=250:lambda=0.6.eval	0.1038
OHSU38	k1=0.7:b=0.5.eval	0.6375	500.eval	0.5826	lambda=0.7.eval	0.5679	mu=250:lambda=0.6.eval	0.603
OHSU39	k1=0.7:b=0.5.eval	0.125	500.eval	0.0691	lambda=0.7.eval	0.0457	mu=250:lambda=0.6.eval	0.0655
OHSU4	k1=0.7:b=0.5.eval	0.6214	500.eval	0.5542	lambda=0.7.eval	0.5198	mu=250:lambda=0.6.eval	0.5606
OHSU40	k1=0.7:b=0.5.eval	0.4596	500.eval	0.3519	lambda=0.7.eval	0.189	mu=250:lambda=0.6.eval	0.41
OHSU41	k1=0.7:b=0.5.eval	0.3057	500.eval	0.2482	lambda=0.7.eval	0.2234	mu=250:lambda=0.6.eval	0.2549
OHSU42	k1=0.7:b=0.5.eval	0.1898	500.eval	0.1836	lambda=0.7.eval	0.1861	mu=250:lambda=0.6.eval	0.1892
OHSU43	k1=0.7:b=0.5.eval	0.7551	500.eval	0.7262	lambda=0.7.eval	0.6398	mu=250:lambda=0.6.eval	0.737
OHSU44	k1=0.7:b=0.5.eval	0.4242	500.eval	0.359	lambda=0.7.eval	0.3526	mu=250:lambda=0.6.eval	0.3724
OHSU45	k1=0.7:b=0.5.eval	0.0676	500.eval	0.034	lambda=0.7.eval	0.0107	mu=250:lambda=0.6.eval	0.0596
OHSU46	k1=0.7:b=0.5.eval	0.2438	500.eval	**0.2487**	lambda=0.7.eval	0.1712	mu=250:lambda=0.6.eval	**0.2513**
OHSU47	k1=0.7:b=0.5.eval	0.3565	500.eval	0.3483	lambda=0.7.eval	0.3376	mu=250:lambda=0.6.eval	**0.3644**
OHSU48	k1=0.7:b=0.5.eval	0.2768	500.eval	**0.4356**	lambda=0.7.eval	**0.3877**	mu=250:lambda=0.6.eval	**0.387**
OHSU49	k1=0.7:b=0.5.eval	0.2734	500.eval	0.2411	lambda=0.8.eval	0.1215	mu=250:lambda=0.6.eval	0.2288
OHSU5	k1=0.7:b=0.5.eval	0.6262	500.eval	0.5909	lambda=0.7.eval	0.5647	mu=250:lambda=0.6.eval	0.5992
OHSU50	k1=0.7:b=0.5.eval	0.3306	500.eval	0.2609	lambda=0.7.eval	0.2804	mu=250:lambda=0.6.eval	0.2603
OHSU51	k1=0.7:b=0.5.eval	0.3498	500.eval	0.3026	lambda=0.8.eval	0.1902	mu=250:lambda=0.6.eval	0.263
OHSU52	k1=0.7:b=0.5.eval	0.126	500.eval	**0.1618**	lambda=0.7.eval	0.1068	mu=250:lambda=0.6.eval	**0.1503**
OHSU53	k1=0.7:b=0.5.eval	0.0958	500.eval	**0.0967**	lambda=0.7.eval	0.0123	mu=250:lambda=0.6.eval	0.0768
OHSU54	k1=0.7:b=0.5.eval	0.3455	500.eval	0.2924	lambda=0.7.eval	0.3081	mu=250:lambda=0.6.eval	0.3254
OHSU55	k1=0.7:b=0.5.eval	0.0947	500.eval	**0.1603**	lambda=0.7.eval	**0.1134**	mu=250:lambda=0.6.eval	**0.1614**
OHSU56	k1=0.7:b=0.5.eval	0.1287	500.eval	**0.1358**	lambda=0.7.eval	0.1144	mu=250:lambda=0.6.eval	0.1004
OHSU57	k1=0.7:b=0.5.eval	0.3136	500.eval	0.3096	lambda=0.7.eval	**0.3194**	mu=250:lambda=0.6.eval	**0.3329**
OHSU58	k1=0.7:b=0.5.eval	0	500.eval	0	lambda=0.7.eval	0	mu=250:lambda=0.6.eval	0
OHSU59	k1=0.7:b=0.5.eval	0.3361	500.eval	0.3125	lambda=0.7.eval	0.3089	mu=250:lambda=0.6.eval	0.3126

OHSU6	k1=0.7:b=0.5.eval	0.1673	500.eval	0.1533	lambda=0.7.eval	0.1334	mu=250:lambda=0.6.eval	**0.1774**
OHSU60	k1=0.7:b=0.5.eval	0.1021	500.eval	0.013	lambda=0.7.eval	0.009	mu=250:lambda=0.6.eval	0.0199
OHSU61	k1=0.7:b=0.5.eval	0.0313	500.eval	**0.0612**	lambda=0.7.eval	**0.0675**	mu=250:lambda=0.6.eval	**0.0546**
OHSU62	k1=0.7:b=0.5.eval	0.1278	500.eval	0.0815	lambda=0.7.eval	0.1016	mu=250:lambda=0.6.eval	0.0958
OHSU63	k1=0.7:b=0.5.eval	7.00E-04	500.eval	**0.0018**	lambda=0.7.eval	**0.0011**	mu=250:lambda=0.6.eval	**0.0011**
OHSU7	k1=0.7:b=0.5.eval	0.2013	500.eval	0.1095	lambda=0.7.eval	0.1296	mu=250:lambda=0.6.eval	0.1375
OHSU8	k1=0.7:b=0.5.eval	0.0915	500.eval	0.0683	lambda=0.7.eval	0.0745	mu=250:lambda=0.6.eval	0.0687
OHSU9	k1=0.7:b=0.5.eval	0.5243	500.eval	0.4755	lambda=0.7.eval	0.4899	mu=250:lambda=0.6.eval	0.4789