2D SIMILARITY, DIVERSITY AND CLUSTERING IN RDKIT

Roger Sayle

NextMove Software, Cambridge, UK

8th RDKit UGM, University of Hamburg, Germany, Thursday 26th September 2019

BACK STORY

- Greg remembered that at the 6th RDKit UGM in 2017 I had mentioned that MaxMin Diversity selection and Taylor-Butina clustering were related, and both had efficient implementations…

MOTIVATION: COMPOUND ACQUISITION

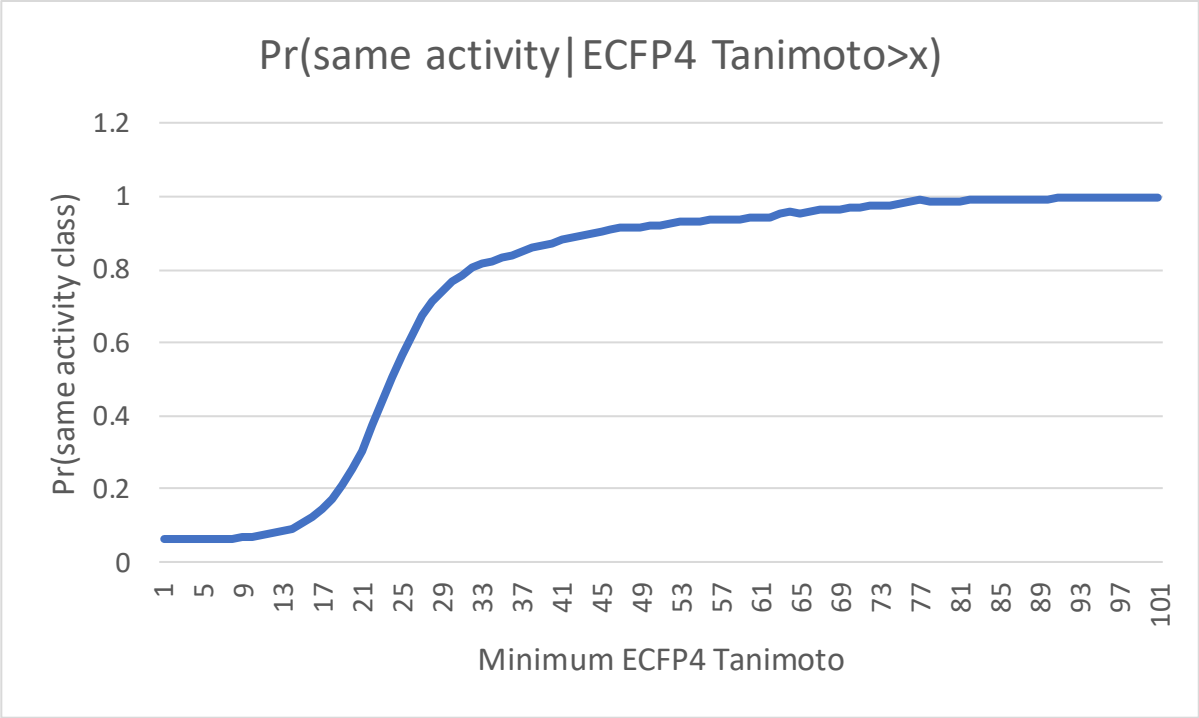
- Given a existing screening collection of *X* compounds, and with *Y* vendor compounds available for purchase, how should I select the next *Z* diverse compounds to buy.

- Typically, *X* is about 2M and *Y* is about 720M.

- Previously ~O(N^3), replaced with << O(N^2).

THE SIMILARITY PRINCIPLE

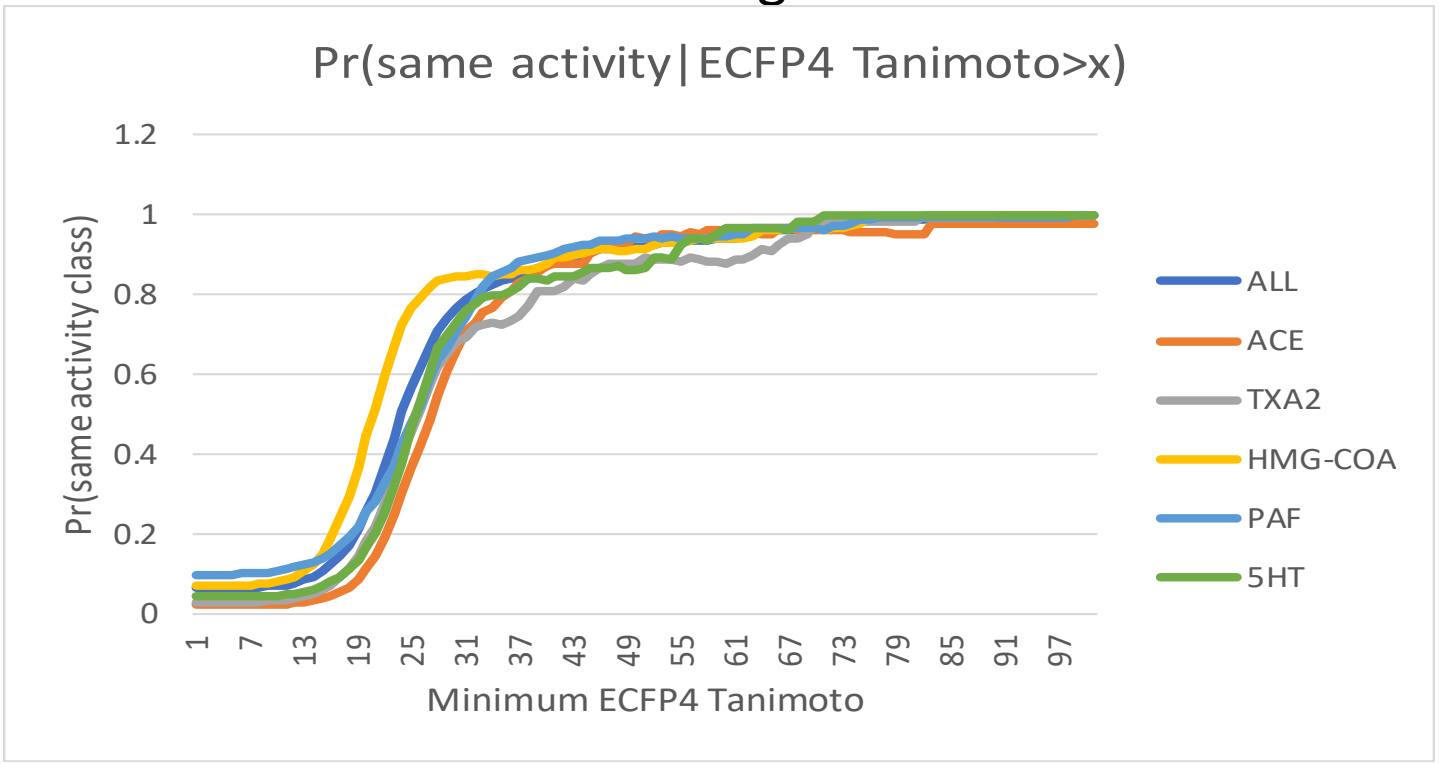
- Compounds with high Tanimoto values are likely to have similar biological activity.



- Briem & Lessel benchmark (Persp. Drug Discov. Des. 20:231, 2000)
- 380 actives (in 5 activity classes) and 574 decoys.

THE SIMILARITY PRINCIPLE

- Real-life differences can be ignored for the rest of this talk.



- The take home is that 0.35 is a reasonable threshold for ECFP4 (and 0.7 is not!).
- The slope of this curve is important in classification, and not its location.

RDKIT'S MAXMINPICKER

- RDKit's MaxMinPicker is described in a number of blog posts by Greg Landrum and Tim Dudgeon:
  - Picking diverse compounds from large sets, 2014/08
  - Optimizing Diversity Picking in the RDKit, 2014/08
  - Revisiting the MaxMinPicker, 2017/11

- M. Ashton, J. Barnard, P. Willett et al., "Identification of Diverse Database Subsets using Property-based and Fragment-based Molecular Descriptors", Quant. Struct.-Act. Relat., Vol. 21, pp. 598-604, 2002.
- R. Kennard and L. Stone, "Computer aided design of experiments", Technometrics, 11(1), pp. 137-148, 1969.

SELECTION VISUALIZATION

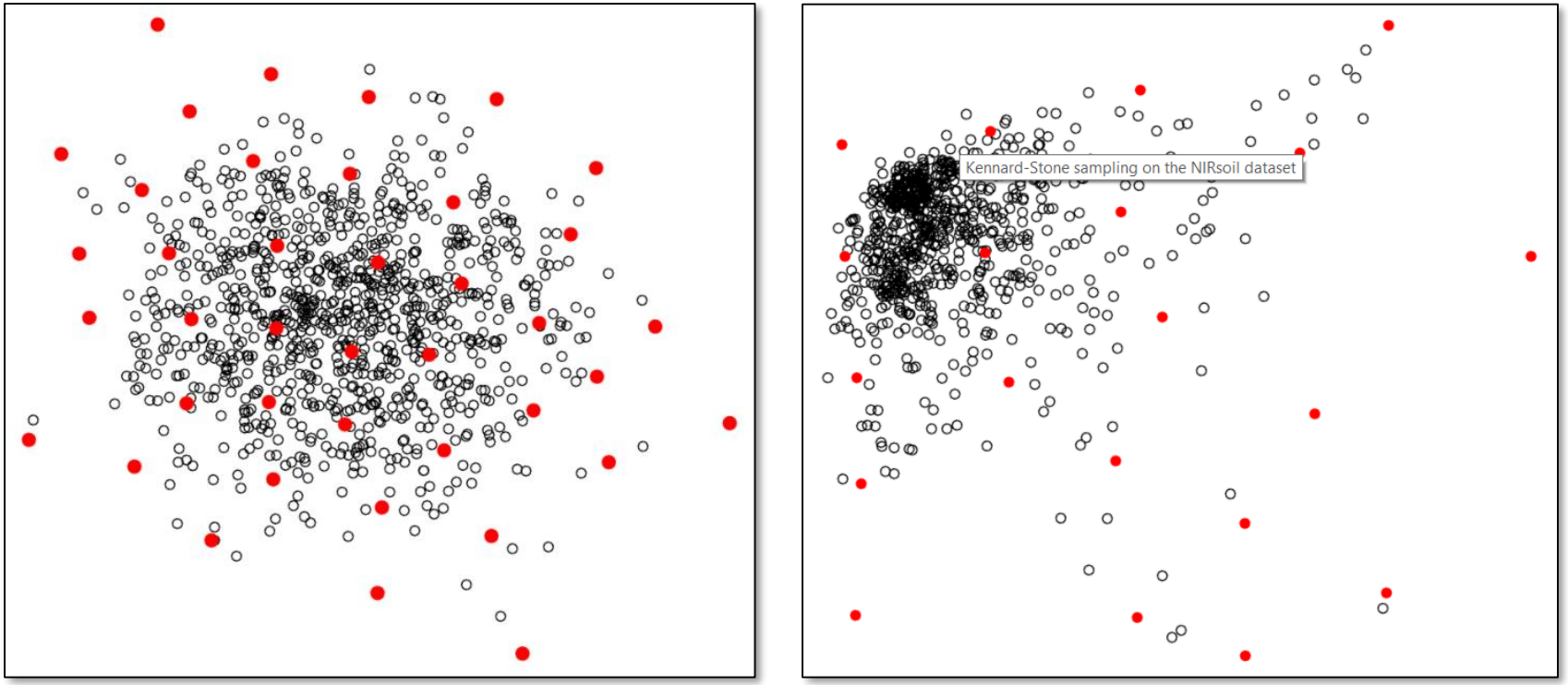


Image Credits: Antoine Stevens, the ProspectR package on github

MAXMIN CONCEPTUAL ALGORITHM

- If no compounds have been picked so far, choose the first picked compound at random.
- Repeatedly select the compound furthest from it's nearest picked compound [hence the name maximum-minimum distance].
- Continue until the desired number of picked compounds has been selected (or the pool of available compounds has been exhausted).

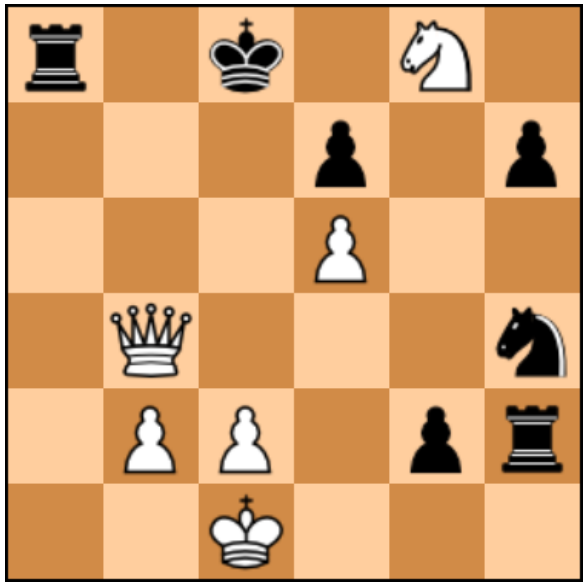
MAXMIN AND MINIMAX

- "Minimax is a decision rule used in AI"

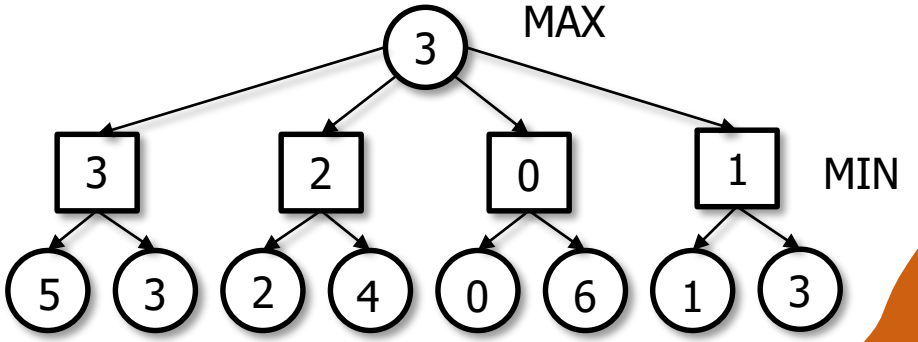
  - Wikipedia

- A miracle of computer science/mathematics:
- $\max_i(\max_j(a_{ij}))$ and $\min_i(\min_j(a_{ij}))$ require testing every item a_{ij}.
- $\min_i(\max_j(a_{ij}))$ and $\max_i(\min_j(a_{ij}))$ can be calculate more efficiently.

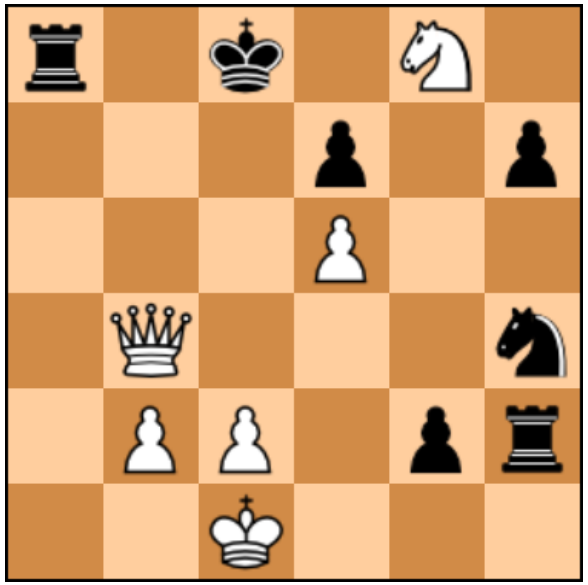
MINIMAX IN CHESS AI



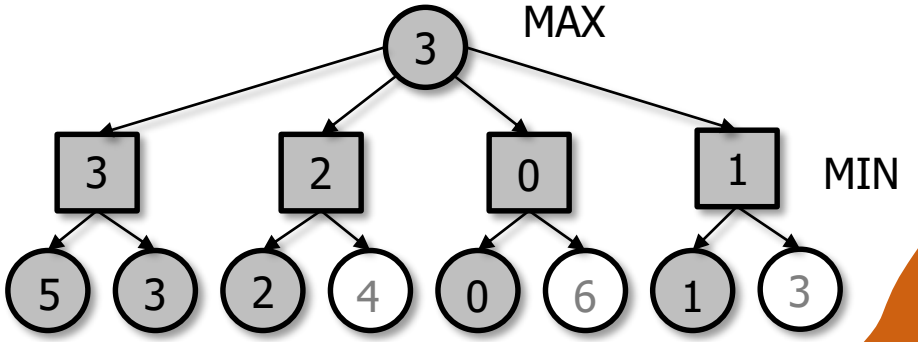
Los Alamos Chess (6x6 board)
White has 16 possible moves.
The 10 that don't check, lose.
Five checks, lose the queen.



MINIMAX IN CHESS AI



Los Alamos Chess (6x6 board)
White has 16 possible moves.
The 10 that don't check, lose.
Five checks, lose the queen.



Alpha cut-offs allow us to prune the search tree.

CLASSIC MAXMIN PICKING

		Candidate Pool									
		0	1	2	3	4	5	6	7	8	9
Picks	0	3	1	4	1	5	9	2	6	5	3
	1	5	8	9	7	9	3	2	3	8	4
	2	6	2	6	4	3	3	8	3	2	7
	3	9	5	0	2	8	8	4	1	9	7
Minimums		3	1	0	1	3	3	2	1	2	3
Maximum		3									

CURRENT MAXMIN PICKING

Picks	Candidate Pool 0	1	2	3	4	5	6	7	8	9
0	**3**	**1**	**4**	**1**	**5**	**9**	**2**	**6**	**5**	**3**
1	**5**	8	**9**	7	**9**	**3**	2	**3**	**8**	4
2	**6**	2	**6**	4	**3**	3	8	3	**2**	7
3	**9**	5	**0**	2	8	8	4	1	9	7
< Bounds	3	1	0	1	3	3	2	3	2	3
Maximum	3									

ITERATIVE MAXMIN PICKING

		Candidate Pool								
		1	2	3	4	5	6	7	8	9
Picks 0	3	**1**	**4**	**1**	**5**	**9**	**2**	**6**	**5**	**3**
Picks 1	5	**8**	**9**	7	**9**	**3**	2	**3**	**8**	4
Picks 2	6	**2**	**6**	4	**3**	3	8	3	**2**	7
Picks 3	9	**5**	**0**	2	**8**	8	4	1	9	7
Picks 0		**1**	6	9	**3**	9	9	3	7	5
< Bounds		1	0	1	3	3	2	3	2	3
Maximum					3					

PERFORMANCE IMPROVEMENT

- Using Andrew Dalke's data set of 12386 benzodiazepines, using the first one thousand as the existing screening library, select the next 18 most diverse molecules from the remaining set (of 11386).
  - Original RDKit Implementation:
    - 224,688,273 FP comparisons 96.30s
  - Pruning using alpha cut-off:
    - 16,069,573 FP comparisons 6.79s (14x)
  - Preserving bounds across picks:
    - 1,047,982 FP comparisons 0.46s (209x)
  - Timings on Dell laptop, using 2048 bit Morgan radius 2 FPs.

SCREENING LIBRARY ENHANCEMENT

- Selecting 500 compounds to purchase from Enamine REAL 2017 (171M) to enhance ChEMBL 23 (1.7M).
  - Reading mols/FP gen: 77194s + 1204s
  - 1st compound: 181.32B FP cmps (1/82750).
  - 10th compound: 301.67B FP cmps.
  - 50th compound: 469.46B FP cmps.
  - 100th compound: 748.97B FP cmps.
  - 500th compound: 2446.80B FP cmps.
- A traditional distance matrix requires 60 petabytes of storage, and 1.5E16 FP comparisons (15 quadrillion).

TRACKING PROGRESS OF MAXMIN

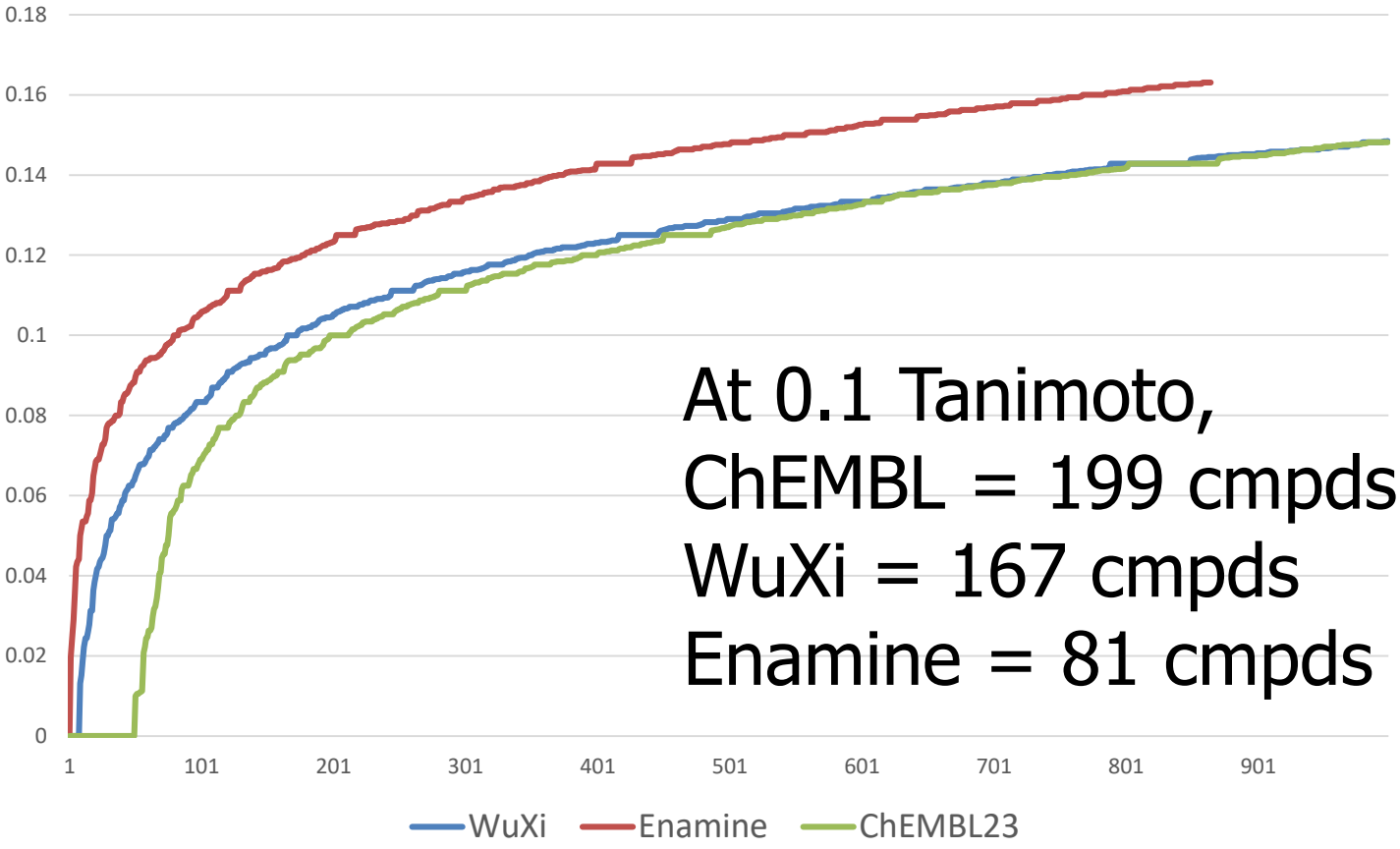
threshold	clusters	1870461	comparisons	1749.3T
0.20	2294	0.12%	702845018	0.04%
0.25	8935	0.48%	3253794011	0.19%
0.30	26131	1.40%	11549883033	0.66%
0.35	56353	3.01%	29091751345	1.66%
0.40	99149	5.30%	57519725846	3.29%
0.45	157651	8.43%	101629196314	5.81%
0.50	224670	12.01%	158848339798	9.08%
0.55	322770	17.26%	249793024261	14.28%
0.60	435222	23.27%	364077628967	20.81%
0.65	585325	31.29%	527089593168	30.13%
0.70	762874	40.79%	731808136660	41.83%
0.75	969454	51.83%	973312937318	55.64%
0.80	1207052	64.53%	1251324722214	71.53%
0.85	1433204	76.62%	1490057349953	85.18%
0.90	1587024	84.85%	1623213653052	92.79%
0.95	1703806	91.09%	1696761101537	97.00%
1.00	1752412	93.69%	1719216584093	98.28%

MAXMIN TO ANALYSE DATA SETS

- The rate at which molecules are picked and their thresholds can be used to assess data set diversity.
- WuXi screening: 215M
- Enamine REAL: 680M
- ChEMBL 23: 1.7M

Count	WuXi	Enamine	ChEMBL23
2	0	0	0
3	0	0.019608	0
4	0	0.025	0
5	0	0.028986	0
6	0	0.035294	0
7	0	0.042254	0
8	0	0.043478	0
9	0	0.044118	0
10	0.013158	0.05	0
11	0.015152	0.051724	0
12	0.018868	0.053571	0
13	0.022222	0.053571	0
14	0.02439	0.053571	0
15	0.02439	0.054795	0
16	0.025974	0.055556	0
17	0.027778	0.058824	0
18	0.03125	0.058824	0
19	0.03125	0.060606	0
20	0.036364	0.064935	0
21	0.038462	0.066667	0
22	0.04	0.068493	0
23	0.041667	0.068966	0
24	0.042254	0.068966	0
25	0.043478	0.070175	0

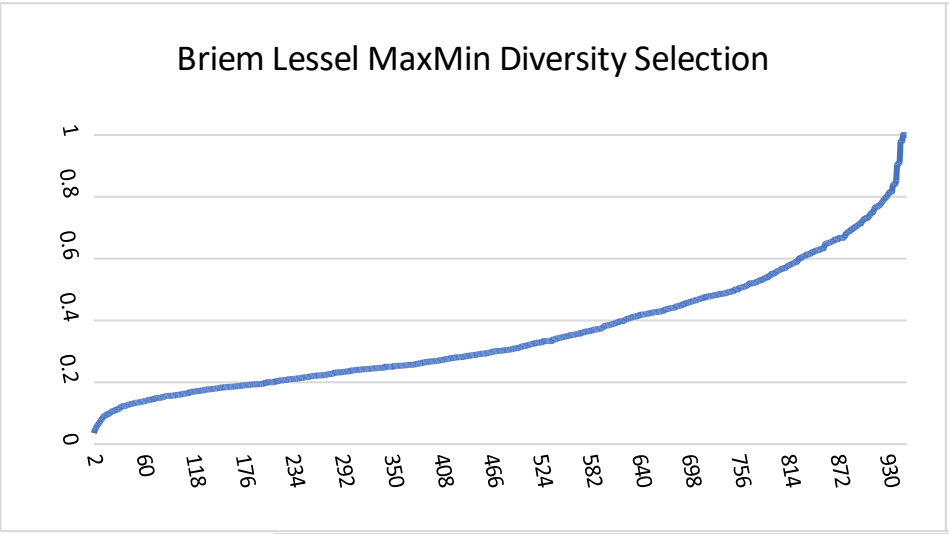
MAXMIN EARLY DIVERSITY ANALYSIS

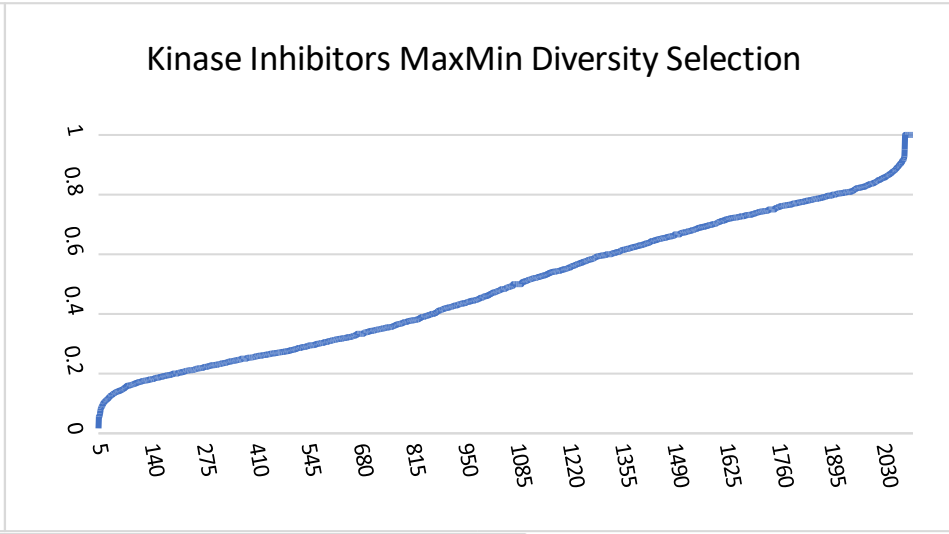


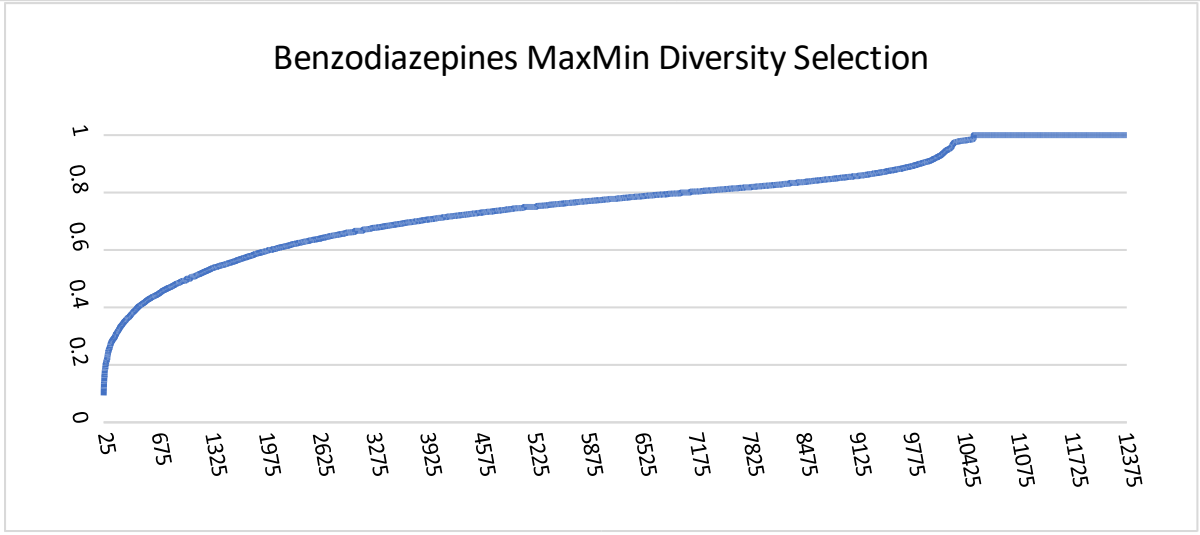
At 0.1 Tanimoto,
ChEMBL = 199 cmpds
WuXi = 167 cmpds
Enamine = 81 cmpds

WuXi covers ~2x space of Enamine, even though Enamine is 3x bigger.

MAXMIN FULL ANALYSIS



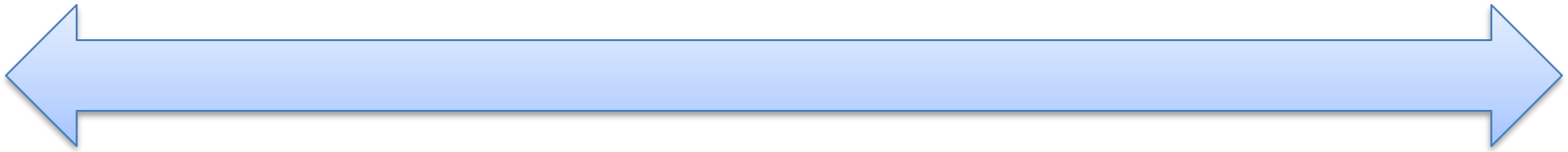




THE USES OF CLUSTERING

- "Round like a circle in a spiral, like a wheel with a wheel…
  Like the circles that you find in the windmills of your mind!"
- The clustering that we are interesting in are for coverage, with bounds on distance to "core points", hence DISE, Taylor-Butina and MaxMin clustering.
- For classification/grouping (of chemotypes), long stringy clusters may be appropriate, using methods such as DBSCAN and Jarvis-Patrick.
- Cluster membership in DBSCAN and Jarvis-Patrick may be calculated efficiently on-the-fly.

COVERAGE CLUSTERING SPECTRUM



Taylor-Butina	MaxMin	K-Means
Specify radius	Iterate radius/clusters	Specify clusters
Bounds on cluster	Bounds on cluster	No bounds on cluster
Can be seeded	Can be seeded	Can be seeded
Traditionally $O(n^2)$	Traditionally $O(n^3)$	Traditionally $O(n^2)$
Now $<<O(n^2)$	Now $<<O(n^2)$	Now $<O(n^2)$

MAXMIN VS. SPHERE EXCLUSION

- MaxMin and Butina-Taylor (sphere exclusion) clustering produce equivalent results.
  - For sphere exclusion, one specifies the threshold radius, and the method returns the number of clusters.
  - For MaxMin, the method returns the threshold required for each number of clusters.
- Historically, MaxMin was $O(n^3)$ and sphere exclusion was $O(n^2)$, but both are now < (or <<) $O(n^2)$.
- So how do these two methods compare?
  - Darko Butina, JCICS 39(4):747-750, 1999.
  - Alberto Gobbi and Man-Ling Lee, JCICS 43(1):317-323.

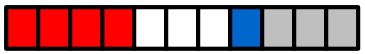
VARIANTS OF LEADER ALGORITHM

- Several variants of sphere exclusion cluster exist.
- All follow the “leader selection” approach but differ in the order in which the input is presented.
- The simplest variants use the provided input ordering or variants thereof.
- Taylor-Butina clustering orders the input to produce large clusters first.
- Alberto Gobbi’s “Directed Sphere Exclusion (DISE) order compounds by similarity to sildenafil!
- Additionally I’d recommend order by price!

SEQUENTIAL ALGORITHMS

- **Check-Past Variant**

```
unsigned int sphere_exclusion(T *db,
      unsigned int N,  T *cluster)
{ unsigned int clusters = 0;
  for (unsigned int i=0; i<N; i++) {
    bool found = false;
    for (unsigned int j=0; j<clusters; j++)
      if (tanimoto(db[i],cluster[j]) >= threshold) {
        found = true;
        break;
      }
    if (!found)
      cluster[clusters++] = db[i];
  }
  return clusters;
}
```



- **Filter-Future Variant**

```
unsigned int compact(T seed, T *src, T *dst,
                                unsigned int n)
{ unsigned int count = 0;
  for(unsigned int j=0; j<n; ++)
    if (tanimoto(seed,src[j]) < threshold) {
      dst[count++] = src[j];
  return count;
}
unsigned int sphere_exclusion(T *db,
      unsigned int N, T *cluster)
{ unsigned int clusters = 0;
  unsigned int left = N;
  for (unsigned int i=1; i<=left; i++) {
    left = i+compact(db[i-1],db+i,db+i,left-i);
    cluster[clusters++] = db[i];
  }
  return clusters;
}
```

PARALLEL ALGORITHM

- Although both “check-past” and “filter-future” sequential algorithms are equally efficient, requiring the same number of comparisons (<<$O(N^2)$) on a single processor, the filter-future variant is more suitable for parallel execution.
- This concept is effective parallelism vs. redundancy.
- The remaining challenge is called “stream compaction” in parallel computing.
- In python: lst=filter(lambda x: tanimoto(x,seed)<thresh,lst)

  or lst = [x for x in lst if tanimoto(x,seed)<thresh]

COROUTINES WITH PTHREADS' BARRIER

- **Inefficient idiom**

```
void *work(void *arg);

pthread_t tid[NPROCS];

for (unsigned int i=0; i<N; i++) {
  for (unsigned long n=0; n<NPROCS; n++)
    pthread_create(&tid[n],NULL,work,(void*)n);
  for (unsigned int n=0; n<NPROCS; n++)
    pthread_join(tid[n],0);
}
```

```
pthread_barrier_t wait, done;
bool all_done = false;
void *task(void *arg) {
  for(;;) {
    pthread_barrier_wait(&wait);
    if (all_done) return (void*)0;
    work(arg);
    pthread_barrier_wait(&done);
  }
}

pthread_barrier_init(&wait,NULL,NPROCS+1);
pthread_barrier_init(&done,NULL,NPROCS+1);
for (unsigned long n=0; n<NPROCS; n++)
    pthread_create(&tid[n],NULL,task,(void*)n);
for (unsigned int i=0; i<N; i++) {
  pthread_barrier_wait(&wait);
  pthread_barrier_wait(&done);
}
all_done = true;
pthread_barrier_wait(&wait);
for (unsigned int n=0; n<NPROCS; n++)
  pthread_join(tid[n],0);
```

PROCESSING A LINKED LIST IN PARALLEL

- **Sequential Implementation**

```
unsigned int next[];
unsigned int head;

unsigned int curr = head;
do {
  work(curr);
} while (curr = next[curr]);
```

- **Parallel Implementation**

```
void thread_work(unsigned int id)
{
  unsigned int cycle = id;
  unsigned int curr = head;
  do {
    if (cycle == 0) {
      work(curr);
      cycle = NPROCS-1;
    } else cycle--;
  } while (curr = next[curr]);
}
```

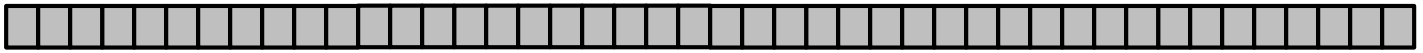
LOCK-FREE LAZY STREAM COMPACTION

- The input array is conceptually organized into blocks of a fixed number of elements, say 10K each.
- Each block is processed by a single processor at a time, and internally uses a sequential compaction algorithm (*dst++=*src++) and it's current length.
- Blocks are organized as (singly) linked lists.

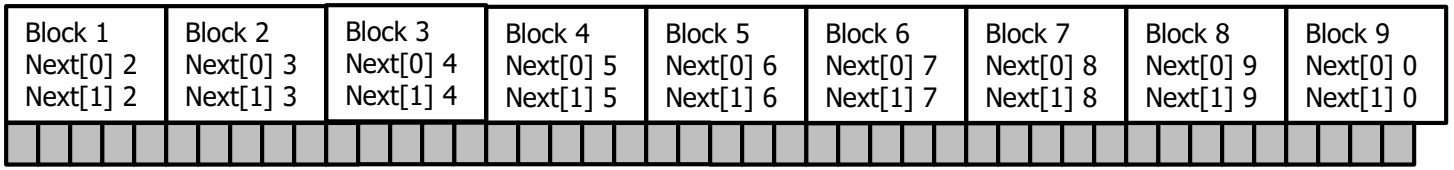
LOCK-FREE LAZY STREAM COMPACTION

- The input array is conceptually organized into blocks of a fixed number of elements, say 10K each.
- Each block is processed by a single processor at a time, and internally uses a sequential compaction algorithm (*dst++=*src++) and it's current length.
- Blocks are organized as (singly) linked lists.
- On each iteration, threads process each consecutive pair of blocks, allowing them to be coalesced.
- Race-conditions/synchronization are avoided using "tick-tock" multi-buffering of block next pointers.

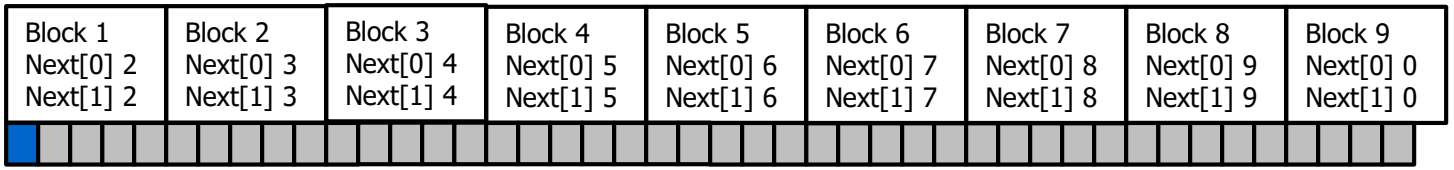
SMP SPHERE EXCLUSION ANIMATION



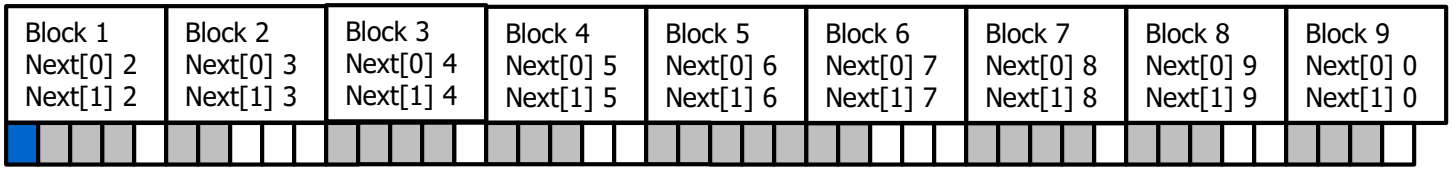
SMP SPHERE EXCLUSION ANIMATION



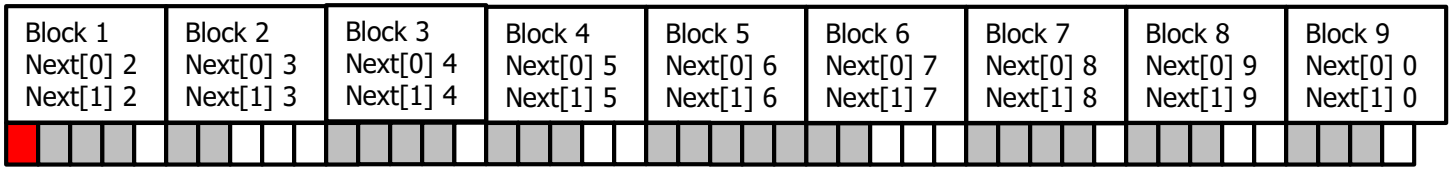
SMP SPHERE EXCLUSION ANIMATION



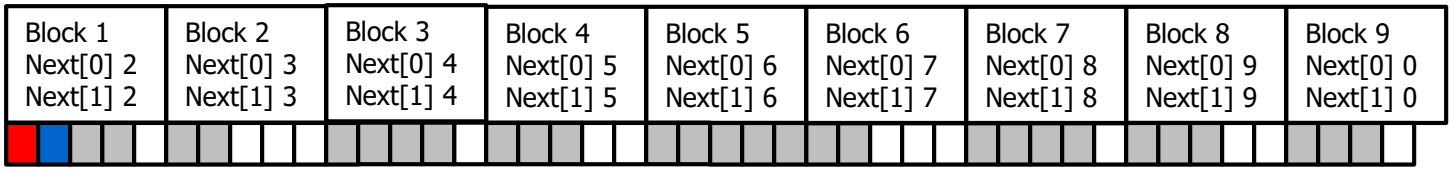
SMP SPHERE EXCLUSION ANIMATION



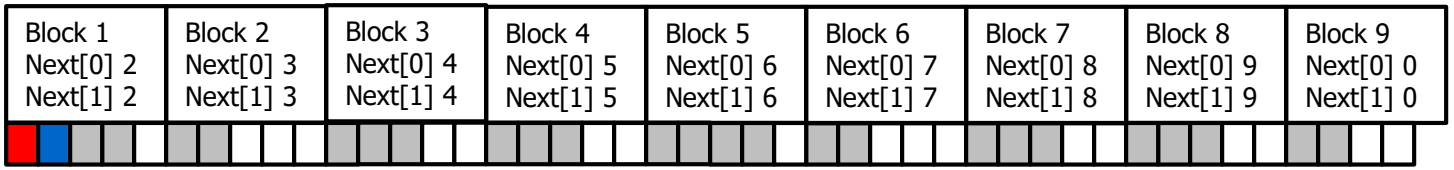
SMP SPHERE EXCLUSION ANIMATION



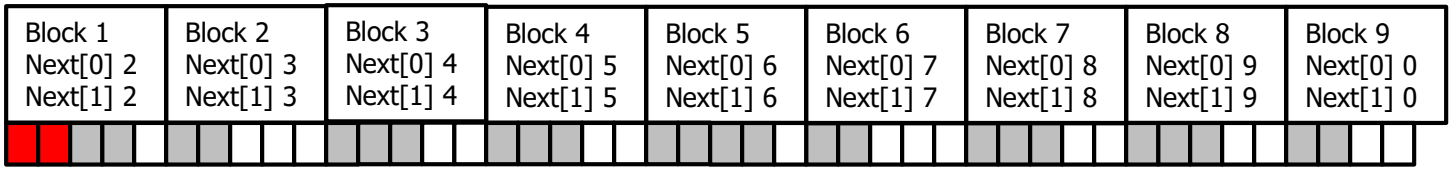
SMP SPHERE EXCLUSION ANIMATION



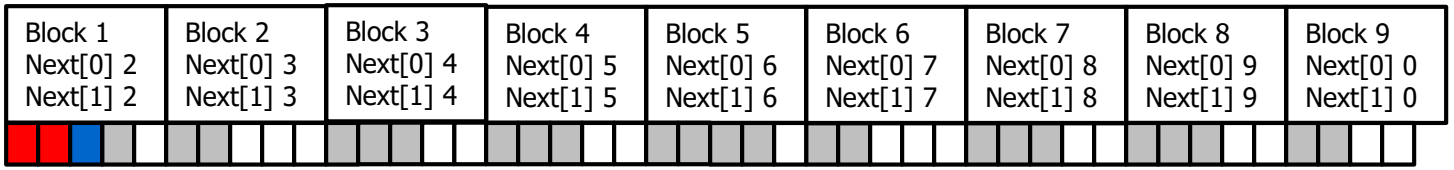
SMP SPHERE EXCLUSION ANIMATION



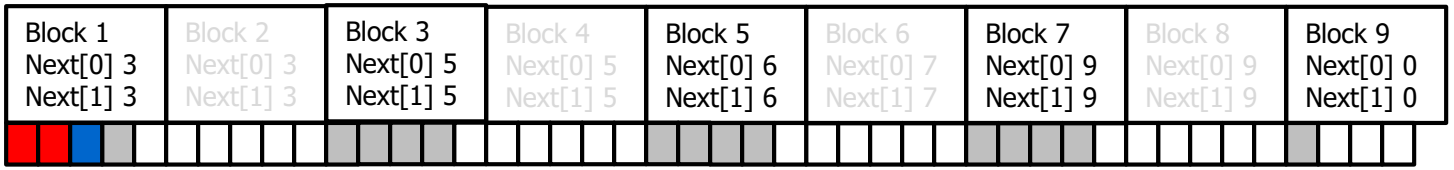
SMP SPHERE EXCLUSION ANIMATION



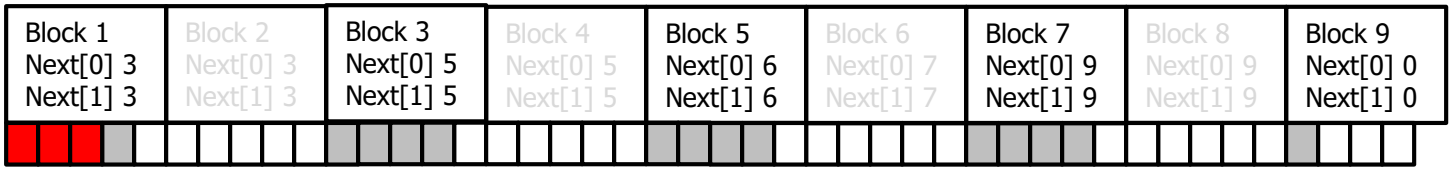
SMP SPHERE EXCLUSION ANIMATION



SMP SPHERE EXCLUSION ANIMATION



SMP SPHERE EXCLUSION ANIMATION



HEAD-TO-HEAD RESULTS (CHEMBL25)

	MaxMin				Sphere Exclusion				
Radius	Clusters	1870461	Comparisons	Fract	Clusters	1870461	Comparisons	Fract	Diff
0.05	45	0.00%	702845018	0.04%	38	0.11%	102463588	0.01%	18.42%
0.10	129	0.01%	3253794011	0.19%	123	0.42%	845022242	0.05%	4.88%
0.15	506	0.03%	11549883033	0.66%	456	1.26%	4887372526	0.28%	10.96%
0.20	2294	0.12%	702845018	0.04%	1996	0.11%	102463588	0.01%	14.93%
0.25	8935	0.48%	3253794011	0.19%	7925	0.42%	845022242	0.05%	12.74%
0.30	26131	1.40%	11549883033	0.66%	23491	1.26%	4887372526	0.28%	11.24%
0.35	56353	3.01%	29091751345	1.66%	51116	2.73%	17480894303	1.00%	10.25%
0.40	99149	5.30%	57519725846	3.29%	90618	4.84%	42127957348	2.41%	9.41%
0.45	157651	8.43%	101629196314	5.81%	145302	7.77%	82429903195	4.71%	8.50%
0.50	224670	12.01%	158848339798	9.08%	207837	11.11%	133346950697	7.62%	8.10%
0.55	322770	17.26%	249793024261	14.28%	299906	16.03%	213032407381	12.18%	7.62%
0.60	435222	23.27%	364077628967	20.81%	407264	21.77%	309049333081	17.67%	6.86%
0.65	585325	31.29%	527089593168	30.13%	552511	29.54%	442005491670	25.27%	5.94%
0.70	762874	40.79%	731808136660	41.83%	728662	38.96%	607768508390	34.74%	4.70%
0.75	969454	51.83%	973312937318	55.64%	938051	50.15%	809400682200	46.27%	3.35%
0.80	1207052	64.53%	1251324722214	71.53%	1184636	63.33%	1049631030937	60.00%	1.89%
0.85	1433204	76.62%	1490057349953	85.18%	1422992	76.08%	1234924388417	70.59%	0.72%
0.90	1587024	84.85%	1623213653052	92.79%	1583659	84.67%	1443068043947	82.49%	0.21%
0.95	1703806	91.09%	1696761101537	97.00%	1703268	91.06%	1565047511449	89.47%	0.03%
1.00	1752412	93.69%	1719216584093	98.28%	1752412	93.69%	1616082787269	92.38%	0.00%

EXECUTIVE SUMMARY

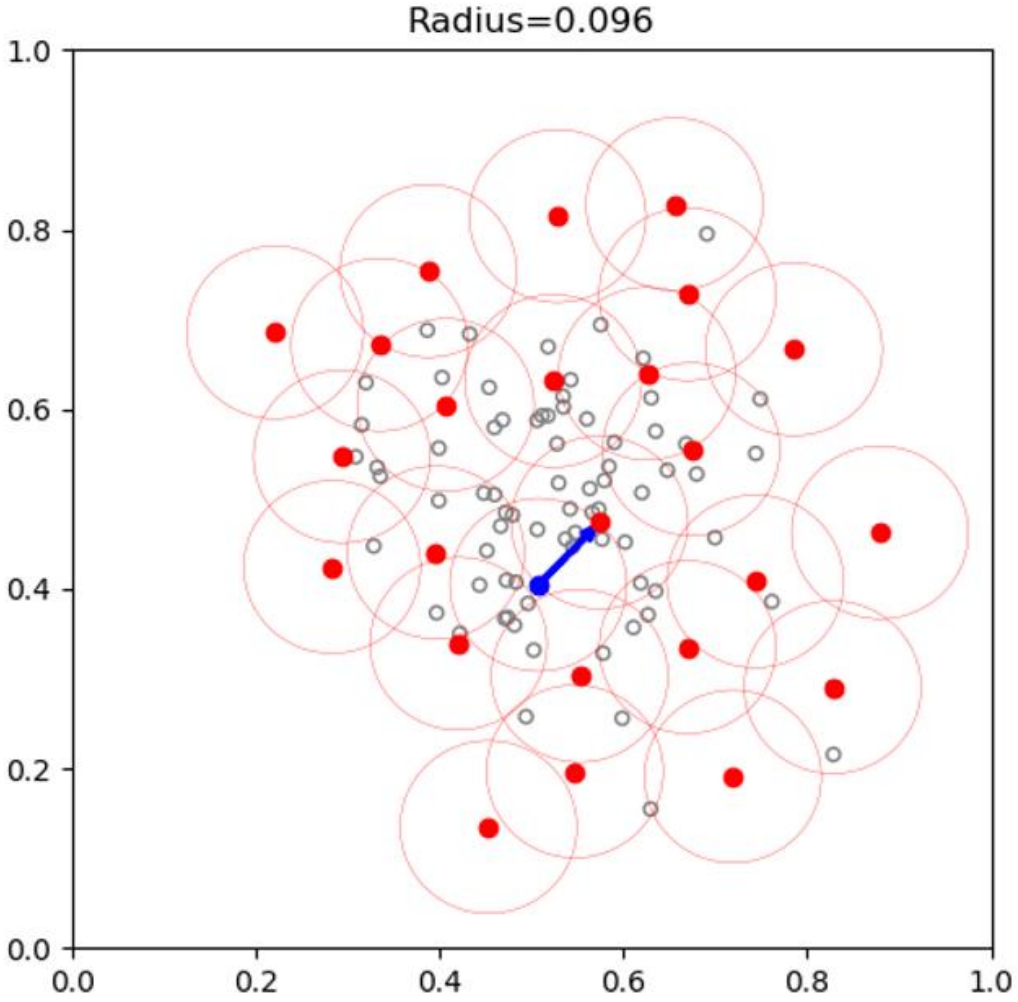
- In the threshold region we care about, MaxMin diversity selection is generating about 10% more clusters than Sphere exclusion.
- This selects about 3% of the original data set for about 1.5% of the comparisons required by all-vs-all.

SO WHAT'S HAPPENING?

- The difference is analogous to the difference between the molecular surface and solvent excluded volume in computational chemistry.
- MaxMin diversity selection extrapolates at the boundaries, where sphere exclusion finds the points that are sufficient to cover/interpolate a data set.

- Best results are obtained using MaxMin to determine a threshold followed by sphere exclusion (DISE).

SO WHAT'S HAPPENING?



STOCHASTIC STABILITY (FOR GREG)

- Input permutations on SE clusters. ChEMBL25 @ 0.35
  - Input 51116 17480894303
  - 1 50272 10624678066
  - 2 50338 10657316452
  - 3 50210 10594460327
  - 4 50214 10514469457
  - 5 50199 10660887976
  - 6 50281 10632572608
  - 7 50243 10641276910
  - 8 50449 10714607581
  - 9 50299 10583545862
  - 10 50144 10600053655

GRAND CHALLENGE: ENAMINE 720M

- All vs All: N.(N-1)/2 = 25862854423844900l cmps.
- At 400M cmps/s:~20.5 years, 1.0345 exabytes.
- MaxMin diversity selection of N=719205874 mols.
  - 0.05: 82 clusters, 747435533 cmps.
  - 0.10: 1630 clusters, 23317011669 cmps.
  - 0.15: 16272 clusters, 513682528576 cmps.
  - 0.20: 107838 clusters, 4312321854941 cmps.
  - 0.216: 187361 clusters, 8055848864691 cmps. [0.0031%]
- Sphere exclusion clustering on N=719205874 mols.
  - 0.35: 4077677 clusters, 23954793226736 cmps. [0.0093%]
    - Requires 4.5 days using 16 threads on a recent Intel workstation.

CONCLUSIONS

- Efficient algorithms exist for selecting representatives of large data sets without calculating the full distance matrix.
- For compound collections, sphere exclusion clustering is surprisingly effective, and Gobbi's heuristic is insightful (less crazy than it sounds), working from dense chemical space outwards.

ACKNOWLEDGEMENTS

- Thanks to…
  - Greg Landrum, T5 Informatics.
  - Roman Affentranger, Nicolas Zorn and Jerome Hert at Hoffman La-Roche, Basel, Switzerland..
  - John Mayfield, NextMove Software.
  - Noel O'Boyle, now at Sosei-Heptares.
  - Andrew Pannifer, Medicines Discovery Catapult.
  - Andrew Dalke, Dalke Scientific Software.

- I've skipped over some details(?!), so please feel free to ask questions at any time.

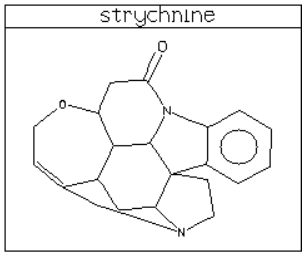
EFFICIENT TANIMOTO SIMILARITY

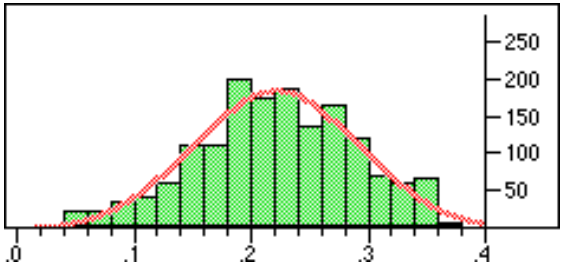
- Tanimoto(A,B) = $\frac{|A \cap B|}{|A \cup B|}$
- Tanimoto(A,B) = popcnt(A&B)/popcnt(A|B)
- Even with modern hardware, popcnt is significantly slower than than bit-wise AND and OR.
- When performing large numbers of comparisons, we can pre-compute the popcnts to reduce the number of popcnts performed in the critical loop.
- Tanimoto(A,B) = popcnt(A&B)/

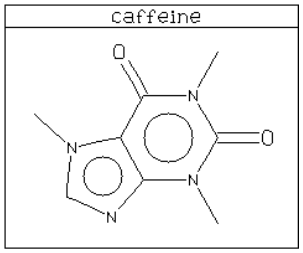
  (popnt(A)+popcnt(B)-popcnt(A&B))

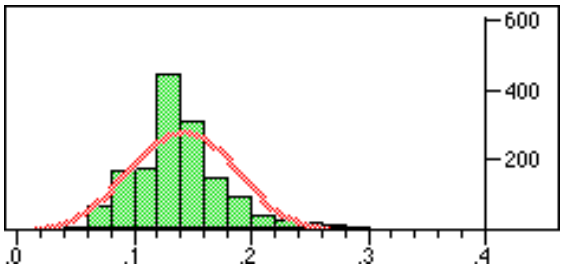
"THE MYTH OF 0.7"

- Bioinformaticians have long known that the significance of database hits in sequence searching depends on the substitution matrix used, the database composition and the query sequence.









- J.F. Collins, A.F.W. Coulson and A. Lyall, "The Significance of Protein Sequence Similarities", CABIOS, Vol. 4, No. 1, pp. 67-71, 1988.
- https://www.daylight.com/meetings/emug97/Bradshaw/Significant_Similarity/Significant_Similarity.html

LINGOS SIMILARITY ON BRIEM & LESSEL

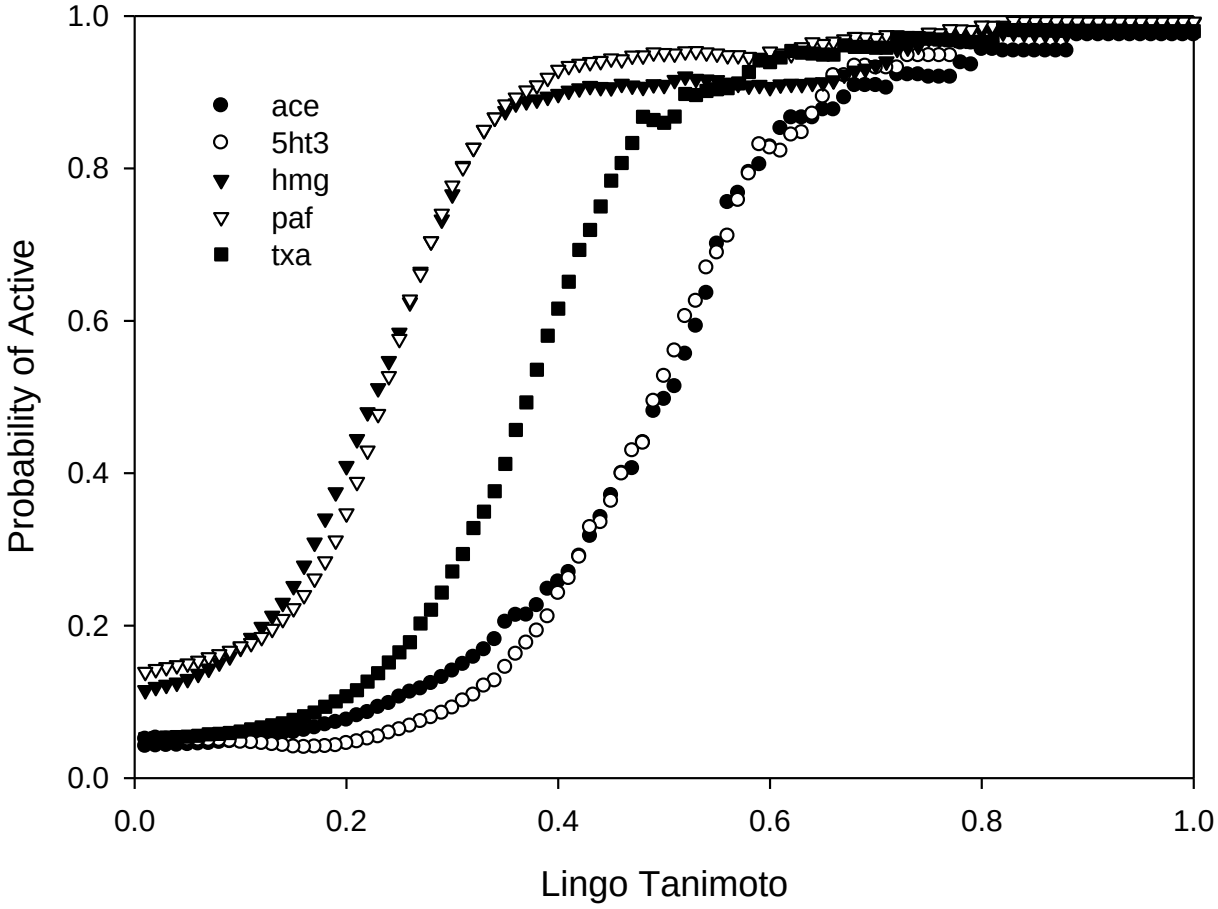


Image courtesy of Andrew Grant, AstraZeneca (2006)

WHY DISTANCE BOUNDS/SPHERES?

- For this use case, we'd hope to have representative compound in our screening collection with similar biological activity to any purchasable compound (as a proxy for any possible compound).
- The Martin-Brown/Brown-Martin hypothesis is 2D similar compounds (by Tanimoto) correlate with biological activity (see next slides).
- Hence, MaxMin and sphere exclusion clustering are more appropriate than Single Linkage, Jarvis-Patrick, DBScan, K-means or K-medians.