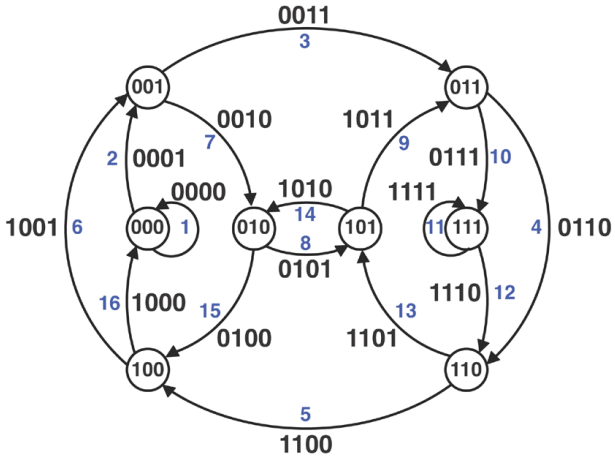
Sequence Design Problems in Discovery of Regulatory Elements

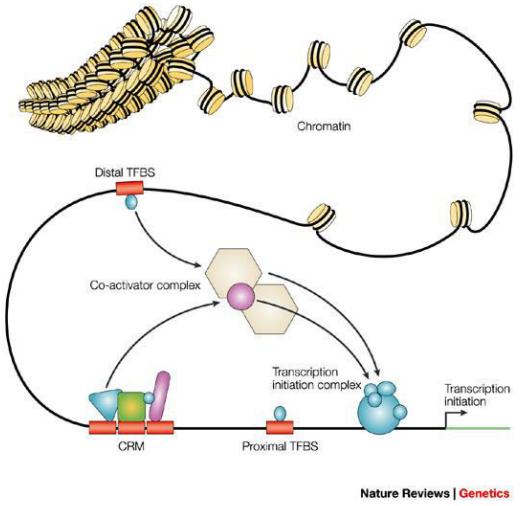
Yaron Orenstein, Bonnie Berger and Ron Shamir



Regulatory Genomics workshop

Simons Institute

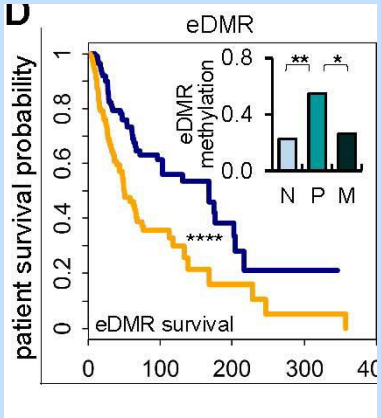
March 10th, 2016, Berkeley, CA

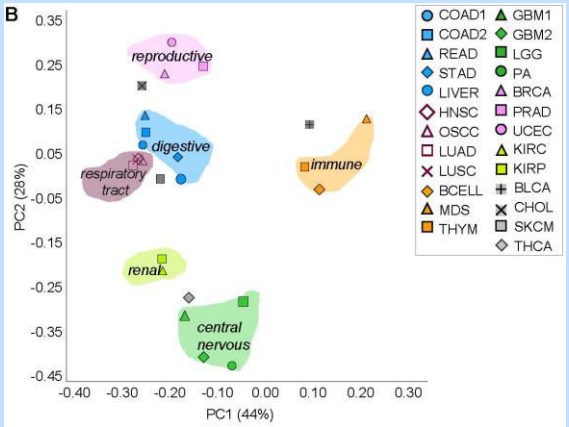


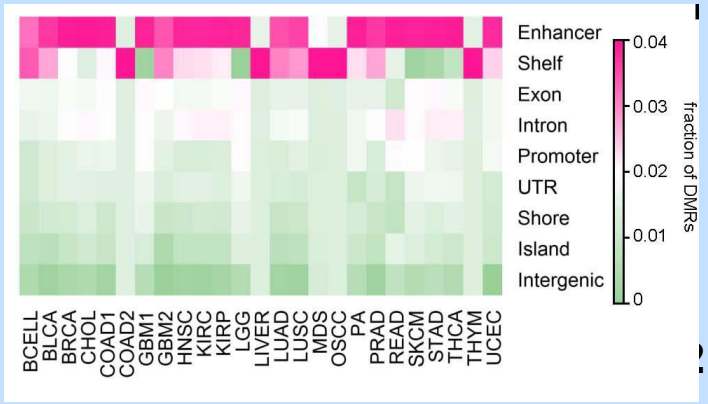
Differentially methylated enhancers in cancer

Bell et al. Genome Res 16

- Analyzed methylation patterns of 6200 tumors & normals from 25 cancer types
  - Enhancers show the most differential methylation patterns
  - Enhancer methylation patterns distinguish primary tumor types
  - Found enhancers whose methylation in metastatic melanoma correlated with patient mortality



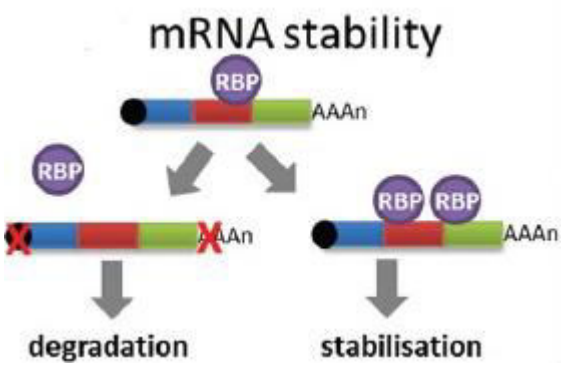


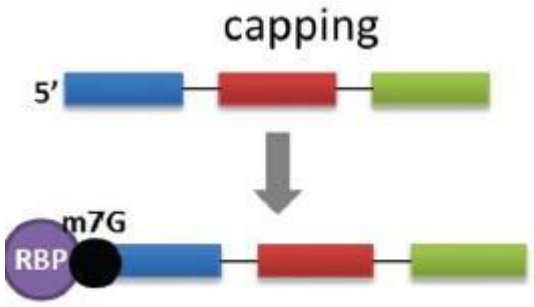


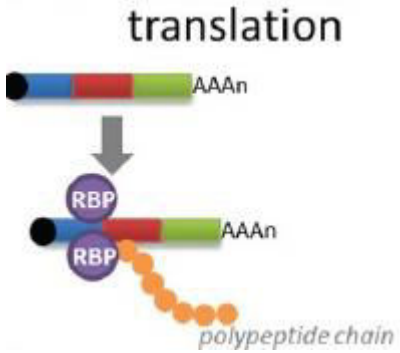
transcription and RNA regulators

- Gene expression regulation by transcription factors binding to DNA

- Post-transcriptional regulation by RNA binding proteins



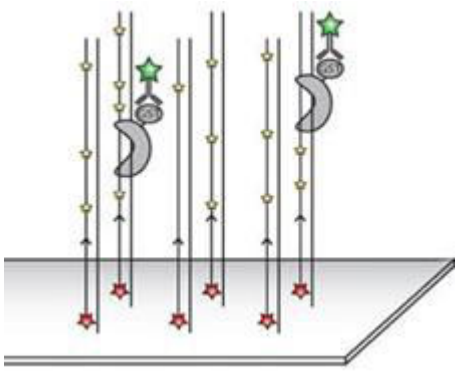




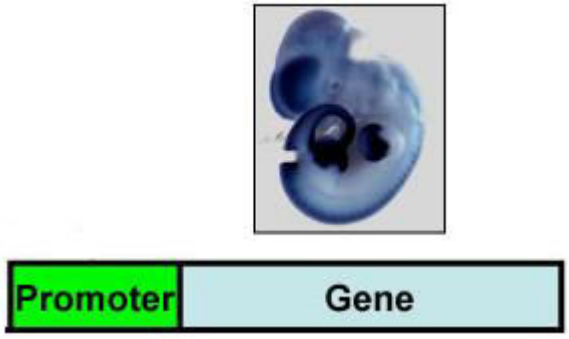
(Sutherland *et al.,* Asian Journal of Andrology. 2015)

Measuring TF binding

Protein binding Microarrays (Berger et al. 06)



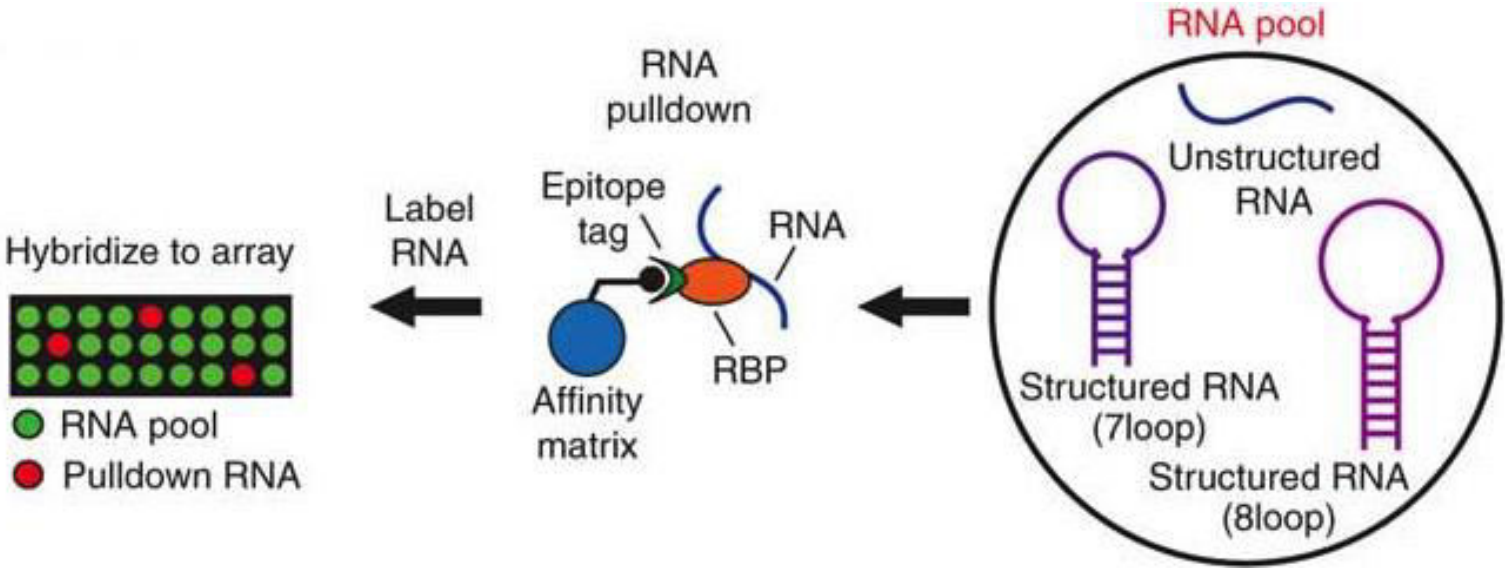
Synthetic enhancers (Smith et al. 12)



- Both technologies require designing a set of double stranded sequences that together cover all possible k-mers.

Measuring protein-RNA binding

- RNAcompete covers each 9-mer at least 16 times in unstructured RNA probes.



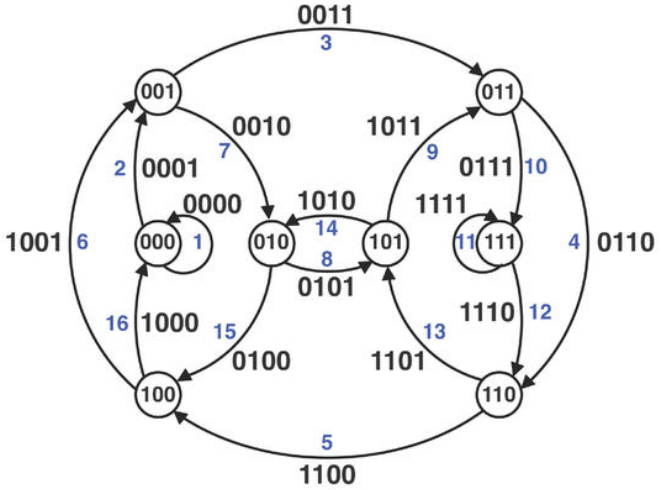
(Ray *et al.,* Nature Biotechnology 2009)

Require coverage of all RNA k-mers.

de Bruijn sequences

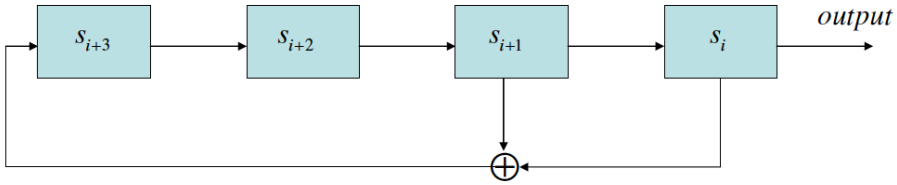
- Def: de Bruijn (dB) seq. of order k over Σ: Each k-mer appears exactly once.
  - Most compact. length = $|\Sigma|^k$.

de Bruijn graphs of order k-1



(Compeau *et al.* Nature Biotechnology 2011)

Linear feedback shift registers

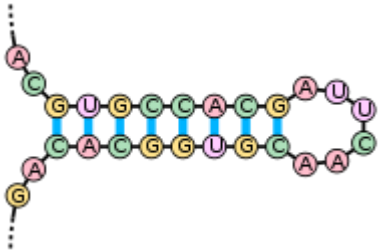


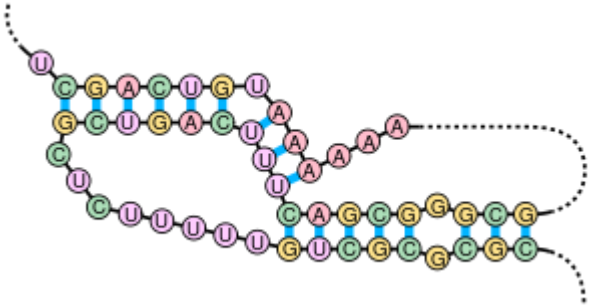
http://comp.ist.utl.pt/pdis-srm/

Using de Bruijn seqs is too naïve

- Redundancy in double-stranded DNA: by covering a k-mer, its reverse complement is covered too.

- Structured RNA probes: most random sequences are structured.





(Chen and Greider , *PNAS* 2005)

Challenge #1: cover all DNA k-mers in double-stranded probes

- Def:

S is a reverse complmentary dB (RCdB) sequence if ∀ k-mer W it includes W or RC(W).

- Goal:

  Generate a minimum length RCdB sequence

(Orenstein and Shamir, *Bioinformatics* 2013)

The RC Euler tour algorithm

- Form two reverse-complementary cycles in a de Bruijn graph.

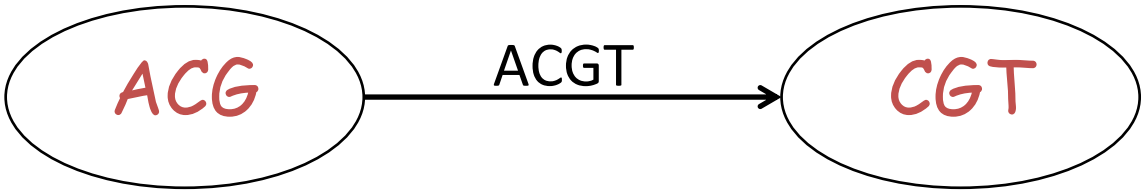
- When traversing an edge – mark both the edge and its RC edge.

- Running time: $O(|\sum|^k)$

The problem with even k

- The alg works on graphs that satisfy:
  1. The graph is strongly connected.
  2. Each vertex is balanced.
  3. ∃ a pairing of the edges in RC pairs.

- Alg fails for even k due to palindromes!

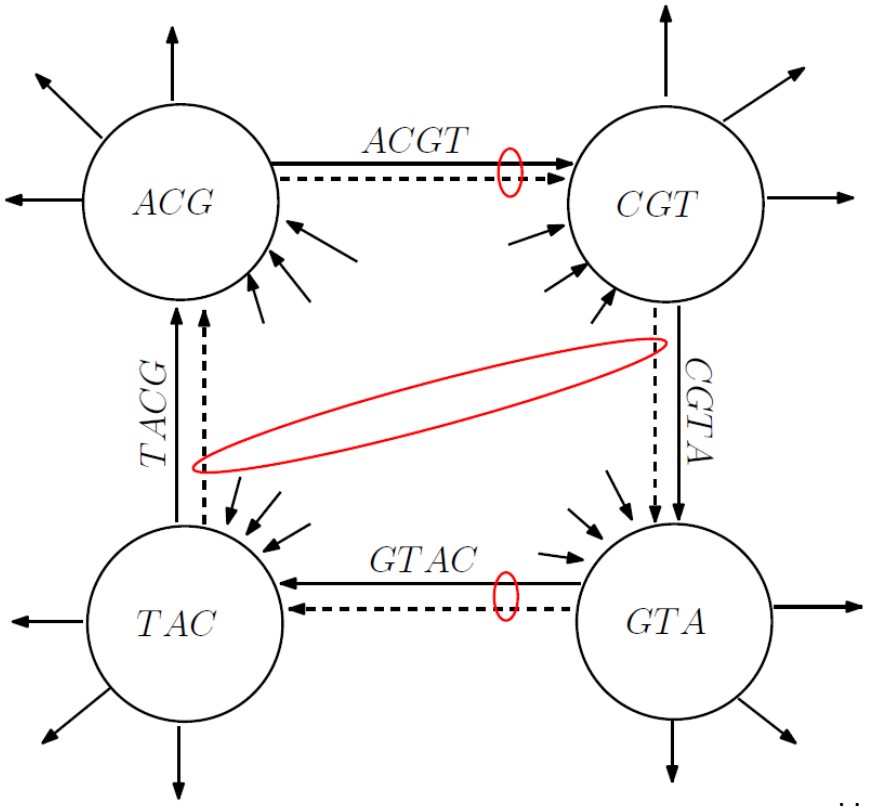


The solution: adding cycles

∀ pair of palindrome edges that are cyclic shifts of each other, add edges for all their cyclic shifts.

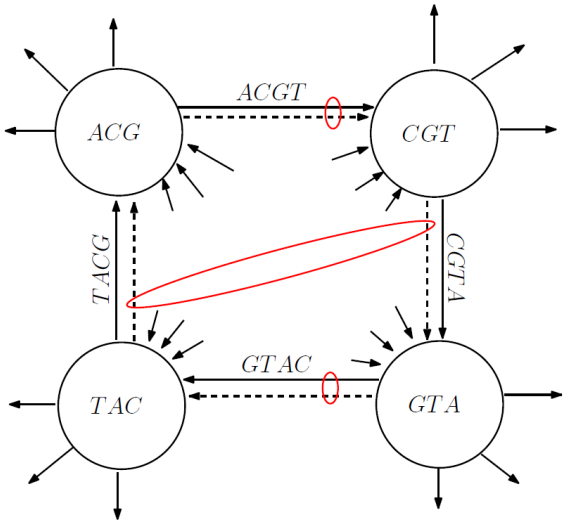
added edges.

RC matched pairs.



The augmented de Bruijn graph

- The addition of cycles preserves connectivity and vertex balance.
- Is the pairing preserved?
  - The added palindromes match the original palindromes in the graph.
  - The non-palindromic edges match each other.



- Alg: Augment the graph, form RC Euler tour
- Linear time, suboptimal seq length
- Developed netflow alg for opt seq length

Computational results

Lemma: length of RCdb seq of order k:

$$n^*(k) \geq \begin{cases} \frac{|\Sigma|^k}{2} & \text{if } k \text{ is odd} \\ \frac{|\Sigma|^k + |\Sigma|^{k/2}}{2} & \text{if } k \text{ is even} \end{cases}$$

Lengths (for even k)

K	2	4	6	8	10	12	14
Original	16	256	4,096	65,536	16,777,216	16,777,216	268,435,456
Lower bound	10	136	2,080	32,896	524,800	8,390,656	134,225,920
Linear algorithm	10	142	2,140	33,262	526,840	8,400,808	134,275,060
Optimal algorithm	10	142	2,140	33,262	526,816	8,400,772	134,274,844
Saving factor	1.6	1.8	1.91	1.97	1.99	1.997	1.999

Challenge #2: cover all k-mers in unstructured RNA probes

- Input:
  - k – k-mers to cover.
  - l – length of probe.
  - p – multiplicity of k-mers.

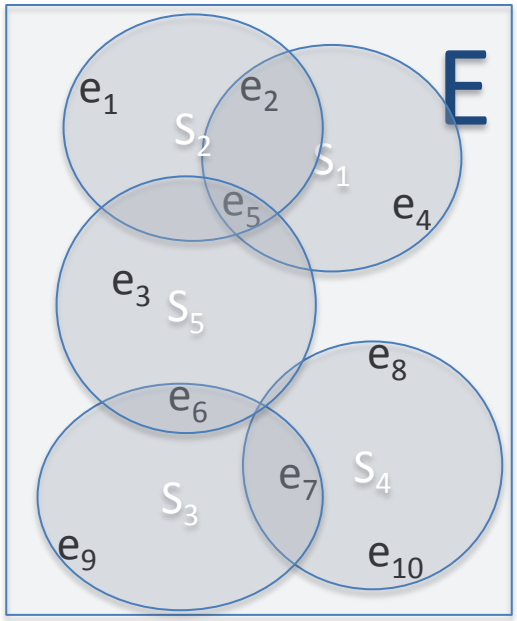
- p-multi k-mer coverage: each k-mer appears p times.

- **Goal**: minimum p-multi k-mer coverage by a restricted set of l-long sequences.

(Orenstein and Berger, *JCB* 2015)

K-mer coverage is NP-hard

- Easy when: all l-long sequences are allowed or l=k.
- Reduction from **minimum m-set cover**: find smallest subset S' of m-sets S that covers all elements in E.



Minimum 3-set cover example:
$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}\}$
$S = \{S_1, S_2, S_3, S_4, S_5\}$

Analogously:

- Each k-mer is an element.
- Each sequence is a set.

ACGU… → CGU[ACGU]

Reduction overview

1. Map elements to k-long {A,U}-representations.

$k = \lceil log_2 |E| \rceil$ $\quad e_{10} \rightarrow f_{01}(e_{10}) = 0001010 \rightarrow f_{AU}(e_{10}) = AAUAUA$

2. Convert each set to an l-long sequence (l=3km).
   - Pad each element w by G^k-w-C^k.

$$\{e_1,\ldots, e_m\} \rightarrow G^k f_{AU}(e_1) C^k \ldots G^k f_{AU}(e_m) C^k$$

3. Find k-mer coverage over {A,C,G,U}.

Reduction time: $O((|E|^2+|S|) \cdot m \cdot \log|E|)$

Approximation algorithm

(H_{l-k+1}-½)-approximation to k-mer coverage.

$$H_n = \sum_{i=1}^{n} {}^{1}/_{i} \leq \ln(n) + 1$$ (Levin, SIAM J. Discrete Math 2008)

Algorithm 1:

1. Find all l-long unstructured RNA sequences.
2. Apply the greedy set cover algorithm:

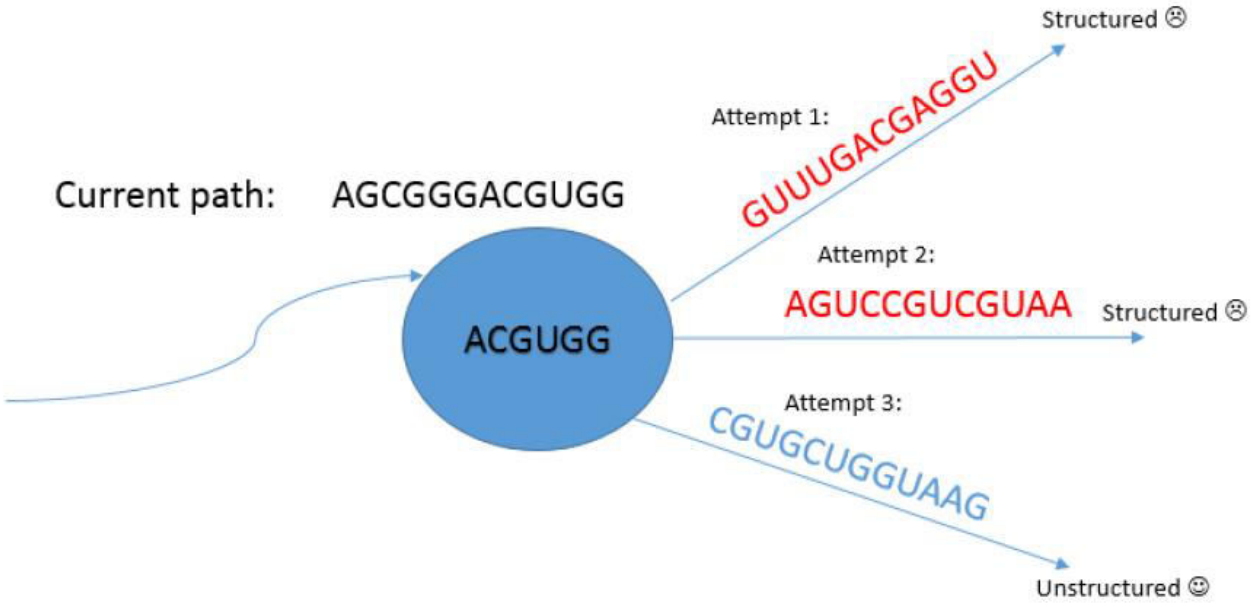
   Elements = all k-mers Sets = unstructured sequences

Running time: $\Omega(4^l \bullet l)$ ➔ impractical.

Heuristic algorithm

Key points:

1. Random walks in de Bruijn graph to cover all edges.
2. Backtracking in case no unstructured oligo is found.



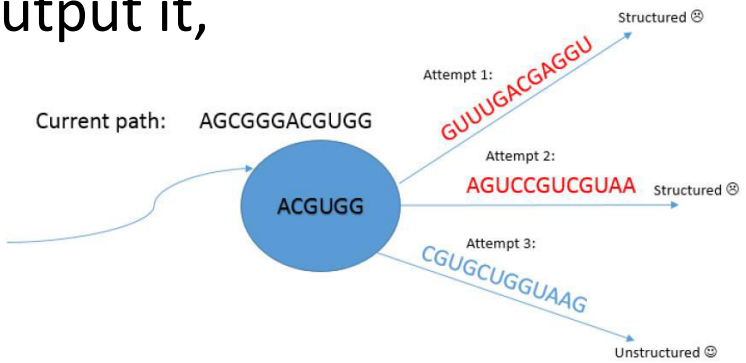
Heuristic algorithm

Algorithm 2 (k, l, p)

1. de Bruijn graph, order k-1, p copies of each edge.
2. L = l, V = arbitrary vertex.
3. While (edges exist):
   a. Find unstructured L-long path. Output it, L = l.
   b. If did not find after 100 attempts, L = L-1.
   c. If (L = k-1), output a random edge from V, L = l.
   d. If closed an unstructured cycle, output it, L = l, set V to a visited vertex.

Run time: O(#probes • f(l) • l)

#probes = $\Theta(4^k / (l-k+1))$, $f(l)=O(l^2)$

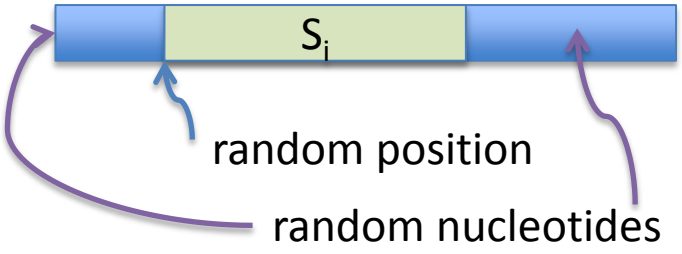


Extension algorithm

- Not all sequences are of length l (cycles, structured).
- Extend them to unstructured l-long sequences.

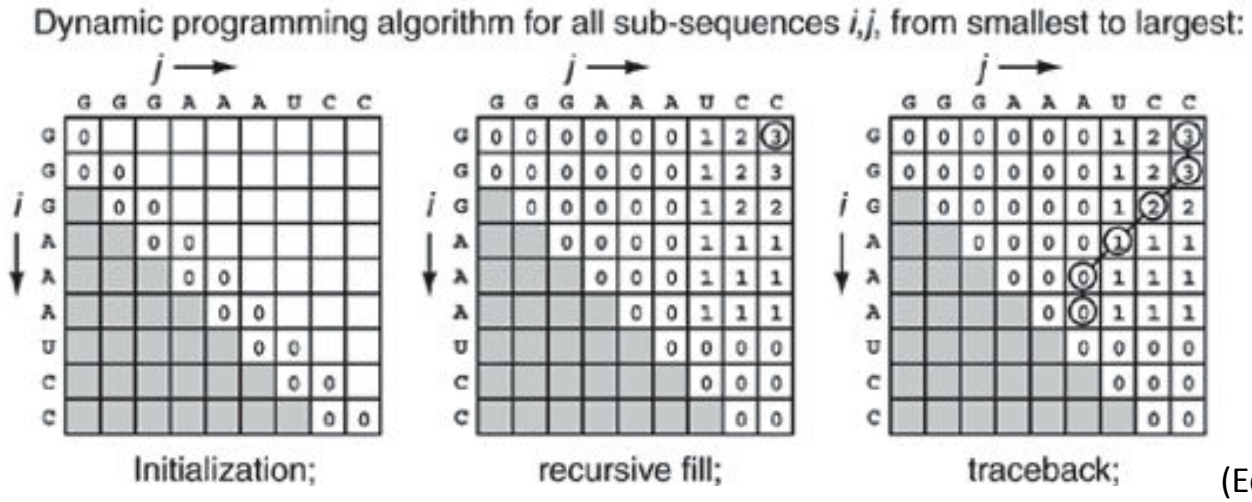
Algorithm 3 (S, k, l):

1. Try at most 100 random extensions to unstructured.
2. If succeeded, output complete sequence.
3. If failed, divide to two overlapping halves: s_1, s_2.
4. If $|s_i|=k$, output s_i.
5. Continue recursively on s_i.



Implementation details

1. Limit number of random attempts (parameter).
2. Extend by doubling probe length.
3. Preform RNA secondary structure predictions based on previous predictions (not implemented).



(Eddy, Nature Biotechnology 2014)

Results comparison

1. Theoretical lower bound
   - Derived from k-mer counts

$$n(k,l) \geq \left\lceil \frac{4^k \cdot p}{l-k+1} \right\rceil$$

2. Naïve algorithm:
   - Generate random oligos.
   - Add those which are unstructured and cover uncovered k-mers.

Results for different (k,l)

ℓ	k	Lower bound	Incomplete set	Incomplete Ratio	Complete set	Complete ratio	Structured	Naive set	Runtime (hh:mm:ss)
30	5	40	50	1.25	51	1.27	0	149	00:02:11
	6	164	182	1.11	182	1.11	0	766	00:07:43
	7	684	737	1.08	739	1.08	0	3 308	00:41:40
	8	2 850	3 081	1.08	3 106	1.09	0	13 801	02:58:52
	9	11 916	12 940	1.09	13 069	1.10	59	57 154	14:42:27
	10	49 934	55 882	1.12	56 526	1.13	670	236 477	82:18:01
35	5	34	41	1.21	41	1.21	0	131	00:03:13
	6	138	158	1.14	162	1.17	0	670	00:21:20
	7	566	635	1.12	648	1.15	0	2 884	01:17:43
	8	2 342	2 670	1.14	2 744	1.17	0	11 961	06:03:05
	9	9 710	11 022	1.14	11 439	1.18	60	49 289	26:47:31
	10	40 330	47 139	1.17	49 225	1.22	609	202 763	137:33:27
40	5	30	37	1.23	38	1.27	0	117	00:02:44
	6	118	140	1.19	148	1.25	0	598	00:36:31
	7	482	561	1.16	611	1.27	0	2 561	02:33:16
	8	1 986	2 362	1.19	2 627	1.32	0	10 597	11:24:15
	9	8 192	9 745	1.19	10 966	1.34	60	43 492	48:02:15
	10	33 826	41 798	1.24	47 457	1.40	557	178 187	246:05:17

Self-structured k-mers – form structure with themselves.

Comparison to RNAcompete design

Parameters: k=9, l=35, p=16.

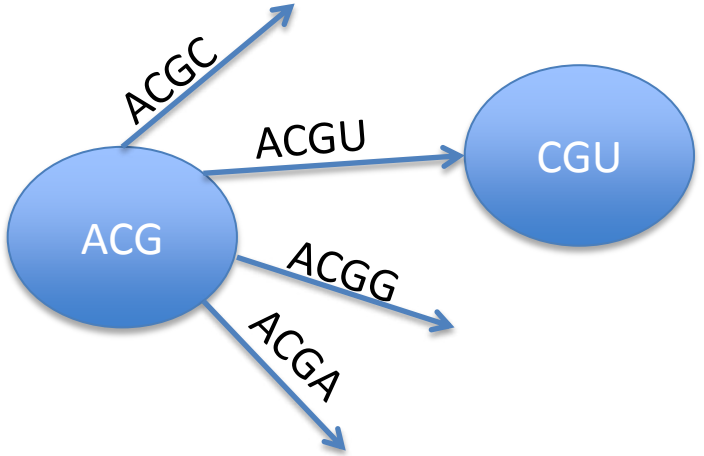
Design	Lower bound	#oligos	Ratio	#structured
Ours	155,346	166,649	1.07	841
RNAcompete		214,498	1.38	2,858

- Ours: all oligos 35-long.
- RNAcompete: varying lengths 35-38.

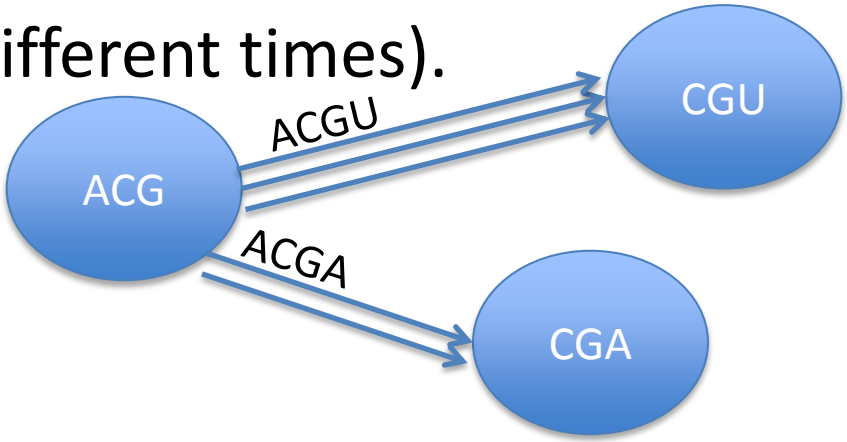
Future extensions

1. Generalize to any property of RNA/DNA probes.

2. Remove specific k-mers
   (by removing their edges).



3. Assign different k-mer multiplicities
   (by multiplying each edge different times).



Summary

- Utilized de Bruijn graph to generate DNA/RNA libraries that cover all k-mers.

- De Bruijn graphs more flexible than LFSRs.

de Bruijn sequences = $(4!)^{4^{k-1}}/4^k$

primitive polynomials = $\phi(4^{k-1})/k$

- General and flexible scheme for library design covering k-mers in specific sequences.

Acknowledgments

curlcake.csail.mit.edu

Berger lab

acgt.cs.tau.ac.il/shortcake

Shamir group

Postdocs available

NATIONAL INSTITUTES OF HEALTH

Edmond J. Safra Center for Bioinformatics
TEL AVIV UNIVERSITY
מרכז אדמונד י. ספרא לביואינפורמטיקה

DAN DAVID

ISRAEL SCIENCE FOUNDATION

i-CORE
Israeli Centers of Research Excellence
Israeli Centers Of Research Excellence