Follow-up about XPCR and Infogenomics

Dr Giuditta Franco

Department of Computer Science, University of Verona, Italy

DNA Extraction Problem

Given:

1. input pool P of heterogeneous DNA strands, same length n, same prefix α, and suffix β: $|\alpha x \beta| = n$;
2. a string γ (shorter than n);

Provide:

- output pool of *all and only* the γ-superstrands from P:

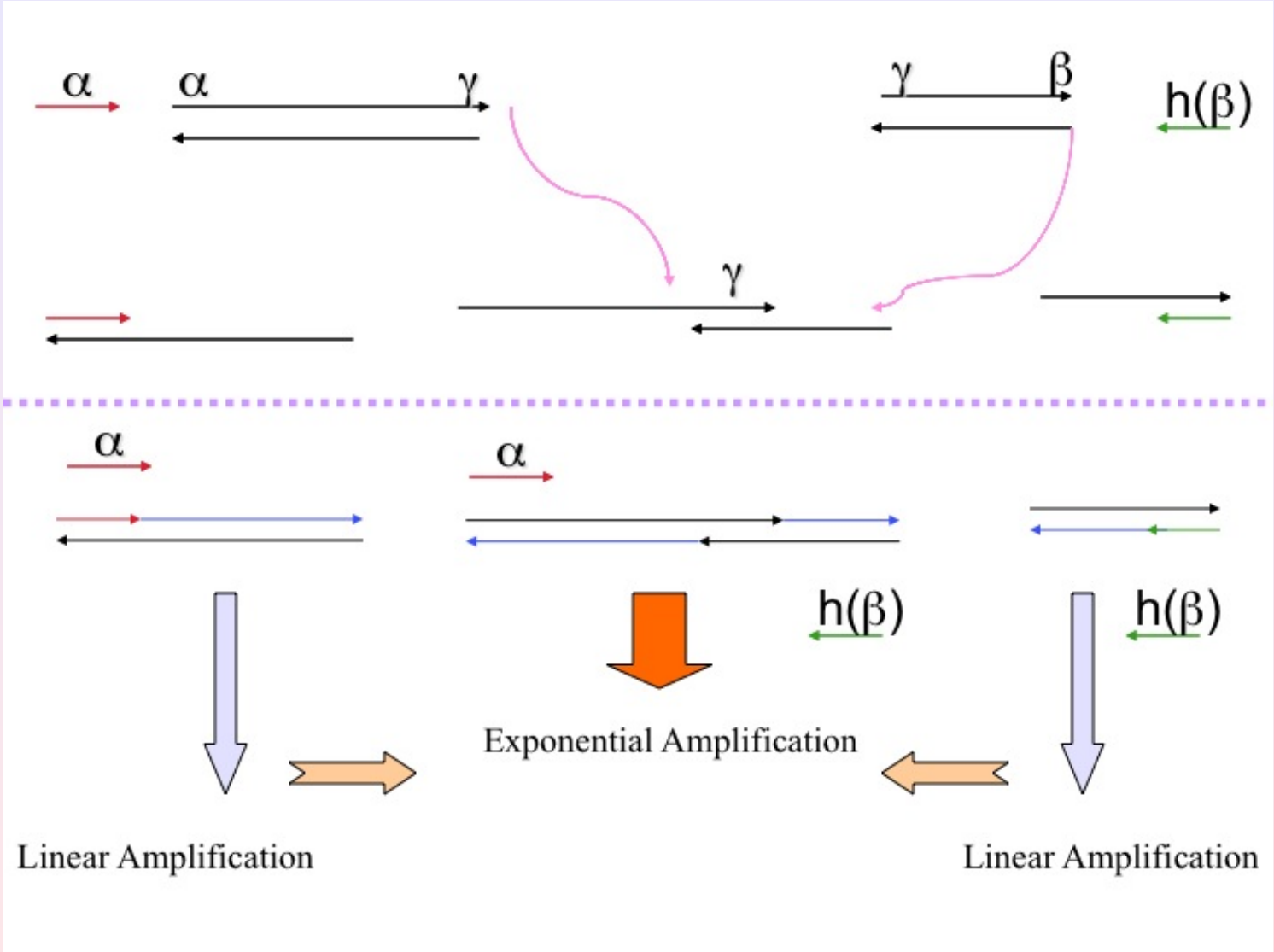
$$P_\gamma = \{\alpha y \gamma z \beta \mid \alpha y \gamma z \beta \in P, y, z \in \Sigma^*\}$$

One additional separation step

Given a string γ, we compute $XPCR_\gamma(\alpha, \overline{\beta})$.

Input pool: $P = \{\alpha x \beta \mid |\alpha x \beta| = n\}$.

1. $(P_1, P_2) := split(P)$;
2. $P_1 := PCR(\alpha, \overline{\gamma})(P_1)$;
3. $P_2 := PCR(\gamma, \overline{\beta})(P_2)$;
4. P:= $Mix(P_1, P_2)$;
5. P: = $\cup_{k<n} El_k(P)$; % what about skipping this step?
6. P: = $PCR(\alpha, \overline{\beta})(P)$; % combinatorial amplification
7. $P := El_n(P)$.



Slide courtesy of prof. Vincenzo Manca, Univ. Verona, IT

The main idea of XPCR Extraction

0. Given a Pool of strands of length n, beginning with α and ending with β
1. If a strand includes γ (of length L) then :
   - 1.2 copy its left part from γ to the beginning (backward), by PCR(α , h(γ))
   - 1.1 copy its right part from γ to the end (forward), by PCR(γ, h(β))
2. Select short strands with "conjugate" lengths L_1, L_2 (L_1+L_2 - L= n), by Gel Electrophoresis
3. Concatenate strands of the previous step, by XPCR(α, h(β)).
4. Keep only strands of length n.

Slide courtesy of prof. Vincenzo Manca, Univ. Verona, IT

XPCR based *Extract*(P, γ)

1. $P := infix(P, \alpha, \beta)$;
2. $L := length(P)$; $S := \emptyset$;
3. **for each** $n \in L$ **do**
    1. $R_1 := \emptyset, R_2 := \emptyset, Q := \emptyset, P_1 := \emptyset, P_2 := \emptyset$;
    2. $P := separate(P, n)$;
    3. $(P_1, P_2) := split(P)$;
    4. $P_1 := PCR(P_1, \alpha, \bar{\gamma})$;
    5. **for each** $m < n$ **do** $R_1 := mix(R_1, separate(P_1, m))$;
    6. $P_2 := PCR(P_2, \gamma, \bar{\beta})$
    7. **for each** $m < n$ **do** $R_2 := mix(R_2, separate(P_2, m))$;
    8. $Q := mix(R_1, R_2)$;
    9. $Q := PCR(Q, \alpha, \bar{\beta})$;
    10. $Q := separate(Q, n)$;
    11. $S := mix(S, Q)$;
    12. **output** S [1].

[1] Problem of the γ-chimeras/mermaids.

EXPCR = DNA Extraction by XPCR

Experimental Check

Consider a pool P of $\alpha \ldots \beta$-strands that are either γ-superstrands or γ'-superstrands ($\gamma \neq \gamma'$),

where all γ-superstrands are either $\gamma 1$-superstrands, or $\gamma 2$-superstrands, or $\gamma 3$-superstrands … ($\gamma 1 \neq \gamma 2 \neq \gamma 3 \ldots$)

Slide courtesy of prof. Vincenzo Manca, Univ. Verona, IT

Experimental Check

Our extraction is correct and complete in the sense that:

1. XPCR-Extraction selected only γ-superstrands
2. XPCR-Extraction selected all kinds of γ-superstrands ($\gamma 1$, $\gamma 2$, $\gamma 3$ …- superstrands).

Slide courtesy of prof. Vincenzo Manca, Univ. Verona, IT

Chimeras/Mermaids

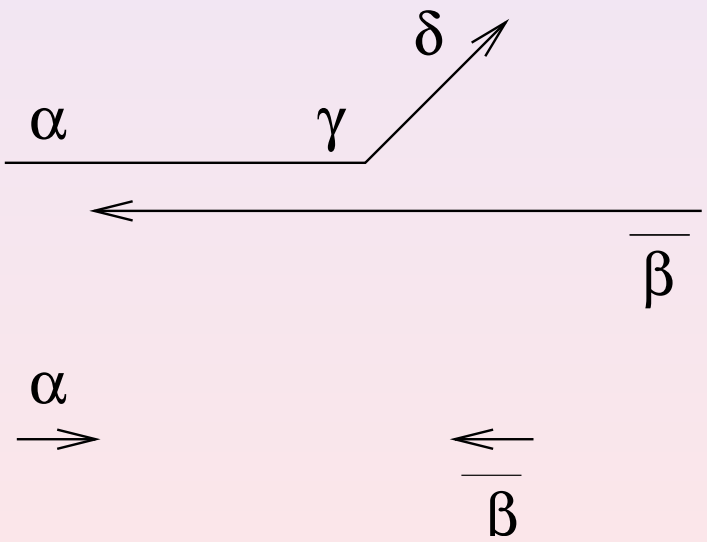
$XPCR - PureExtract(P, \gamma)$

1. $L := length(P)$;
2. **for each** $n \in L$ **do**
3. $P := separate(P, n)$;
4. $(P, Q) := split(P)$;
5. $P := infix(P, \alpha, \lambda)$;
6. $Q := infix(Q, \lambda, \beta)$;
7. $P := PCR(P, \alpha, \overline{\gamma\delta})$;
8. $P := mix(P, Q)$;
9. $P := PCR(P, \alpha, \bar{\beta})$;
10. $P := separate(P, n + |\alpha| + |\beta|)$;

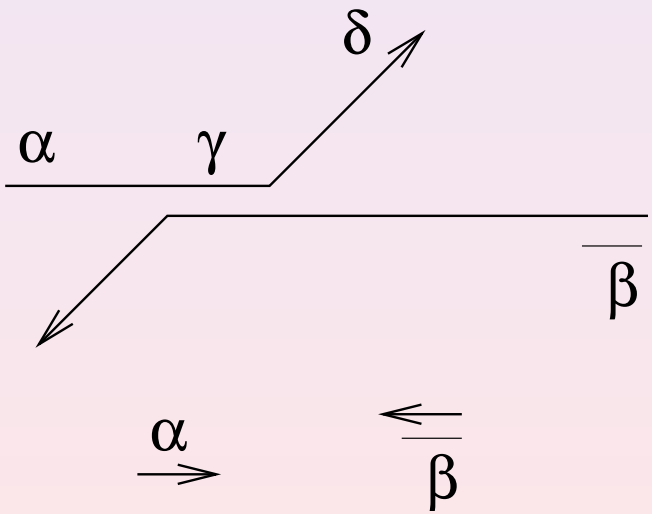
output P

Extraction with no chimeras

amplification



no amplification



$XPCR - Mutagenesis(P, \gamma, \delta)$

let $P = \{< \alpha\gamma\beta >\}$, and $Q = \{< \alpha[-18,-1]\ \delta\ \beta[1,20] >\}$;

1. $(P_1, P_2) := split(P)$;
2. $P_1 := PCR(P_1, \alpha[1,18], mir(\alpha[-18,-1]))$;
3. $P_2 := PCR(P_2, \beta[1,20], mir(\beta[-20,-1]))$;
4. $P_1 := separate(P_1, |\alpha|)$; $P_2 := separate(P_2, |\beta|)$;
5. $P_1 := mix(P_1, Q)$;
6. $P_1 := PCR(P_1, \alpha[1,18], mir(\beta[1,20]))$;
7. $P_1 := separate(P_1, |\alpha| + |\delta| + 20)$;
8. $P := mix(P_1, P_2)$;
9. $P := PCR(P, \alpha[1,18], mir(\beta[-20,-1]))$;
10. $P := separate(P, |\alpha| + |\beta| + |\delta|)$;

Output: P.

Quaternary Recombination Algorithm

This method starts from α-prefixed and β-suffixed I_1, I_2, I_3 I_4, works in linear time, by using polymerase extension.

Let P_1 and P_2 be two copies of the pool

$$\{\alpha I_1 \beta,\ \alpha I_2 \beta,\ \alpha I_3 \beta,\ \alpha I_4 \beta\}$$

for $i = 2, 3, 4, 5$ **do**

- **perform** $XPCR_{X_i}$ on P_1 and $XPCR_{Y_i}$ on P_2 [2]
- **mix** the two pools into one $P := P_1 \cup P_2$, then **split** P randomly in two new pools P_1 and P_2

[2] Run together (no intermediate electrophoresis) have a worse efficiency and complexity

Splicing Examples

Initial sequences: $I_1 = X_1X_2X_3X_4X_5X_6$, $I_2 = Y_1Y_2Y_3Y_4Y_5Y_6$, $I_3 = X_1Y_2X_3Y_4X_5Y_6$, $I_4 = Y_1X_2Y_3X_4Y_5X_6$.

1. $I_1,\ I_4 \xrightarrow{r_{X_2}} X_1X_2Y_3X_4Y_5X_6,\ Y_1X_2X_3X_4X_5X_6,$
   $I_2,\ X_1X_2Y_3X_4Y_5X_6 \xrightarrow{r_{Y_5}} Y_1Y_2Y_3Y_4Y_5X_6,\ \mathbf{X_1X_2Y_3X_4Y_5Y_6}.$
2. $I_2,\ I_4 \xrightarrow{r_{Y_3}} Y_1Y_2Y_3X_4Y_5X_6,\ Y_1X_2Y_3Y_4Y_5Y_6,$
   $I_1,\ Y_1Y_2Y_3X_4Y_5X_6 \xrightarrow{r_{X_4}} X_1X_2X_3X_4Y_5X_6,\ \mathbf{Y_1Y_2Y_3X_4X_5X_6}.$
3. $I_1,\ I_3 \xrightarrow{r_{X_5}} X_1Y_2X_3Y_4X_5X_6, X_1X_2X_3X_4X_5Y_6,$
   $I_2,\ X_1Y_2X_3Y_4X_5X_6 \xrightarrow{r_{Y_2}} X_1Y_2Y_3Y_4Y_5Y_6,\ \mathbf{Y_1Y_2X_3Y_4X_5X_6}$

Correctness/completeness of recomb. algorithm

The n-dimensional library $\{\alpha_1 \cdots \alpha_n \mid \alpha_i \in \{X_i, Y_i\}, i = 1, \ldots, n\}$ is the null context splicing language generated by the system $\mathcal{N} = (\Sigma, A, R)$, where $\Sigma = \{A, T, C, G\}$, $A = \{I_1, I_2, I_3, I_4\}$, and $R = \{r_{X_2}, r_{Y_2}, \ldots, r_{X_{n-1}}, r_{Y_{n-1}}\}$.

The recombination algorithm, by construction, generates the null context splicing language. Any null context splicing rule (over axioms and products) generates an element of the library. We have to show that it generates the whole n-dimensional library, that is, **each element of the library is generated by a sequence of null context splicing rules**.

Algorithm Correctness Proof (1/2)

Proof. For any recombination $\alpha_1\alpha_2\ldots\alpha_n$ there exists the subset of rules $\{r_{\alpha_2}, r_{\alpha_3}, \ldots, r_{\alpha_{n-1}}\}$ that generates it by means of the following computation starting from the initial sequences.

Let us call L_i the initial sequence containing $\alpha_{i-1}\alpha_i$ as subsequence, for $i = 2, \ldots, n$, and let c, s_1, s_2 be string variables.

By construction, for each value of i there exists only one of such an initial sequence.

Algorithm Correctness Proof (2/2)

$c := L_2$

for $j = 2, \ldots, n-1$
apply $r_{\alpha_j} : c,\ L_{j+1} \longrightarrow s_1,\ s_2$;
$c := s_1$;

output: c

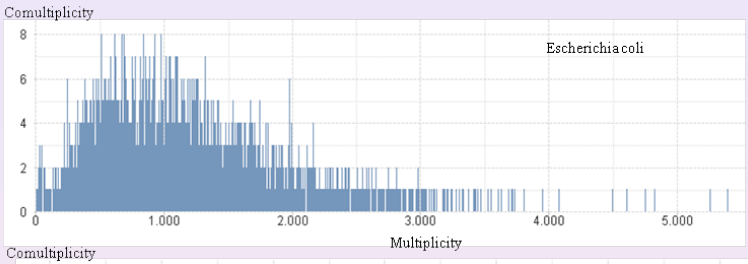
The string c contains the substrand $\alpha_1\alpha_2$ before the for cycle, and $\alpha_1\alpha_2 \ldots \alpha_i$ after the cycle corresponding to $j = i - 1$. Since its length remains constant during the computation, after the for cycle the string c is equal to $\alpha_1\alpha_2 \ldots \alpha_n$. □

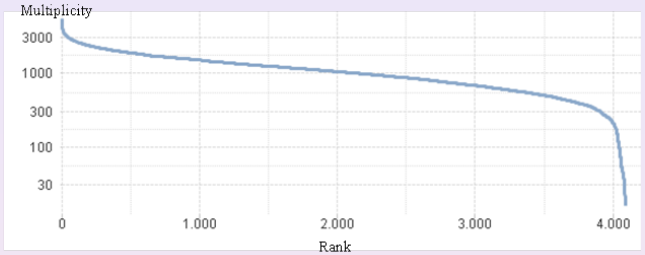
Infogenomics introduced a method

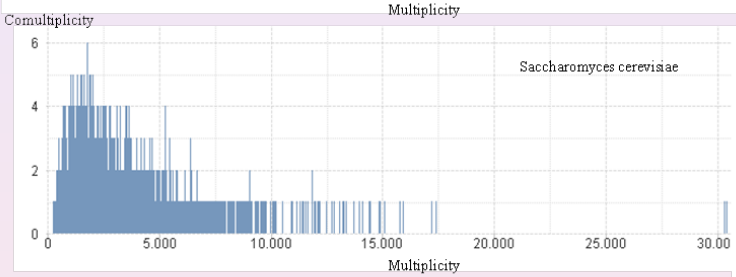
- to represent and compare genomes (genomic profiles, dictionary intersections): Zipf diagrams
- ...
- to (automatically) find tandem repeats, and good genomic dictionaries.

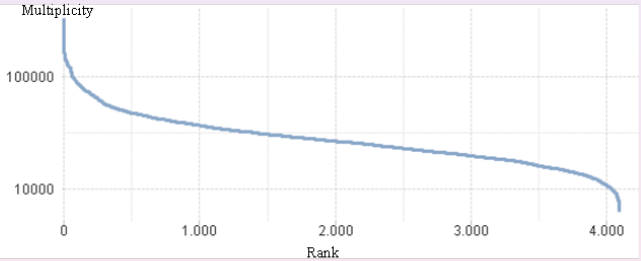
Genomic profiles and Zipf curves

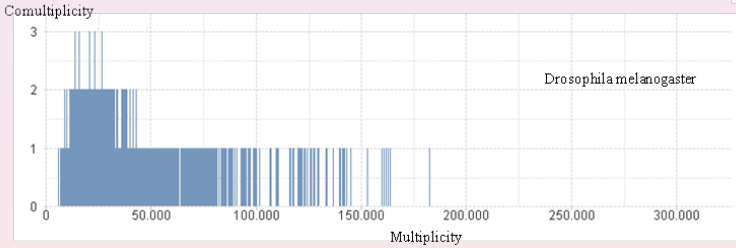
Zipf curves measure word frequencies in natural languages

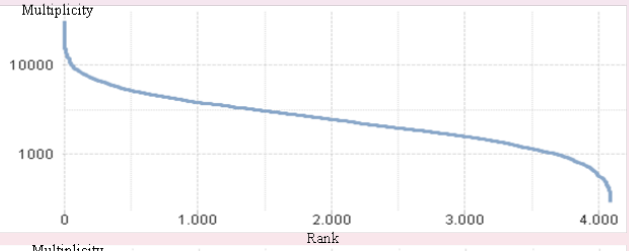


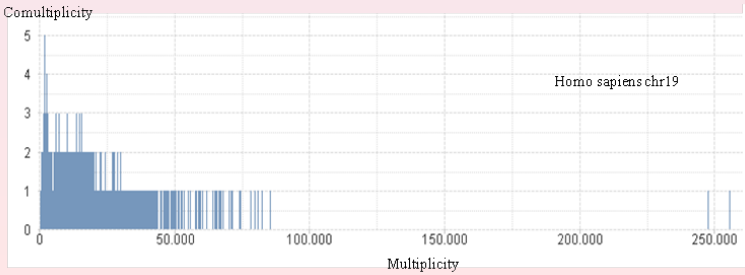


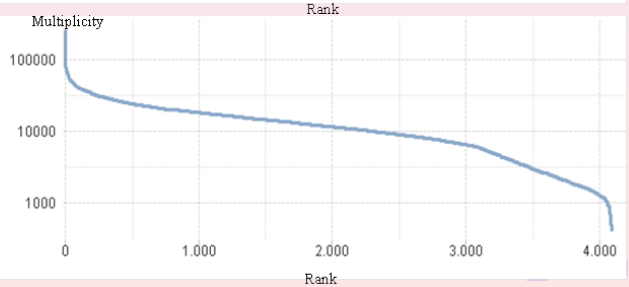












Genomic Dictionaries

- D(G) = {G[i,j] |1 ≤ i ≤ j ≤ |G|} (square dim. w.r.t. |G|)
- $D_k(G) = D(G) \cap \Gamma^k$
- L included in D(G) is a dictionary of G
- A position p of G is m-covered in D if there are m words G[i,j] of D with i ≤ p ≤ j **(positional coverage)**
- D covers G if every position of G is k-covered with k ≥ 1 by D **(lexical coverage)**
- D minimally covers G if D covers G and no D' included in D covers G
- G is D-segmentable if G belongs to D*

Slide courtesy of prof. Vincenzo Manca, Univ. Verona, IT

A Word Selection Algorithm based on EXP/KL

- Word Recurrence distance distribution RDD(α)
- RDD(α) ➔RDD*(α) (norm. removing peaks and holes)
- Best Exponential Distr. approx. to RDD*(α), NED(α)
- Entropic “distance” between distributions RDD and NED (symmetric Kullback-Leibler) KL~(RDD||NED)
- Extraction of words by elongation stability
  (starting from different seeds < 10)
- Union of the maximal elongations
- Word filtering by different tests

(length, multiplicity, sequence coverage, positional coverage, …)

Slide courtesy of prof. Vincenzo Manca, Univ. Verona, IT

Extraction of Genomic Dictionaries
Vincenzo Bonnici, University of Verona, Italy

A novel word recurrence based approach

- **Approach**
  - Characterize words by the **divergence** between their RDD and a theoretical distribution
  - The divergence is used as a measure of the information content of a word
  - **Elongate** low expressive words until they acquire a reasonable level of significance

- **Random deviation** of a word α
  1) **Extract** the RDD of α in G, R_α
  2) Remove distribution noise (peaks)
  3) Force $\mathbf{R_\alpha}$ to be a probability distribution
  4) **Estimate** an exponential distribution $\mathbf{E_\alpha}$ from R_α
  5) Force E_α to be a probability distribution
  6) Calculate the **random deviation** as

$$\mathbf{r(\alpha)} = \max(KL(R_\alpha,E_\alpha), KL(E_\alpha,R_\alpha))$$

  where **KL** is the **entropic divergence** (namely the Kullback-Leibler divergence)

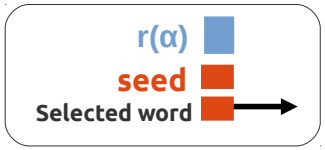
Slide courtesy of Dr. Vincenzo Bonnici, Univ. Verona, IT

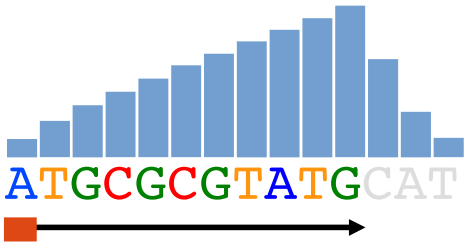
Extraction of Genomic Dictionaries

Vincenzo Bonnici, University of Verona, Italy

Elongation procedure

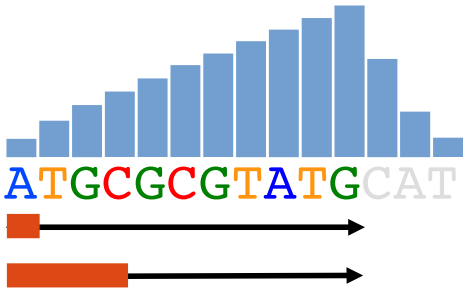
Words, factors and roots





Basic elongation strategy

Elongate a seed until r(α) increases



Elongation strategy is inversely inclusive w.r.t. seeds

Elongation from large seeds include what smaller seed elongate

Slide courtesy of Dr. Vincenzo Bonnici, Univ. Verona, IT

Extraction of Genomic Dictionaries

Vincenzo Bonnici, University of Verona, Italy

Elongation procedure

Preferred word lengths in WLD

Hg19, chromosome 1

extracted word length \ seed length	1	2	3	4	5	6	7	8
4		5	5					
5	17	54	108	179				
6	41	305	666	1,306	1,666			
7	**92**	**337**	**616**	**1,478**	**2,310**	**2,925**		
8	79	178	280	468	593	1,474	4,151	
9	43	142	248	562	811	3,879	14,614	39,347
10	8	**221**	**542**	**1,325**	**2,140**	**9,106**	48,112	144,355
11	**13**	197	479	1,284	2,115	6,986	**50,442**	**224,644**
12		122	297	838	1,363	2,201	24,687	303,163
13	2	53	119	327	579	774	6,403	136,135
14	2	19	36	80	145	194	1,094	20,805
15	2	7	9	21	33	50	291	4,193
16		5	7	12	17	24	99	1,196
17		2	3	5	6	9	27	327
18		1	1	1	1	4	12	128
19							2	43
20							2	15
21								6
23								1
24								1

Slide courtesy of Dr. Vincenzo Bonnici, Univ. Verona, IT

Extraction of Genomic Dictionaries

Vincenzo Bonnici, University of Verona, Italy

Elongation procedure

Preferred word lengths and their sequence coverage

Hg19, chromosome 1
Sequence coverage

extracted word length \ seed length	1	2	3	4	5	6	7	8
4		0.0291	0.0291					
5	0.0309	0.0790	0.1362	0.1681				
6	0.0269	**0.3149**	**0.5504**	**0.7767**	**0.8426**			
7	**0.0742**	0.2479	0.3878	0.6430	0.7691	**0.8141**		
8	0.0285	0.0616	0.0899	0.1187	0.1384	0.1643	**0.2634**	
9	0.0115	0.0209	0.0303	0.0499	0.0615	0.0714	0.1593	**0.6315**
10	0.0008	0.0054	0.0071	0.0128	0.0206	0.0329	0.0974	0.5388
11	**0.0025**	**0.0077**	**0.0088**	**0.0108**	**0.0127**	0.0174	0.0602	0.3509
12		0.0028	0.0031	0.0081	0.0089	0.0101	0.0342	0.2858
13	0.0000	0.0006	0.0013	0.0054	0.0065	0.0070	0.0155	0.1209
14	0.0035	0.0048	0.0049	0.0056	0.0065	0.0066	0.0101	0.0451
15	0.0026	0.0036	0.0036	0.0050	0.0052	0.0052	0.0065	0.0214
16		0.0016	0.0017	0.0017	0.0017	0.0028	0.0032	0.0090
17		0.0011	0.0011	0.0012	0.0013	0.0013	0.0014	0.0031
18		0.0006	0.0006	0.0006	0.0006	0.0012	0.0012	0.0020
19							0.0000	0.0003
20							0.0000	0.0002
21								0.0001
23								0.0000
24								0.0000

Slide courtesy of Dr. Vincenzo Bonnici, Univ. Verona, IT

Extraction of Genomic Dictionaries

Vincenzo Bonnici, University of Verona, Italy

Elongation procedure

Preferred word lengths and their positional coverage

Hg19, chromosome 1
Positional coverage

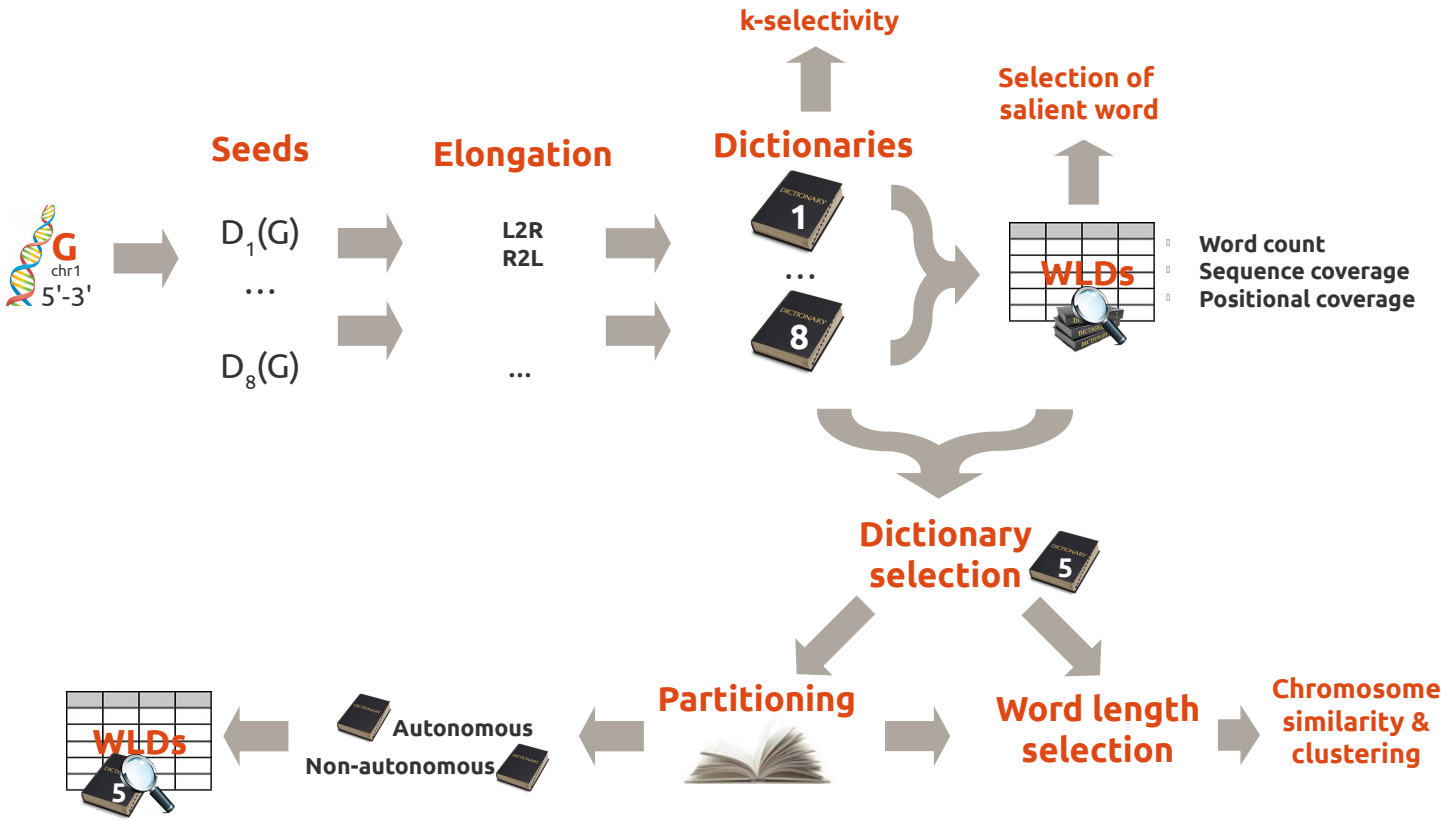
extracted word length \ seed length	1	2	3	4	5	6	7	8
4		1.0078	1.0078					
5	1.0807	1.1690	1.2411	1.4198				
6	**1.1539**	**1.3022**	**1.6590**	**2.3201**	**2.7715**			
7	1.0934	1.2876	1.4587	1.9817	2.5877	2.9160		
8	1.1569	1.2590	1.3125	1.4228	1.5184	1.5836	1.5572	
9	**1.4480**	**1.5411**	**1.5211**	**1.7039**	**1.8791**	**1.8661**	1.5470	1.7484
10	1.0006	1.1090	1.1033	1.1697	1.1926	1.2632	1.2580	1.5457
11	**4.0810**	**2.1729**	**2.0809**	**1.9100**	**1.7829**	**1.6131**	**1.3009**	1.3658
12		1.0654	1.0624	1.1926	1.1809	1.1716	1.1507	1.3455
13	1.0000	1.0000	1.0000	1.1355	1.3769	1.3530	1.2340	1.3709
14	1.0000	1.0000	1.0000	1.0551	1.2244	1.2235	1.1687	1.3807
15	1.0000	1.1446	1.1445	1.1065	1.1739	1.1725	1.1444	1.2559
16		1.2684	1.2636	1.2588	1.2539	1.1544	1.1447	1.1148
17		1.0000	1.0000	**1.3982**	**1.3957**	**1.3948**	**1.3608**	**1.3440**
18		1.0000	1.0000	1.0000	1.0000	1.0000	1.0015	1.0187
19							1.0000	1.0000
20							1.0000	1.0000
21								1.0000
23								1.0000
24								1.0000

Slide courtesy of Dr. Vincenzo Bonnici, Univ. Verona, IT

Extraction of Genomic Dictionaries

Vincenzo Bonnici, University of Verona, Italy

Informational Analysis Pipeline



Slide courtesy of Dr. Vincenzo Bonnici, Univ. Verona, IT

Dictionary Validation

Words extracted by informational methods are informationally relevant, but what about their biological meaning?
(Infogenomics is analogous to ENCODE)

Words are pieces on which genomes were built.
Which categories emerge?

Words are, in this perspective, ***iper-dense information units***

How defining and discovering biological significance?
Can information tell us deep biological mechanisms?

****

Slide courtesy of prof. Vincenzo Manca, Univ. Verona, IT