INFERRING MOBILE ELEMENTS IN *S. CEREVISIAE* STRAINS

Giulia Menconi[1], Giovanni Battaglia[2], Roberto Grossi[2], Nadia Pisanti[2] and Roberto Marangoni[2,3]
[1]*Istituto Nazionale di Alta Matematica, Città Universitaria, 00185 Roma, Italia*
[2]*Dipartimento di Informatica, Università di Pisa, 56127 Pisa, Italia*
[3]*CNR-Istituto di Biofisica, 56124 Pisa, Italia*

Keywords: Residentome, Retrotransposons, Mobile DNA elements, L-grams.

Abstract: We aim at finding all the mobile elements in a genome and understanding their dynamic behavior. Comparative genomics of closely related organisms can provide the data for this kind of investigation. The comparison task requires a huge amount of computational resources, which in our approach we alleviate by exploiting the high similarity between homologous chromosomes of different strains of the same species. Our case study is for `RefSeq` and two other strains of *S. cerevisiæ*. Our fast algorithm, called REGENDER, is driven by data analysis. We found that almost all the chromosomes are composed by resident genome (more than 90% is conserved). Most importantly, the inspection of the non-conserved regions revealed that these are putative mobile elements, thus confirming that our method is useful to quickly find mobile elements. The software tool REGENDER is available online at http://www.di.unipi.it/∼gbattag/regender.

1 INTRODUCTION

Mobile elements play a prominent role in eukaryotic genomes. They represent around 15–20% of the Human genome (Lewin, 2007). They are very vivacious, since they are able to jump over the genome, to duplicate in different positions, and to possibly express their own genes. They can also change cells' phenotype, when jumping within a gene sequence (or its regulatory elements), thus altering its expression. Some complex pathologies whose molecular mechanisms and global inheritance are hard to explain by standard inheritance laws, have found to be correlated to mobile elements translocations (see e.g. (Conti et al., 2006)). This scenario suggests to look at eukaryotic genomes as evolving ecosystems, where the *resident genome* (intended as the immotile DNA) and the different elements of the *mobilome* (the total of mobile elements, mainly transposons) act like different species competing for the available biochemical resources (Le Rouzic et al., 2007; Venner et al., 2009). The importance of mobile elements strongly supports the development of tools for quickly mapping all of them in a genome. The "classic" strategy to face this problem consists in taking the known mobile element sequences and finding their occurrences in the genome. However, we cannot apply this strategy when we do not know *a priori* all the different kinds of mobile elements. Moreover, there are further problems. Mobile elements are subject to mutation, fragmentation, and fusion of consecutive elements in certain cases: hence, many false negative outcomes are possible. Also, unresolved sequences are a primary source of difficulty.

A more promising strategy is to compare genomes of closely related organisms. The rationale is that most of the chromosomal rearrangements observed in such datasets are probably caused by mobile elements.

Recently, a dataset suitable for this purpose has been made available: 39 different strains of *S. cerevisiæ* have been sequenced, and the relative genomes published without annotations (Liti et al., 2009). The coverage of the dataset is relatively low (one-to-fourfold), and the genomic sequences contain a fraction of unresolved sequences. Unfortunately, performing a genome-wide alignment is computationally demanding.

Our proposed approach exploits the high similarity between homologous sequences of different strains of the same species, so as to perform a simple, but powerful *ad hoc* alignment. Using *L*-grams we detect the non-conserved regions by identifying and masking the long conserved ones. In other words, we identify the resident genome and then we analyze its complementary part to infer elements of the

Menconi G., Battaglia G., Grossi R., Pisanti N. and Marangoni R..
INFERRING MOBILE ELEMENTS IN S. CEREVISIAE STRAINS.
DOI: 10.5220/0003137001310136
In *Proceedings of the International Conference on Bioinformatics Models, Methods and Algorithms* (BIOINFORMATICS-2011), pages 131-136
ISBN: 978-989-8425-36-2

mobilome. Our method is very efficient, since the resident genome localization requires just few seconds for chromosomal sequences of millions of bases. What remains for non-conserved regions is very small when compared to the length of the chromosomes at hand. Consequently, more sophisticated and expensive methods can be focused to this restricted set of candidates.

Our method builds a map for the mobilome of *S. cerevisiæ* as a case study. For any given chromosome in our dataset, we classify its segments as either *conserved* or *non-conserved* regions, using `RefSeq@SGD` (SGD, 2010) as the reference genome for this yeast (since `RefSeq` is the only annotated strain). We then inspect and mark all the non-conserved regions trying to infer putative mobile elements. Since *S. cerevisiæ* is probably the organism where mobile elements are best characterized, it is an optimal benchmark for validating our approach.

We apply the following classification by referring to the available annotations of `RefSeq`: (i) completed transposons, including their bounding segments called *Long Terminal Repeats* (LTRs), in this case indicated as Ty; (ii) pairs of LTRs without the transposon in between; (iii) soloLTRs, where a single LTR element is found in each of them. Then, the map annotates all these occurrences of putative mobile elements. Interestingly, the map can be seen as a puzzle with some missing pieces (the non-conserved regions). A global view at its configuration allows to locate a single candidate position for several non-conserved regions with high confidence, even if their relocation involves unresolved symbols marked by `Ns`. Note that we deploy the peculiar structure of the transposons that can be evinced from `RefSeq`: they are long between 5000b–6000b (bases) and are delimited by two LTRs of 200b–300b. We illustrate our approach by involving the *S. cerevisiæ* strains `Y55` and `YPS128` from the given dataset (Liti et al., 2009), whose chromosomes are compared against `RefSeq`'s. What we present in this paper can be easily extended to the rest of the strains in the above dataset. Moreover, given its speed, the method can be applied to the analysis of similar strains of species with possibly much longer genomes.

2 SYSTEM AND METHODS

Preliminary Data Analysis. Our approach is driven by data analysis. We performed a preliminary study to understand how to grasp the high similarity in our dataset. Consider a chromosomes' pair $(\texttt{Chr}N_{\texttt{A}}, \texttt{Chr}N_{\texttt{B}})$, where `A` is `RefSeq` and `B` is either `Y55` or `YPS128`; also, $1 \leq N \leq 16$ since the yeast genome consists of 16 chromosomes. Examine all the possible (overlapping) L-grams of $\texttt{Chr}N_{\texttt{B}}$ as candidates, where an L-gram is a segment of L consecutive bases.

Table 1: Statistics (%) for the L-grams ($L = 32$) satisfying properties (a)–(c). The length of each chromosome N in `Y55` is also given.

N	bases	(a)	(b)	(c)	N	bases	(a)	(b)	(c)
1	248 261	81.32	59.92	0.43	9	467 776	89.53	74.02	0.29
2	800 992	98.54	82.55	0.14	10	770 597	94.60	76.43	0.62
3	321 691	93.82	83.74	2.13	11	693 726	97.64	78.98	0.11
4	1 522 688	96.24	77.38	0.89	12	1 067 059	95.12	78.66	2.15
5	577 152	96.33	77.66	0.39	13	923 317	96.96	84.35	0.52
6	273 660	97.56	76.05	0.19	14	781 629	98.20	82.46	0.15
7	1 113 452	95.05	78.63	0.73	15	1 105 914	95.67	81.07	0.33
8	566 494	95.47	78.70	0.84	16	946 183	96.53	83.55	0.65

Assuming that $\texttt{Chr}N_{\texttt{B}}$ contains m bases, there are $m - L + 1$ L-grams, accounting for possible duplicates. Call *valid* the L-grams that do not contain any symbol `N`. The *common* L-grams are the valid L-grams that occur *exactly* (i.e. fully conserved with no mutation) both in $\texttt{Chr}N_{\texttt{A}}$ and $\texttt{Chr}N_{\texttt{B}}$.

In our experiments, $L = 32$ resulted to be a good choice, leading to the following empirical facts that were observed for chromosomes $N = 2, 3, \ldots, 16$, with chromosome $N = 1$ (whose percentages are shown inside parentheses below) being an outlier. The reported percentages are absolute, as they are obtained by dividing the number of wanted L-grams by $m - L + 1$.

(a) The *valid* L-grams are numerous: they are in the range 89.53%–98.54% in `Y55` and from 88.84% to 97.27% in `YPS128` (81.32% in `Y55` and 77.29% in `YPS128` for chromosome 1).

(b) The *common* L-grams are also numerous: they are between 74.02%–84.35% in `Y55` and between 71.71%–83.24% in `YPS128` (59.92% in `Y55` and 58.47% in `YPS128` for chromosome 1).

(c) The common L-grams that occurs *once* in each genome are the vast majority: indeed, those occurring two or more times are very few, between 0.11%–2.15% in `Y55` and 0.07%–1.93% in `YPS128`.

A summary reporting the above percentages for the L-grams in the 16 chromosomes of `Y55` is shown in Table 1. The implication of (a)–(c) is that we can localize the conserved regions using the common L-grams, as discussed next.

Conserved Regions. Our algorithm for the rapid detection of large highly-conserved segments, called REGENDER (REsident GENome DEtectoR), is driven by the above data analysis. It performs a two-phase processing of all the possible chromosomes' pairs $(\texttt{Chr}N_{\texttt{A}}, \texttt{Chr}N_{\texttt{B}})$, where `A` is `RefSeq`, `B` is either `Y55`

or YPS128, and N ranges from 1 to 16. In the first phase, REGENDER finds the common L-grams between Chr$N_{\mathtt{A}}$ and Chr$N_{\mathtt{B}}$. In the second phase, REGENDER aggregates consecutive L-grams in a greedy fashion using some user-defined parameters that control when the next conserved region begins in both Chr$N_{\mathtt{A}}$ and Chr$N_{\mathtt{B}}$. We provide the details of the algorithm in Section 3.

REGENDER is somewhat related to the *anchor-based* algorithms (Ohlebusch and Abouelhoda, 2006) that circumvent the quadratic costs. Such algorithms share a common mechanism. First, they build a dictionary to store the fragments or seeds that are common to both Chr$N_{\mathtt{A}}$ and Chr$N_{\mathtt{B}}$. Second, they extend the fragments/seeds into longer sequences called *anchors* using dynamic programming, except chaining algorithms (Ohlebusch and Abouelhoda, 2006). The sequence of anchors thus found are required to be *colinear*; namely, the anchors should occur in the same relative order inside both Chr$N_{\mathtt{A}}$ and Chr$N_{\mathtt{B}}$. Third, these algorithms apply an expensive dynamic programming scheme to the regions of Chr$N_{\mathtt{A}}$ and Chr$N_{\mathtt{B}}$ that are left uncovered by the anchors. Driven by our data analysis, REGENDER can go simpler. First, the L-grams of Chr$N_{\mathtt{A}}$ are stored in a hash table, and those of Chr$N_{\mathtt{B}}$ are searched in the table during a scan of Chr$N_{\mathtt{B}}$. The high similarity of Chr$N_{\mathtt{A}}$ and Chr$N_{\mathtt{B}}$ justifies our choice of exact L-grams as fragments. Second, our dataset gives almost surprisingly a natural set of anchors: contrarily to the anchor-based algorithms, we do not need any dynamic programming or chaining techniques to enforce the colinearity and the non-overlapping property, since there is almost a one-to-one mapping between the occurrences of the L-grams (see Section 2). Actually, we take advantage of the fact the L-grams overlap and, if they are not colinear, we get a hint for a possible translocation. As a result, REGENDER performs just a scan of Chr$N_{\mathtt{A}}$ and Chr$N_{\mathtt{B}}$. One execution of REGENDER takes less than a second on a standard PC with limited amount of memory. This is a major requirement, since we need to execute REGENDER for all pairs of corresponding chromosomes of Chr$N_{\mathtt{A}}$ and Chr$N_{\mathtt{B}}$. Third, we remark that we do not need a complete alignment of Chr$N_{\mathtt{A}}$ or Chr$N_{\mathtt{B}}$ for the purposes of the analysis performed in this paper. A high-quality alignment of the conserved regions in Chr$N_{\mathtt{A}}$ or Chr$N_{\mathtt{B}}$ is unnecessary in our case, as illustrated by the clear patterns emerging from Fig. 1. What we really care about is the description of the dynamics of the mobilome, identifying and locating all the mobile elements in the input sequences, together with the genomic rearrangements they are involved into. A merit of our approach is that of being able to select a small set of candidates for the latter investigation, as discussed next.

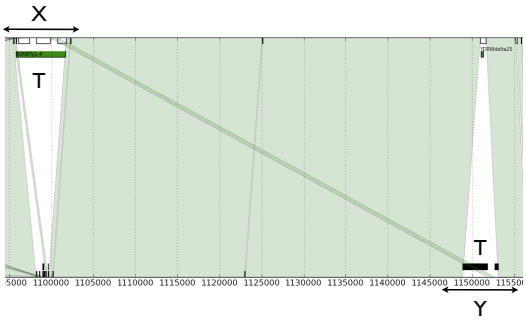

Figure 1: A plot of the common L-grams for Chr4 (1 095 000–1 155 000) of RefSeq (top sequence) and Y55 (bottom sequence), where $L = 32$. Each line connects the starting positions of a common L-gram. Thus, the empty triangles or trapezoids represent non-conserved regions. Annotated mobile elements are represented by the green rectangles just below the top line; unresolved sequences are the black rectangles just above the bottom line. High resolution plots are available online.

Non-conserved Regions. The outcomes of our experiments with REGENDER are analyzed as follows.

Graphically, we represent the two homologous chromosomes as two horizontal straight lines, and place A in the top and B in the bottom, as in Fig. 1. We mark the conservations with some color. The non-conserved regions are then detectable as non-colored trapezoids. The action of a transposable element T that has changed position from region X of strain A to region Y of strain B within two homologous chromosomes is then represented by two triangles (Fig. 1): we detect a white downward triangle inside region X (marking presence of T only in region X of strain A and absence in strain B), and an upward white triangle in Y (marking presence of T only in region Y of strain B and absence in corresponding position on strain A). Therefore, when strain A is the referring sequence RefSeq, we can infer that T probably moved from X to Y inside strain B by projecting region X of A onto the corresponding part in B.

A picture of possible situations is shown with some detail in Fig. 2 and described in section 4.

We followed the above conceptual scheme to collect statistics for all the chromosomal rearrangements among the 16 chromosomes' pairs from the selected strains (B is Y55 or YPS128) with the same chromosome in A=RefSeq, thus classifying any resulting rearrangement. We refer the reader to Section 4 for an aggregate view of all the chromosomal differences found and their relationships with the mobilome. We remark that we considered significant events that involve regions containing at least 200b, since very short indels or mutations are not linked with mo-

bilome nor with chromosomal rearrangements.

The proposed approach allowed us to obtain a fast and efficient localization of the resident genome, by working on a standard computer. Our results clearly show that the significant chromosomal indels involve almost exclusively the mobilome. Moreover, we show that unresolved sequences take place almost always in the correspondence of telomeres or mobile elements. Our approach allows us to infer putative insertions and deletions of transposons or LTR elements also in the presence of unresolved sequences.

3 ALGORITHM AND IMPLEMENTATION

As previously mentioned, we exploit the high similarity between genomes of different strains by running a massive computation involving all the possible chromosomes pairs $(\texttt{Chr}N_\texttt{A}, \texttt{Chr}N_\texttt{B})$, where `A` is `RefSeq`, `B` is either `Y55` or `YPS128` strain, and $N = 1, \ldots, 16$. We follow a two-phase approach for REGENDER, whose inputs are two chromosomes $\texttt{Chr}N_\texttt{A}$ and $\texttt{Chr}N_\texttt{B}$, the length L of the grams, and two user-defined parameters δ_1 and δ_2 to be used in the second phase. First, we find all the common L-grams between $\texttt{Chr}N_\texttt{A}$ and $\texttt{Chr}N_\texttt{B}$. Second, we detect highly conserved regions by aggregating consecutive L-grams. Finally, we inspect the non-conserved regions that are found by REGENDER, so as to infer mobilome elements. Due to space constraint we omit the implementation details.

Phase 1 of REGENDER: Common L-grams. We aim at finding which L-grams of $\texttt{Chr}N_\texttt{B}$ occur inside $\texttt{Chr}N_\texttt{A}$, where an L-gram is any sequence of L consecutive bases. First, we construct a dictionary for all the L-grams in $\texttt{Chr}N_\texttt{A}$ and, then, we search for the L-grams of $\texttt{Chr}N_\texttt{B}$ inside the dictionary. This task can be performed in expected linear time by employing a rolling hash approach based on cyclic polynomial, as described in (Cohen, 1997). Note that using a general purpose hash function would be more expensive by a multiplicative factor of L. Also, using a trie-based dictionary instead of hashing would guarantee a linear-time worst-case performance, but hashing is faster in practice.

A detailed description of the rolling hashing is beyond the scope of the current paper. However, it can be easily proved that the first phase of the algorithm REGENDER requires $O(|\texttt{Chr}N_\texttt{A}| + |\texttt{Chr}N_\texttt{B}|)$ time on average.

The output of the first phase is a mapping M, associating each L-gram s_2 of $\texttt{Chr}N_\texttt{B}$, with its occurrence list $occs(s_2)$ in $\texttt{Chr}N_\texttt{A}$. If s_2 does not occur in $\texttt{Chr}N_\texttt{A}$, $occs(s_2)$ is empty. Although not optimal in the worst case, our hash based approach turned out to be effective on our datasets, yielding few collisions, and allowing us to compare two entire chromosomes in few seconds. We implemented a prototype in Java, using the `fastutil` Java collections library to reduce as much as possible the memory usage (Vigna, 2006). The experiments have been performed on an Intel Core 2 Duo P8400 notebook, with 4GB of RAM. The code is single-threaded, and the maximum amount of RAM available for the first phase has been set to 200MB. The value of the parameter L has been set to 32, and the load factor of the hash table is set to $\alpha = 0.75$.

Phase 2 of REGENDER: Conserved Regions. During the second phase, the information about the L-gram occurrences, stored in the mapping M computed in the first phase, is used to establish a correspondence between segments of consecutive bases in $\texttt{Chr}N_\texttt{B}$ and $\texttt{Chr}N_\texttt{A}$, mapping a segment $I_2 = \texttt{Chr}N_\texttt{B}[l_2, r_2]$ into a corresponding segment $I_1 = \texttt{Chr}N_\texttt{A}[l_1, r_1]$. This information is represented by the mapping M_2, and it is graphically shown with green lines in Fig. 1.

We perform a left-to-right scan of $\texttt{Chr}N_\texttt{A}$ and $\texttt{Chr}N_\texttt{B}$, according to the following greedy rule. Initially, I_1 and I_2 are empty. During the scan, the current segments I_1 and I_2 are extended when the following conditions are met: (a) there exists a common L-gram s, which occurs both to the right of I_1 and I_2, and no other L-gram with this property can be found between I_1 and s, and I_2 and s; (b) letting d_1 be the number of bases between I_1 and s, and d_2 be the number of bases between I_2 and s, it is $|d_1 - d_2| \leq \delta_2$ and $d_2 \leq \delta_1$ (hence, $d_1 \leq \delta_1 + \delta_2$).

To compute the time complexity of the second phase of REGENDER algorithm, we observe that the sum of the sizes of the occurrence lists in M is upper bounded by $|\texttt{Chr}N_\texttt{A}| - L + 1$. In other words, the size of the mapping M is $O(|\texttt{Chr}N_\texttt{A}| + |\texttt{Chr}N_\texttt{B}|)$, hence the REGENDER algorithm requires $O(|\texttt{Chr}N_\texttt{A}| + |\texttt{Chr}N_\texttt{B}|)$ time on average.

Inspection of Non-conserved Regions. The contribution of REGENDER is that of reducing a potentially huge number of candidates to very few of them, so that the direct inspection of the non-conserved regions is doable. We perform this crucial analysis of the regions that have not been mapped into segments by M_2. These are the potential candidates for being mobile elements. Following, we discuss in details the performed analysis.

Table 2: Running time in seconds for each chromosome of the Y55 strain (columns). We run each tool with its default parameters, specifying, whenever possible, the L, δ_1, δ_2 arguments (rows). Experiments performed on an Intel Core 2 Duo P8400 notebook, with 4GB of RAM, running Ubuntu 10.04.

Chrm	Avid	Lagan	Lastz	GSAligner	Mauve	Murasaki	REGENDER
1	5.60	11.50	5.10	2.20	5.60	14.10	1.20
2	21.30	24.10	12.30	16.50	14.50	24.10	2.90
3	7.60	8.60	5.50	3	6	10.60	1.50
4	53.40	36.90	42.60	51.90	27.70	36.50	5.60
5	29.70	12.70	12.40	11.10	11.40	28.70	2.10
6	6.10	4.30	4.40	3.50	5.60	9.50	1.40
7	30.90	30.10	29.90	28.90	21.40	31.10	3.90
8	25.50	15.10	9	8	11.40	25	2.10
9	13.90	14	7.60	4.90	8.80	15.90	1.90
10	77.60	17.10	13.10	15.50	15.10	31	2.70
11	55.10	15.30	10.70	11.80	13.80	20.60	2.60
12	30.10	34.40	19	24.70	20.30	31.80	3.80
13	23.20	37.60	18	21.50	17.60	25.50	3.50
14	88.60	20.60	7.80	14.30	14.80	25.80	2.80
15	30.30	33.30	23.10	24.90	21.70	36.70	3.90
16	24.50	64.40	19	20.20	18.20	34.50	3.60

4 RESULTS AND DISCUSSION

REGENDER Performances. After the preliminary data analysis described in Section 2 that led to the choice of $L = 32$ for this dataset, we performed some experiments on all the 16 chromosomes of the two selected strains against RefSeq. REGENDER has proven to be very fast ($\delta_1 = \delta_2 = 100$): for instance, it can process the longest pair of chromosomes (Chr4, about 2Mb) in only 6 seconds.

In Table 2 REGENDER is compared with other existing tools, reporting the running time for each chromosome of the Y55 strain. Although the REGENDER prototype has been implemented in Python and Java, and the code has not been fine tuned, it is, on average, from four to ten times faster than the other tools. This is not surprising, since REGENDER does not perform a high-quality alignment of the input chromosomes as the other tools do (as explained in Section 2 this is not necessary in identifying and locating the mobile elements in the input sequences).

Qualitative Analysis of REGENDER Results. A local exploration of the results obtained by REGENDER showed the following scenario: most of the chromosomes appear to be conserved, therefore they are graphically covered by a uniform color given by the overlapping of parallel straight lines connecting identical L-grams. A set of examples is reported in Fig. 2, where the top line always represents a region of a chromosome of RefSeq, while the bottom line represents the same region in either Y55 or YPS128; here the mobile elements annotated in RefSeq are represented by green rectangles placed just below the top line, while unresolved sequences are represented by black rectangles placed just above the bottom line.

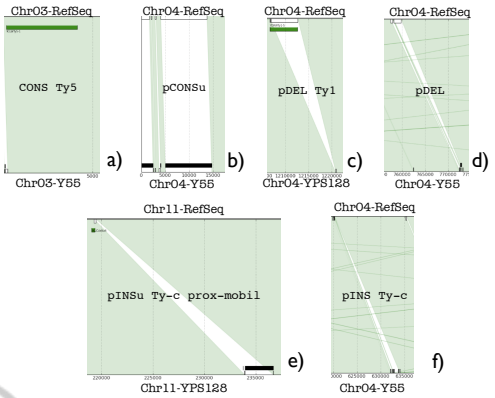

Figure 2: Features detected after REGENDER results.

Conserved regions belong to this category (including the mobile elements, when they are annotated on the RefSeq) and appear to be identically maintained in the screened strain. These regions are marked as CONS on Fig. 2(a).

This uniform coverage can be interrupted when, for example, the screened strain has a long run of unresolved bases. Such unresolved sequences are graphically marked by black rectangles. When the lines connecting their flanking regions are all parallel, it is likely that this fragment contains exactly the same sequence as RefSeq. In this case, we have an example of putative conservation, marked as pCONSu, that graphically appears as shown in Fig. 2(b).

Cases in which there is a sequence on RefSeq that has no correspondent on the homologous region of the screened strain are putative deletions. They can occur when a mobile element is annotated in RefSeq, in which case they are marked as pDEL-Ty or pDEL-LTR, if they occur for transposons or LTR, respectively. They are, instead, marked as pDEL when this putative deletion is not related to mobile elements (Fig. 2(c),(d)).

Putative insertions are more difficult to categorize, as the screened strain where they take place are not annotated. If the sequence is resolved, we employ standard alignment tools to search it in the RefSeq, trying to detect whether the fragment has actually been moved rather than deleted. On the other hand, when the sequence is unresolved, we can explore only two features. First, whether or not the length of the inserted sequence is compatible with either a transposon (when the length of the inserted sequence is $\geq$ 4000b) or an LTR (when the length of the inserted sequence is $\leq$ 500b). Second, whether these insertions take place in a region where in the RefSeq a mobile element is annotated at a distance less than 200b. For example, the event marked as "pINSu Ty-c prox-mobil" in Fig. 2(e) accounts for an inser-

tion (in the Chromosome 11) in YPS128 strain with respect to RefSeq, in an unresolved sequence. Since such insertion takes place less than 200b away from an LTR annotated in RefSeq, we consider this event as "proximal" to a mobile element. This is relevant, since several observations in the literature suggest that transposons prefer to migrate in zones where there are LTR. Finally, Fig. 2(f) shows an event of "pINS Ty-c" since the inserted sequence length is compatible with a transposon.

These cases represent a complete spectrum of the situations we have found in our screening. The following subsections report an aggregation of the data we collected for all these categories of events.

Conserved Regions and Mobile Elements. More than 95% in Y55 and 93% in YPS128 are conserved regions. Most of these are part of the resident genome, but not all of them. The fraction of conserved transposons or LTRs (i.e. mobilome) within conserved regions contains two possible elements: the truly conserved transposons (only 1) or LTRs (in a relative low number), which are annotated onto the RefSeq and exactly mapped on the screened strain and the putative conservations of annotated transposons or LTRs, which are mapped onto unresolved sequences in the screened strain: in this case, a direct attribution is impossible. The pCONSu are always found in the telomeres because the presence of long repeats is a source of noise for the assembly phase. In all cases but one, telomeres do not involve sequences related with mobile elements. Concerning pCONSu that are outside the telomeres, the number of unresolved sequences that are located in correspondence or in proximity of mobile elements, is greater than 90% for Y55 and around 70% for YPS128. This supports the hypothesis that unresolved regions are often located in correspondence of a mobile element (annotated onto RefSeq).

Deletions. Deletions concern almost only the mobilome. In Y55 strain, for instance, there are four putative deleted regions on RefSeq that do not correspond to annotated Ty or soloLTR (against more than 90 pDELs corresponding to mobilome annotations). We found that the length of the two regions is compatible to that of a soloLTR. This evidence strongly suggests that in genomes closely related, the only significant (i.e., for our work, those involving sequences at least 200 long) chromosomal rearrangements are due to the mobilome.

Insertions. The landscape for the putative insertions, without (pINS) and with (pINSu) unresolved regions is rich. We may label the inserted sequences by proximity to annotated Ty or soloLTR in the insertion site on RefSeq: from 40% to 50% of the cases, it is a putative mobilome-proximal insertion. As for deletions, we may also distinguish on the basis of inserted sequence length: Ty-c, LTR-c or in-between. Also in this case, the large majority of events are concerned with the mobilome. Even if this result is partly derived from our classification methods, it supports the same conclusion anyway.

5 FUTURE DIRECTIONS

We proposed an approach aimed at a rapid and efficient localization of the resident genome through algorithm REGENDER. We shall generalize this approach to a multiple comparison extracting all the chromosomal rearrangements on a dataset of 39 strains of the same specie *S. cerevisiæ* (Liti et al., 2009). Our long term goal is to develop a model able to describe the dynamics of the mobilome in these strains.

ACKNOWLEDGEMENTS

We thank Emiliano Biscardi for performing tests whose results are reported in Table 2.

REFERENCES

Cohen, J. D. (1997). Recursive hashing functions for n-grams. *ACM Trans. Inf. Syst.*, 15(3):291–320.

Conti, V., Aghaie, A., Cilli, M., and *et al.* (2006). crv4, a mouse model for human ataxia associated with kyphoscoliosis caused by an mrna splicing mutation of the metabotropic glutamate receptor 1 (grm1). *Int. J. Molec. Med.*, 18:593–600.

Le Rouzic, A., Boutin, T. S., and Capy, P. (2007). Long term evolution of transposable elements. *PNAS*, 104:19375–19380.

Lewin, B. (2007). *Genes (IX ed.)*. Jones and Bartlett.

Liti, G., Carter, D. M., Moses, A. M., and *et al.* (2009). Population genomics of domestic and wild yeast. *Nature*, 458:337–341.

Ohlebusch, E. and Abouelhoda, M. (2006). Chaining algorithms and applications in comparative genomics. *Handbook of Computational Molecular Biology*.

SGD (2010). SGD project. Saccharomyces Genome Database. http://www.yeastgenome.org.

Venner, S., Feschotte, C., and Biemont, C. (2009). Dynamics of transposable elements: towards a community ecology of the genome. *Trends Genet.*, 25:317–323.

Vigna, S. (2006). fastutil: Fast and compact type-specific collections for java. http://fastutil.dsi.unimi.it.