



## Pooled library tissue tags for EST-based gene discovery

A. J. Gavin<sup>1</sup>, T. E. Scheetz<sup>1</sup>, C. A. Roberts<sup>1</sup>, B. O'Leary<sup>1</sup>,  
T. A. Braun<sup>1</sup>, V. C. Sheffield<sup>2</sup>, M. B. Soares<sup>2,3</sup>, J. P. Robinson<sup>1</sup>  
and T. L. Casavant<sup>1,\*</sup>

<sup>1</sup>Departments of Electrical and Computer Engineering, <sup>2</sup>Pediatrics and <sup>3</sup>Physiology and Biophysics, The University of Iowa, Iowa City, IA 52242, USA

Received on December 14, 2001; revised on March 27, 2002; accepted on March 31, 2002

### ABSTRACT

**Motivation:** In gene discovery projects based on EST sequencing, effective post-sequencing identification methods are important in determining tissue sources of ESTs within pooled cDNA libraries. In the past, such identification efforts have been characterized by higher than necessary failure rates due to the presence of errors within the subsequence containing the oligo tag intended to define the tissue source for each EST.

**Results:** A large-scale EST-based gene discovery program at The University of Iowa has led to the creation of a unique software method named UITagCreator usable in the creation of large sets of synthetic tissue identification tags. The identification tags provide error detection and correction capability and, in conjunction with automated annotation software, result in a substantial improvement in the accurate identification of the tissue source in the presence of sequencing and base-calling errors. These identification rates are favorable, relative to past paradigms.

**Availability:** The UITagCreator source code and installation instructions, along with detection software usable in concert with created tag sets, is freely available at <http://genome.uiowa.edu/pubsoft/software.html>

**Contact:** tomc@eng.uiowa.edu

### INTRODUCTION

Accurate and efficient methods for feature detection, identification, and annotation are an integral part of the overall goals of gene discovery projects. With the discovery of a new gene, it is beneficial to also determine the region of gene expression as one part of the annotation of each sequence. For cDNA libraries derived from single tissues, identification of the tissue source is trivial. However the use of pooled (multiple-tissue) libraries (Bonaldo *et al.*, 1996), which are more efficient for gene

discovery, makes determination of the tissue of origin more difficult. As a result, the creation of an intelligent model for identification of the tissue of origin is needed in cases where the cDNA is derived from a pooled library. This model must also be very efficient as several hundreds or even thousands of cDNAs are processed on a daily basis in high-throughput sequencing projects.

At The University of Iowa, the method used to identify the tissue of origin utilizes a synthetic oligonucleotide tag to uniquely identify the source tissue from which the clone was derived. Unfortunately, the presence of errors within sequences is inevitable and may affect the tissue tag. The presence of such errors complicates the accurate identification of the tissue source, adding another level of complexity to our model. The ability to detect and, when possible, correct such errors creates a more efficient method to identify the source tissue from which a cDNA clone is derived.

We have developed a process for the creation of library tags that increases accuracy in identification of the source tissue with little processing time overhead. There have been three generations of tissue tags, with each successive generation possessing increased error detection and correction capabilities. The first generation consisted of *ad hoc* variable-length tags, shown in Table 1, with little error detection or correction capability. For example, the tissue tags for kidney and liver differ in only one position. Thus a single substitution error from an A to a C at the third position would change the identified tissue from kidney to liver. The second-generation of tissue tags developed, shown in Table 2, added the ability to detect and correct for single-substitution errors within the tag sequence. This was achieved by designing tag sets such that each tag in the set was at least Hamming distance 3 (i.e. different at three or more positions) from every other. The third generation of tissue tags, shown in Table 3, adds the ability to detect and correct for single insertion or deletion errors or up to two substitution errors.

\*To whom correspondence should be addressed.

**Table 1.** First-generation motif example pooled library set

Library source tissue	Tissue motif
Brain	TAGAG
Heart	ACAAC
Kidney	CAAAC
Liver	CACAC
Lung	TTCCA
Muscle	AAG
Ovary	TCAC
Placenta	ATGTG
Spleen	GAGA

**Table 2.** Second-generation motif example pooled library set

Library source tissue	Tissue motif
Amygdala	GTGAG
Basal ganglia	TGTAC
Brain stems	TCATG
Cerebellum	GACTC
Corpus striatum	ACGGC
Hippocampus	TTCGA
Hypothalamus	CGGTA
Olfactory bulbs	CATGG
Prefrontal cortex	GCTCA

**Table 3.** Third-generation motif example pooled library set

Library source tissue	Tissue motif
Seminal vesicles	GTGATTACAC
Penis	TTGCGAACA
Rat heart pool	ATAAGATAAC
Rat kidney pool	CAAGACTGTC
Rat aorta pool	CTGTAGGATC
Rat placenta pool	TCACGACAGT
Unused source	GAAGTGCTCC
Unused source	GAATAATACA
Unused source	TCAGTGCTA

## SYSTEM AND METHODS

The creation of robust tissue tag sets is extremely important for cDNA clones derived from pooled libraries. Unless an intelligent model is used in the formulation of tag sets, the inevitable presence of errors within the observed tissue tag sequence renders accurate identification of the EST tissue source extremely difficult. Such errors result in the reporting of false positive matches that can introduce inaccuracies into the system affecting other parts of the production sequence annotation pipeline. In practice, two major classes of errors exist which are relevant in de-

tection of a tissue tag. These error types are *substitution* errors, also called misreads, and *indel* errors, which include insertions and deletions of bases in a sequence. The prevalence of substitution and indel errors in our EST sequencing projects are 4% and 0.5% respectively (T. Braun, data not published). The errors may occur at any of several stages (e.g. cDNA creation, DNA sequencing) either due to imperfect fidelity of the polymerase, cloning artifacts, or imperfect oligo synthesis.

A substitution error is one base (e.g. an 'A') misread as another (e.g. as a 'C'). Shown below is an example of a segment of a single 3' EST that contains a tissue tag (GAGTC) of length 5 from a particular library. The tissue tag is preceded by a NotI restriction site (GCGGCCGC) and followed by a polyadenylation tail.

### Example Segment 1:

... ATCTGCGGCCGCGAGTCTTTTTTTT...  
(no errors)

The following subsequence contains an example of a tissue tag with a substitution error in the second position of this same EST. In such cases, the tag is said to be 'Hamming distance one' away from its ideal form (Weldon and Peterson, 1972).

### Example Segment 2:

... ATCTGCGGCCGCGCGTCCTTTTTTTT...  
(substitution error)

The second class of errors, insertions or deletions, makes correct detection of the tissue tag significantly more difficult because the length of the tag is altered. This creates framing problems during the detection of the tag. Based on the previously presented clone subsequence, the following two cases illustrate the two possible types of indel errors.

### Example Segment 3:

... ATCTGCGGCCGCGAGCTCTTTTTTTT...  
(indel error—insertion)

### Example Segment 4:

... ATCTGCGGCCGCGAG\_CTTTTTTTTT...  
(indel error—deletion)

In the first case, an insertion of the underlined 'C' occurs between the original third and fourth bases of the tissue tag. In the second case, the fourth base has been deleted from the tissue tag and could result in the detection of an alternate tag of length four. In the presence of indel errors, we use the notion of edit distance (also known as the Levenshtein distance) as a metric of the divergence between two tags. The edit distance is the minimum

number of errors—substitution or indel—between two tags. In all three prior examples of possible errors the edit distance is equal to one. A thorough examination of edit distance can be found in Gusfield (1997).

The rationale behind the use of edit distance is simple; defining a frame of reference with which to compare two tag sequences becomes a problem in the presence of indel errors because the apparent tag length differs. When detection is limited to substitution errors, a comparison of motifs from the ideal and first indel error case results in two *different* measurements for the Hamming distance. This is possible due to the unequal lengths, which changes the Hamming distance depending on which end of the longer motif the alignment begins. If the comparison is anchored further upstream (i.e. to the left), the comparison of the two motifs GAGTC and GAGCT (from Example Segment 3) shows a Hamming distance of 2. However, if anchored on the downstream end (i.e. to the right), the comparison of the ideal tag GAGTC to AGCTC returns a Hamming distance between tags of 3. As mentioned previously, edit distance is robust in the presence of indel errors and should provide a consistent measure regardless of framing.

We have used several methodologies in detection and identification of the tissue source throughout the lifetime of our gene discovery projects (<http://genome.uiowa.edu/>). The most primitive method, utilizing an exact-match criterion, results in accurate identification of the tissue source in the absence of errors, but lacks the capability to correct for errors within the subsequence. A second strategy allows for the detection and correction of substitution errors within the tag. Although this method greatly increases detection rates, a small portion of the increase is caused by misidentification due to indel mapping between second-generation tags. The next logical evolution in the tag generation and identification software is the added ability to detect and correct both substitution and indel errors, thus increasing the detection rate while limiting the number of false positives introduced into the results. UITagCreator was developed to generate sets of such tags, and EST-prep (Trivedi *et al.*, 2001) incorporates the ability to detect these tags within ESTs.

## ALGORITHM

### Linear feedback shift register

A linear feedback shift register (LFSR; Weldon and Peterson, 1972) with a primitive feedback polynomial can iterate through all possible non-zero register contents in a pseudo-random order. We use the LFSR to generate candidate tags to be evaluated. The candidate tag is formed by mapping every two binary bits to one of the four nucleotides. The use of a maximum length linear feedback shift register ensures a mutually exclusive and exhaustive search of all possible combinations in a

pseudo-random manner. A pseudo-random search strategy was selected because it was empirically found to yield more tags than systematic (e.g. lexicographic order) or pure random search.

Consider generating a set of tags of length 10. There is an equivalent binary representation of length 20. Using Weldon and Peterson (1972), we select one minimum term primitive polynomial of the 20th degree, 4000011, in octal format. The non-zero terms are the 20th, third, and zeroth power, respectively. Thus, the bit positions at which XOR operations are performed are at the 20th, third, and zeroth bit positions, respectively. From the same reference, the prime factors of the shift period can be found and include 3, 5, 11, 31, and 41. The number of shifts required to cycle through all non-zero binary candidate words must be prime with respect to the prime factors of the shift period. Although the minimum number of shifts is beneficial in reducing the computational overhead, a shift length of 1 may not be desirable due to an associated limitation in the 'space' from which the candidate tags are drawn.

### Calculating edit distance

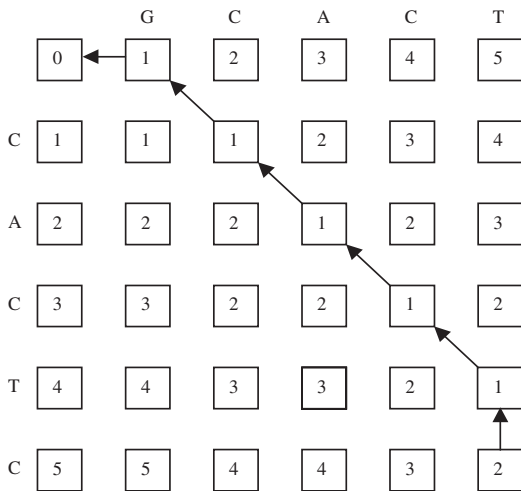
The method used to calculate edit distance is similar to the Needleman–Wunsch global alignment algorithm (Needleman and Wunsch, 1970). An in-depth description of the global alignment algorithm, including an illustrated example, can be found in Setubal and Meidanis (1997).

The edit distance between two tags of the same length is calculated using the Levenshtein distance algorithm (LDA; Levenshtein, 1966). The LDA differs from the Needleman–Wunsch algorithm in using a penalty of +1 for both indel and mismatch errors, while having a match reward of 0. Figure 1 shows the Levenshtein matrix (also referred to as the score matrix) used to determine the distance between two tags (tag1, CACTC, on the left, and tag2, GCACT, on the top).

In this example, the edit distance is 2, the value contained in the lower right-hand cell of the Levenshtein matrix. The alignment can then be calculated by backtracking from the bottom-right to the upper-left similar to the procedure used by the Needleman–Wunsch algorithm. Note, however, that the calculation of the alignment is not necessary to determine the edit distance.

### Creation of third generation tags

The software algorithm used to generate the third generation set of tissue tags (UITagCreator) utilizes a pseudo-random search to identify candidate tags. The program requires three data as inputs: tag length, edit distance, and Hamming distance. Candidate tags are evaluated iteratively for addition into the growing tag set. The algorithm utilizes the LDA to determine the edit distance between pairs of tags, and implements a linear-feedback shift register (LFSR) to quickly perform an exhaustive



**Fig. 1.** Levenshtein matrix for calculating edit distance between motifs.

pseudo-random search of each candidate tag. As described previously, a candidate tag is encoded within the state of the LFSR. Beyond length, the only constraint on selection of the initial tag is that it must not be homogeneous, i.e. it must contain at least two unique bases. The set of tags that is generated may not be of *maximal* size, however it is an *optimal* set, i.e., no other valid tags of the correct length exist that could be added to the set without violating the edit or Hamming distance criteria.

The initial tag for the set is inserted into the tag set as the initial tissue tag. Each new candidate tag generated by the feedback shift register is compared against every tag in the current tag set to ensure it meets the desired minimum edit distance. The candidate tag is first assessed against the entire set for Hamming distance. Only if it passes the Hamming distance criteria will the edit distance be checked. In empirical tests, this strategy reduced the number of edit distance calculations by 40%.

### Tag detection

Prior to tag detection, the sub-sequence corresponding to the putative tissue tag must be identified. The basic algorithm used to accomplish this is as follows. First, the EST sequence must pass an initial quality assessment. Next, the restriction site and upstream vector sequence must be identified. A sequence that fails either of these steps is removed from further processing. The next step is the identification of the polyA tail. As shown in Example Segment 1, the cDNAs in our libraries have the tissue tag located between the NotI restriction site and the polyA tail. Thus the putative tissue tag sequence can only be determined for those sequences with an identified polyA tail, and the success rate for tag detection is based only on those sequences with an identified polyA tail. To

```

potential_tag_sequence ∈ {anchored at restriction site, frame+1, frame-1}
∀ tag ∈ {TAGS_IN_LIBRARY}
if(edit_distance(potential_tag_sequence,tag) < ERROR_LIMIT)
return(tag)

```

**Fig. 2.** Pseudo-code algorithm for library tag detection.

accommodate for errors in accurate identification of the boundaries between the restriction site and tissue tag, and between the polyA tail and the tissue tag, the tissue tag is evaluated across multiple frames. The initial frame begins at the detected end of the restriction site. Successive iterations utilize frames that are single base shifts from the initial frame.

Each candidate tag is evaluated against a set of valid tags to be identified. If a candidate tag is found that is less than *ERROR\_LIMIT* away from a valid tag ( $T_1$ ), the identified tag is noted as  $T_1$ . The set of tag candidates includes the originally defined sub-sequence between the restriction site and the polyA tail. If no matching tag is found, the set of cumulative left and right shifts from the initial candidate tag is used. The number of candidate tag shifts, *ERROR\_LIMIT*, and list of valid tags are all configurable options to ESTprep. The current implementation does not differentiate between indel errors and substitution errors, but an updated version is planned. It should be noted that although the correct tag can be determined in the presence of errors, the tag sequence within the EST sequence is *not* changed. Tag-less ESTs are acceptable within our sequence processing pipeline, but are lacking the tissue of origin annotation

### Implementation and results

Much of the previous algorithm description centered on methods to create an optimal set of third-generation tags from a non-homogeneous initial candidate. Generating an optimal set rather than a maximally spaced set is necessary due to the extremely long runtime the search for such a set of tags of any length requires. A maximally spaced set would require evaluating all possible combinations of tags for a specified tag set size.

Consider the scenario where one wishes to generate the best-possible set of 25 tags, each of length 5 ( $L = 5$ ). The number of candidates is equal to  $4^L$ , meaning that 1024 possible candidates exist from which 25 must be selected to be included in the set. In this case, the number of unique sets that exist, computing the combination of 1024 candidate tags taken 25 at a time, is approximately  $8.68 \times 10^{49}$ . To put this number into perspective, let us assume the edit distance properties of one set of 25 tags can be assessed in one 1 ns of CPU time. Assessing all sets would require  $10^{23}$  computers the estimated lifetime of the Universe (15 billion years) to complete. Clearly this is an unreasonable approach. A rational alternative is to

satisfy one's requirements using an optimal set. A single optimal set of length 10 tissue tags was constructed in 30 s on a 600 MHz Pentium III processor. This set contained over three hundred tags of minimum edit distance 3 and minimum Hamming distance of 5.

A set of third generation tags of length 10, minimum Hamming distance 5, and minimum edit distance of 3 were generated and synthetically inserted into clones in single-tissue cDNA libraries that were later pooled. The detection rates calculated using the third generation tag set (95.4%) showed significant improvement over both first- and second-generation tag sets (84.1% and 94.2% respectively). Of over 8000 EST sequences from pooled libraries with third generation tags that were found to have tail, more than 7700 sequences had a properly identified valid library tag, a detection rate of 95.4%. A post-mortem analysis of failure indicated that the majority of failures were due to excessive accumulation of errors. For the third-generation tags these were primarily multiple substitution errors, or an indel and one substitution error. An updated version of the detection algorithm that properly differentiates between substitution and indel errors could thus correct many of these cases.

## DISCUSSION

As expected, computation time increases exponentially with the tag length. As the tag length increases, the number of possible candidates that satisfy the minimum Hamming distance required become significantly greater, and most of the compute time derives from calculation of the edit distance.

Although the detection rate for third generation tags is only slightly greater than for second generation tags they are significantly more robust with respect to indel errors. First, the detection rate for the second generation tissue tags is slightly inflated, due to indel errors that mapped one correct tag into another. However, such occurrences are rare. Second, the third generation tags are twice the length of the second generation tags. Thus there should be a greater number of errors that accumulate within the tag sequence. Finally, the detection software used in conjunction with the third generation tags did not differentiate between substitution and indel errors. A more sophisticated approach would allow accurate tag identification in the presence of one indel error or up to two substitution errors.

As presented earlier use of the above methods substantially increases detection rates in pooled libraries by simplifying the problem of defining a reference frame between two motifs. Using the LDA as the criteria to generate tag sets of a desired minimum edit distance, one can accurately detect and correct errors in the tissue tag. The UITagCreator program allows creation of an optimal set of robust tissue identification tags with similar error

detection and correction properties to the best-possible set while offering drastic improvements in performance with regard to execution time. Care must be taken during tag selection, beyond the verification of the required error correction and detection properties. Specifically, tags should not contain a recognition sequence for any restriction enzyme that will be used during library construction. To avoid formation of secondary structures (e.g. self-annealing), sequences highly similar to any other portion of the primer used during reverse transcription should also be excluded. Currently, this is accomplished through a post-processing filter.

The tags generated by the UITagCreator can also be used as robust identifiers to label strains of interest. The error detection and correction properties of these tags should be robust with respect to errors caused by point mutations. Future extensions to the UITagCreator program include specification of search order—lexical or pseudo-random, and a generalized method for specifying sequences that are not allowed within the final set of tags.

## ACKNOWLEDGEMENTS

We thank all the members of the Coordinated Laboratory for Computational Genomics (Casavant lab) and the Library Creation and Sequencing Groups (Soares lab). Specifically, we would like to thank T.A.Kucaba for supervision and coordination of the EST sequencing. This work was supported in part by NIH grant 2R01HL59789. T.E.S. and T.A.B. were partially supported by a NIH training grant GMO8629-04. V.C.S. is an Associate Investigator of the Howard Hughes Medical Institute.

## REFERENCES

- Bonaldo,M.F., Soares,M.B. and Lennon,G. (1996) Normalization and subtraction: two approaches to facilitate gene discovery. *Genome Res.*, **6**, 791–806.
- Conway,J.H. and Sloane,N.J.A. (1986) Lexicographic codes: error-correcting codes from game theory. *IEEE Trans. Information Theory*, **32**, 337–348.
- Gusfield,D. (1997) *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, New York.
- Levenshtein,V.I. (1966) Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phys. Dokl.*, **6**, 707–710.
- Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.
- Setubal,J.C. and Meidanis,J. (1997) *Introduction to Computational Molecular Biology*. Brooks/Cold Publishing Company, Pacific Grove, California, pp. p. 296.
- Trivedi,N., Roberts,C.A., Gavin,A.J., Robinson,N.L., Birkett,C.L., Scheetz,T.E. and Casavant,T.L. (2001) Technical report, TR-ECE-20011213.
- Weldon,Jr,E.J. and Peterson,W. (1972) *Error-Correcting Codes*, 2nd edn, The MIT Press, Cambridge, pp. 560.