Automated clustering and assembly of large EST collections

David P. Yee and Darrell Conklin
ZymoGenetics, Inc.
1201 Eastlake Ave. E., Seattle, WA, USA 98102
email: {yee,conklin}@zgi.com
phone: 206-442-6{707,664}
fax: 206-442-6608

Abstract

The availability of large EST (Expressed Sequence Tag) databases has led to a revolution in the way new genes are cloned. Difficulties arise, however, due to high error rates and redundancy of raw EST data. For these reasons, one of the first tasks performed by a scientist investigating any EST of interest is to gather contiguous ESTs and assemble them into a larger virtual cDNA. The REX (Recursive EST eXtender) algorithm described in this paper completely automates this process by finding ESTs that can be clustered on the basis of overlapping bases, and then assembling the contigs into a consensus sequence. By combining the clustering and assembly steps, REX can quickly generate assemblies from EST databases that are frequently updated without having to preprocess the data. A consensus assembly method is used to correct miscalled bases and remove indel errors. A unique feature of this method is that it addresses the issues of splice variants and unspliced cDNA data. Since REX is a fast greedy algorithm, it can address the problem of generating a database of assembled sequences from very large collections of EST data. A procedure is described for creating and maintaining an Assembled Consensus EST database (ACE) that is useful for characterizing the large body of data that exists in EST databases.

Introduction

The large and rapidly growing databases of ESTs (Expressed Sequence Tags) (Boguski et al. 1995) contain a wealth of information on both known and unknown genes. There are about 1.5 million ESTs available in public databases, and at least that number exist in private and proprietary databases. EST sequencing is an efficient way to "tag" a large number of expressed genes. ESTs are obtained by short (roughly 300bp), single-pass reads of clones from selected cDNA libraries. There are a number of novel genes that have been identified from EST databases (e.g., Bailleul et al. 1997; Lin et al. 1997; Wu et al. 1997; Yamada et al. 1997).

As EST databases have grown, using these data in their raw form has become inefficient for two main reasons. First, ESTs inherently have a high sequencing error rate, especially toward their 3′ ends. Second, several ESTs can tag the same gene, and since an unnormalized cDNA library reflects the abundance of the mRNA in the underlying tissue source, high abundance messages are more likely to be sequenced multiple times than low abundance ones. This redundancy of high abundance messages makes mining of raw EST databases inefficient.

A powerful way to manage this redundancy is to assemble clusters of ESTs representing the same message into longer virtual cDNA sequences. There are several advantages to working with assemblies rather than individual ESTs: first, there are fewer sequences to analyze. Second, the assembled sequences are longer and potentially contain more interpretable coding sequence than their individual component ESTs. Third, with sufficient coverage, miscall and indel errors present in individual component ESTs have a good chance of being removed during the assembly process. Fourth, the virtual cDNA sequences may be full-length, greatly facilitating cloning of the gene in the lab.

It is useful to make a conceptual distinction between the clustering and the assembly processes. The goal of the clustering process is to incorporate overlapping ESTs which tag the same transcript of the same gene into a single cluster. This process is usually done offline and can be extremely time consuming due to the intrinsic need for all pairs of ESTs to be tested for overlap. Once clustering is complete, one or more consensus assemblies for each individual cluster can be produced. For example, the TIGR Assembler (Sutton et al. 1995) first creates a table of potential gene overlaps (clustering) and builds assemblies using appropriate sequences from the overlap list. STACK (South African National Bioinformatics Institute 1997) creates all clusters and then assembles these using the TIGR assembler. UniEST (Boguski and Schuler 1995) contains clusters but not assemblies of EST data.

The first part of this paper describes REX (Recursive EST eXtender), a simple yet effective greedy algo-

rithm which completely integrates the clustering and assembly steps for EST data. It generates an assembly concurrently with the clustering process. A major feature of REX is the ability to rapidly generate assemblies from an arbitrary set of EST databases without the need to preprocess the data.

The second part of this paper describes a process for managing and maintaining large numbers of EST assemblies generated by the REX algorithm described in the first section. Creating a database of fully assembled ESTs requires an enormous amount of computer time. The process described in this paper allows the assembled database to be updated as new data becomes available. This alleviates the need to completely redo the assembly computation and ensures that the assembled database is kept current. Finally, a unique feature of this method is that it specifically recognizes and addresses the problems of splice variant and unspliced cDNA data, in that multiple assemblies for the same gene may be created.

The REX algorithm

The REX algorithm is a simple yet effective algorithm which integrates the clustering and assembly steps necessary to create a consensus assembled sequence from EST data. It is fast and reflects an intuitive idea of extending an anchor sequence in the 5′ to the 3′ direction with ESTs derived from an arbitrary number of sequence databases. A rough assembled sequence is created in the process of extending an anchor. This rough assembly is then improved by analyzing a multiple alignment of contigs to make consensus base calls and to remove potential insertion and deletion errors.

After this work was completed we became aware of the GAP system (Gill et al. 1997), which describes an interface for interactive EST assembly in which a scientist can probe a database for ESTs to extend a current consensus assembly. The iterative process is entirely manual, and constructing a single assembly could be a time consuming exercise for the scientist, particularly if the assembled sequence is long. REX completely automates the clustering and assembly of a virtual cDNA sequence using a single specified sequence as an anchor.

Contigs

Given an anchor sequence, REX finds contiguous ESTs (contigs) by using blastn (i.e., ungapped local alignment, X=5, W=16, E=1e-15, Z=5e+8, Y=300, M=5, N=-4) (Altschul et al. 1990) on one or more EST databases using the anchor sequence as a query. The Y and Z parameters are set in order to normalize the blastn expectation values between different database sizes and anchor sequence lengths. This set of parameters requires approximately a 70 base pair overlap between two ESTs. It was determined that these parameters are sufficiently liberal to accommodate EST sequencing errors, and also that roughly 95% of all human paralogous gene families will be separated (data not shown). (The ortholog issue is easily addressed by using species-specific EST databases). In problematic cases, where two different genes are equivocated, the stringency of overlap can be adjusted with the E (blastn expectation value) parameter.

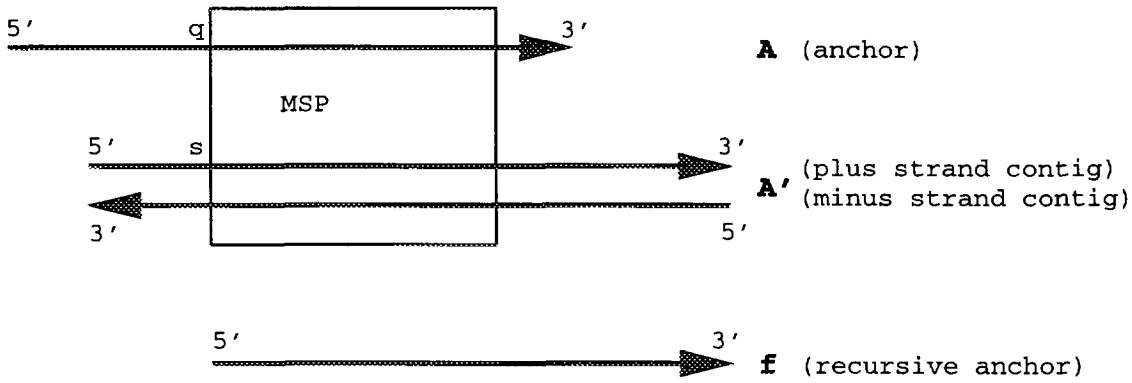
During the blastn process, an anchor sequence is masked using a very fast sequence filter which replaces vector sequences, ALU, LINE, and other repetitive elements (including stretches of 30 or more identical bases) by the base "N". These masked regions will therefore never be used to identify contiguous ESTs. All reported blastn hits using the parameters given above are considered to be contigs; these contigs are retained in a list for later use by the consensus assembly process (described below). Contigs which extend the anchor in the 3′ direction are considered for use as anchors for recursive extension. REX is a greedy algorithm, choosing only one contig to extend an anchor sequence. Below the algorithm is described, then the procedure for choosing the extender contig is explained in detail.

Building an extended sequence

Figure 1 illustrates the REX algorithm. First, all extender contigs to an anchor sequence are gathered. An extender contig is any contig which extends the anchor sequence in the 3′ direction. The coordinates of the Maximum-scoring Segment Pair (MSP) (i.e., the highest-scoring blastn High-scoring Segment Pair (HSP) or ungapped local alignment) fix the alignment with the chosen extender contig. The appropriate segment (sequence fragment **f** in Figure 1) is cut out of the extender contig, and that segment is used recursively as an anchor. If the extender contig is in the minus strand, the segment is reverse complemented before being used as a recursive anchor. The recursion ends when no extender contig can be found. The details of how to modify the MSP coordinates for a minus strand contig have been omitted.

A slight distinction is made in REX between an anchor sequence encountered during recursive extension and the "top-level" anchor sequence specified by a user or calling process. The distinction is that the top level anchor sequence must be extended in both the 3′ and the 5′ directions. To accomplish this, the anchor is extended: next, the reverse complement of the anchor is extended. To preserve the orientation of the original anchor sequence, the reverse complement of the second result is taken and the results of the 5′ and the 3′ extensions are spliced together to produce the final extended sequence (see Figure 1).

Note that the algorithm does not actually build the assembly until it starts to unwind from the recursion. This means that only single ESTs, rather than complete assemblies as in the TIGR and GAP assemblers, are used to gather contigs. The time taken by the REX algorithm to assemble a virtual cDNA is primarily dependent on the number of recursive extension steps




```
/* extend an anchor sequence A */
rex(A) {
      if A has no extender contigs, return A
      else
            let A' be the chosen extender contig to A
            let (q,s) be the left coordinates of their MSP
            let f be the sequence fragment A'[s...length(A')]
            if A' is a minus strand contig, return
                  A[1...q-1] + rex(revcomp(f))     /* recursion */
            else return
                  A[1...q-1] + rex(f)              /* recursion */
}

/* extend the top-level anchor sequence A */
let r be the sequence rex(A)
let l be the sequence revcomp(rex(revcomp(r[1...length(A)])))
return l + r[length(A)+1...length(r)]
```

Figure 1: The REX algorithm. $S[x...y]$ denotes the substring of S from index x through y. + in a string context represents string concatenation.

needed (i.e., the number of blastn runs) and is otherwise independent of the length of the assembly. The TIGR assembler, on the other hand, iteratively merges similar fragments to a growing current assembly using the Smith-Waterman algorithm and it presupposes the existence of a precomputed table of pairwise EST overlaps.

Choosing an extender contig

If all mRNA transcripts derived from the same gene were identical it would be irrelevant which extender contig was chosen since (aside from error-laden ESTs) roughly the same virtual cDNA will be computed by the REX algorithm. Two complications with ESTs, however, are that they can be derived from alternatively spliced mRNA and unspliced (intron containing) hnRNA (Wolfsberg and Landsman 1997). REX does not choose ESTs which appear to be unspliced or alternatively spliced in relation to the anchor sequence. To check for alternatively spliced ESTs, it is required that the quantity $(q' - q) - (s' - s)$, is in a narrow interval (see Figure 2: bottom). To check for unspliced ESTs, the maximum possible overlap between the extender and anchor (see Figure 2: top) is computed based on the coordinates of the MSP. The summed length of all consistent HSPs is required to cover a minimum percentage of this region. In practice it is not possible to distinguish between unspliced and alternatively spliced messages in short EST data. Consequently, REX can only recognize that an EST may fall into one of these categories. All such ESTs are flagged by the REX algorithm and are not chosen as extender contigs for this particular run.

There are several modes for choosing one out of many possible consistent extender contigs. The first mode chooses the extender contig with the maximum percent identity when averaged over all consistent HSPs. In the second mode, the extender contig which extends the sequence the furthest 3′ is chosen. The second mode is usually faster since fewer recursive extension steps generally need to be performed. It is also possible to choose the extender contig with the most coverage of the potential overlap region.

Consensus assembly

REX differs from other assembly methods in the way the final consensus assembly is created. The TIGR assembler (Sutton et al. 1995), for example, represents a rough assembly as a frequency profile from which consensus base calling is performed. REX represents the rough assembly as a single extended sequence which is a composite of fragments from extender contigs used during the extension process. As a result, sequencing errors in these individual fragments will also be in the rough assembly. To correct sequencing errors, a consensus assembly is created by aligning the blastn2 (i.e., gapped local alignment, parameters as above for blastn, with Q=14) MSP of all contigs (which were gathered during the extension process) with the rough assembly. This provides a full multiple alignment of all contigs. A frequency profile is built for every column of this multiple alignment, and consensus base calls are made using a set of rules similar to those used by the TIGR assembler (Sutton et al. 1995). In addition, we use a novel rule which handles common base expansion errors as often encountered in EST data. This automated procedure for consensus assembly can lead to the correction of both base call errors and indels in the rough assembly.

As the consensus sequence is built, a map showing the location of all contigs in the assembly is developed. Figure 3 shows an example of an Assembled Consensus EST database entry (described in the next section). The section of the figure labeled "REX MAPFILE" shows a contig map generated by REX. ESTs which were used as left and right extenders are marked with <* or >* respectively. The top-level anchor is marked with ***. The percent identity between the rough assembly and the contig in the region of the MSP is indicated in the next column. Regions of a contig which are not contained in the MSP are indicated with dots. A dotted region may represent a piece of unspliced hnRNA (i.e., intron), a splice variant, a masked repetitive element, or simply poor, error-laden sequence. Following this are the query coordinates of the MSP, and in brackets the name of the EST clone partner, when available. When both the 3′ and 5′ ESTs of a clone pair are included in an assembly, the EST identifiers are marked with an asterisk.

Figure 3 also displays the assembled sequence which is identical to the Placentin gene (Genbank accession L34838). Translation of the assembled sequence for this example yields a sequence that is free of indels, despite the fact that all individual contigs have indel errors and some contigs appear to derive from unspliced hnRNA. In fact, there is an average of 6 indels per EST in the cluster. The TIGR assembler produces 5 THCs (Tentative Human Consensus sequence) for this gene. All but one (THC155875, one indel) have several indel errors. Another (THC207858) appears to be a chimeric assembly (composed of two distinct messages).

The example above is for illustration only and a full description and evaluation of the REX assembly and consensus base calling method will appear elsewhere.

ACE: Assembled Consensus ESTs

There are about 1.5 million ESTs available in public databases. Many questions concerning these data come to mind. How many of the approximately 100,000 human genes do these ESTs represent? Are all human genes "tagged" in existing EST databases? If not, how far away are EST databases from complete coverage? Which ESTs represent highly expressed messages? Which ESTs represent rare messages? What proportion of ESTs represent mis-spliced messages? How many splice variants are observed for an typi-

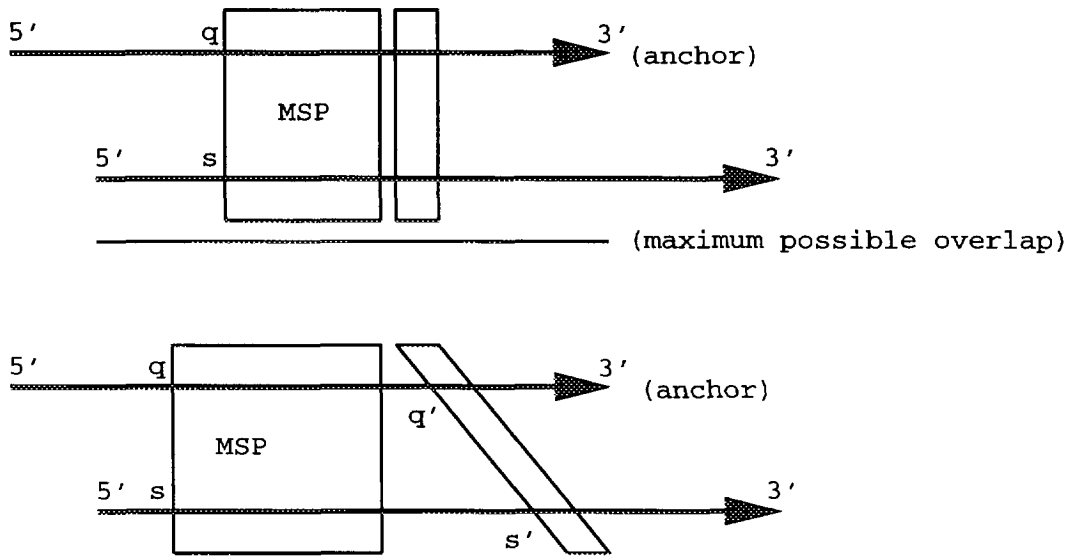


Figure 2: Identification of possible anomalous ESTs. Top: the HSPs between the anchor and extender contig do not cover enough of the possible overlap region. This contig may derive from hnRNA (heteronuclear, unspliced RNA) or alternatively spliced mRNA. Bottom: an HSP is inconsistent with the MSP. This contig probably derives from an alternatively spliced mRNA.

cal gene? One way to address these questions is to construct a fully assembled EST database.

In this section, our effort to apply REX systematically to a large collection of ESTs in order to represent every EST in at least one cluster is described. REX is well suited for creating an assembled EST database since it is highly amenable to non-interactive, automated batch processing of EST databases. By iteratively running REX with anchor sequences selected from a database of EST sequences, it is possible to create an assembled database, which is called ACE for Assembled Consensus ESTs.

ACE is constructed by iteratively selecting an anchor EST from a set of unassembled ESTs. After picking an anchor, the first step is to determine if the anchor is contiguous with an existing assembled consensus sequence. If so, the anchor sequence is removed from the pool of unassembled ESTs and is added to the cluster of ESTs that is associated with the matching consensus sequence. The consensus sequence is then marked for update so the assembly will be recalculated using the original anchor sequence in order to incorporate the new data.

If the anchor sequence is not contiguous with an existing consensus sequence, then REX is used to produce a cluster of contigs and an assembled consensus sequence representing the cluster. The new consensus sequence is then compared to the set of existing ACE consensus sequences. If an existing ACE sequence shares 90% sequence identity over 90% of the length of the new consensus sequence, the shorter of the two sequences is removed and the EST clusters are merged. All of the contig ESTs contained in either assembly are removed from the unassembled EST pool. If necessary, the retained cluster is marked for update and subsequent recalculation.

If the new consensus sequence is not contiguous with an existing ACE sequence, then it represents either a novel assembly or a splice variant of an existing assembly. In either case, the new assembly is added to the set of ACE sequences and the contig ESTs are removed from the unassembled EST pool. By not considering any EST that already exists in an ACE assembly as an anchor sequence, calculation of multiple assemblies representing the same gene is avoided. The iterative assembly process is illustrated in Figure 4.

Since an enormous amount of computation time is required to fully assemble a large EST database, it is desirable to both maximize the diversity of the assemblies and to preferentially create "interesting" assemblies first. One way to maximize the diversity of the assemblies is to initially ignore splice variants. REX identifies inconsistent ESTs that may represent splice variants relative to the current assembly. By using these ESTs as anchor sequences in subsequent REX runs, it would be possible to calculate several potential splice variants of a given gene. While this is desirable, it is more useful to first maximize the number of unique genes in ACE as opposed to initially calculating all possible variants for a smaller set of genes.

```
;ID ACE_EST1086973
;DT 11-Oct-1997 (CREATED)
;DT 11-Oct-1997     (UPDATED)
;DT 27-Jan-1998 (LAST ANNOTATION UPDATE)
;CC ASSEMBLED BY (rex)
;CC
;CC -!- LIBRARIES  (instances (freq)  library name) -!-
;CC  17 (0.708333)    Soares placenta Nb2HP
;CC  6 (0.250000)     Soares fetal liver spleen 1NFLS
;CC  1 (0.041667)     Soares total fetus Nb2HF8 9w
;CC
;CC -!- REX MAPFILE -!-
;CC locus                pid 1-------------------------------------------------------------763
;CC EST201890          95.00  ----------------------.....> 14-303 (EST201839*)
;CC EST262328          97.60  ----------------------.........> 14-300 (EST262329*)
;CC EST201933          92.60  ------------------------> 14-343 (EST201323*)
;CC EST1086973       ***100.0  --------------------------------> 1-431
;CC EST252669          95.20   ----------------------.........> 25-309 (EST252559*)
;CC EST244553          96.80   ----------------------..> 25-331 (EST244554*)
;CC EST362059          97.80    --------------------..> 32-300 (EST361244)
;CC EST125542          96.70    ------------------------> 30-358 (EST125468*)
;CC EST255024          96.90    --------------------------.........> 28-370 (EST254662*)
;CC EST233419          97.20    --------------------------.........> 27-372 (EST233579)
;CC EST331976          95.50    ----------------------------......> 40-411
;CC EST238995          93.80    --------------------------------...> 33-446 (EST238891*)
;CC EST497330          95.80     --------------------> 43-329 (EST475577)
;CC EST371834          96.40     ----------------------------> 47-432
;CC EST252559          97.60          <........--------------------------- 226-589 (EST252669*)
;CC EST201323          96.80            <..............-------------------- 349-619 (EST201933*)
;CC EST244554          92.60              <..........---------------------- 313-616 (EST244553*)
;CC EST254662          94.20              <.......--------------------------. 254-603 (EST255024*)
;CC EST238891          97.20                <..---------------------------- 233-616 (EST238995*)
;CC EST262329          96.90                 <.......----------------------- 298-619 (EST262328*)
;CC EST125468          93.30                    <.....----------------------- 313-618 (EST125542*)
;CC EST245284        >*96.90                         ----------------------> 310-626 (EST245285)
;CC EST253958        >*98.00                             ------------------------------> 364-761
;CC EST201839          97.90                              <------------------ 377-617 (EST201890*)
;CC
;CC -!- ZGI ANNOTATED COMPONENT ESTS -!-
;CC            Data not shown
;CC
;SQ SEQUENCE   763 nucleotides
ACE_EST1086973
ATTCGGCACTAGGGAAGGGACACACCAGCACAGTCTGGTAGGCTACAGCAGCAAGTCTCTAAAGAAAGGC
TGAGAACACCCAGAACAGGAGAGTTCAGGTCCAGGATGGCCAGCCTGTTCCGGTCCTATCTGCCAGCAAT
CTGGCTGCTGCTGAGCCAACTCCTTAGAGAAAGCCTAGCAGCAGAgCTGAGGGGATGTGGTCCCCGATTT
GGAAAACACTTGCTGTCATATTGCCCCATGCCTGAGAAGACATTCACCACCACCCCAGGAGGGTGGCTGC
TGGAATCTGGACGtCCCAAAGAAATGGTGTCAACCTnCCAACAACAAAGATGGACAAGCCTTAGGTACGA
CATCAGAATTCATTCCTAATTTGTCACCAGAGCTGAAGAAACCACTGTCTGAAGGGCAGCCATCATTGAA
GAAAATAATACTTTCCCGCAAAAAGAGAAGTGGACGTCACAGATTTGATCCATTCTGTTGTGAAGTAATT
TGTGACGATGGAACTTCAGTTAAATTATGTACATAGTAGAGTAATCATGGACTGGACATCTCATCCATTC
TCATATGTATTCTCAATGACAAATTCACTGATGCCCAATTAAATggATtgCTGTTTATTAGGAACATGAG
GAATCATTATTAGGTATTCACATGTTTTCACTTGGCTCnTGTACnTAATTATAGGTCTCTTCCCnAATTG
GGGGGGAGGTTAGGnTAGGnTTGGGGTTTAGGTTCCCGGTAGGTTGnTAGGnTCCAnTGGGGG
```

Figure 3: An ACE entry representing the insulin-like hormone Placentin (Swiss-Prot INL4_HUMAN, Genbank accession L34838) using dbEST entry EST1086973 as an anchor sequence. The ACE entry shows the assembly date, the library distribution of the contigs, a contig map, and the consensus sequence.

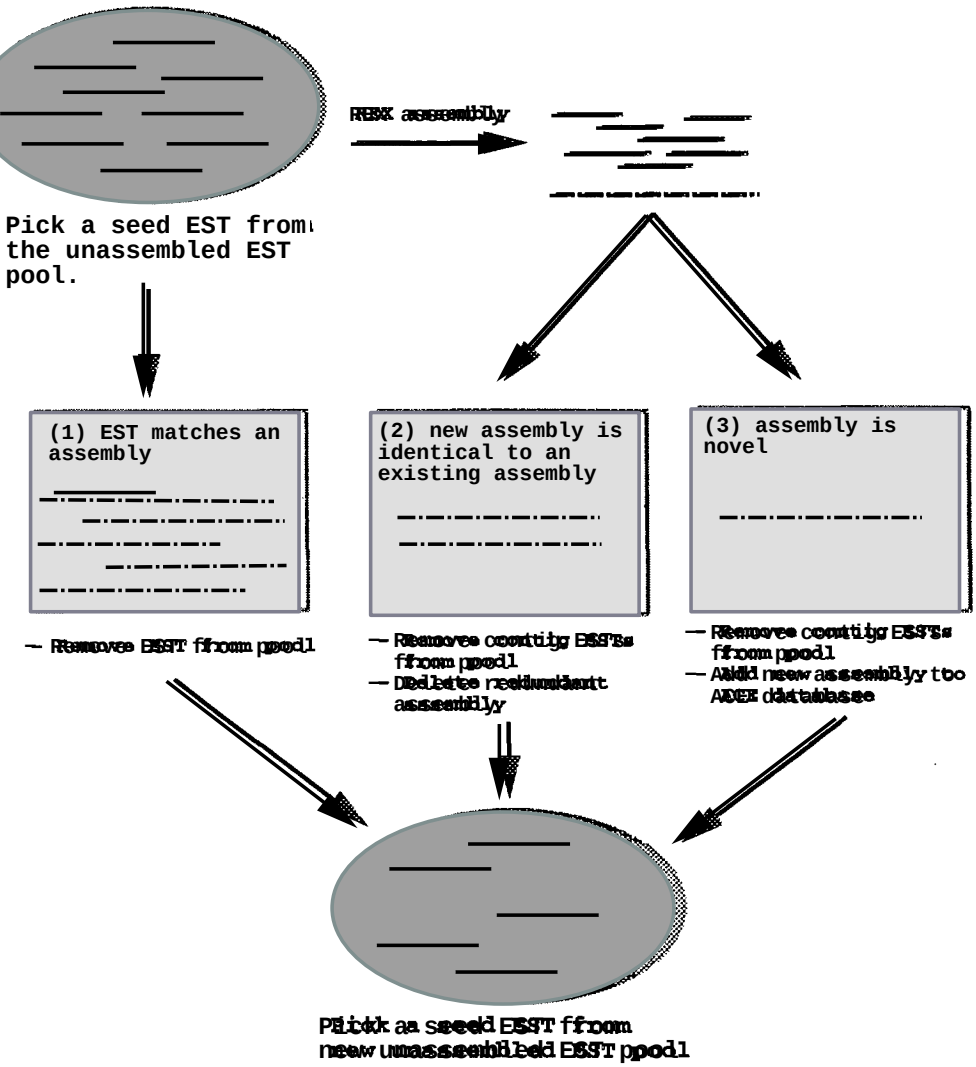


Figure 4: ACE assembly process. The pool of unassembled ESTs is represented by shaded ovals. After picking an anchor EST from the unassembled EST pool, (1) the anchor may be subsumed by an existing assembly, (2) the assembly generated by the anchor may be identical to an existing assembly, or (3) the assembly may be novel. In each of the three cases, the relevant ESTs are removed from the unassembled EST pool and the process is repeated.

Consequently, the splice variant information generated by REX is not explicitly used when initially constructing ACE.

More importantly, our aim was to enrich ACE with "interesting" assemblies that have a higher than random chance of representing a gene of interest. One way to enrich ACE is to select anchor ESTs that look "interesting". For example, if one were mining EST databases for novel G-protein coupled receptors (GPCRs), one could search the EST databases for all ESTs that match a known GPCR above a low stringency threshold. All of the potentially interesting ESTs based on the GPCR hits can then be used as REX anchor sequences. While some of the assemblies may represent genes that are not GPCRs (i.e., false positives), the calculated assemblies are enriched for GPCR homologs.

ACE is maintained using a relational database (SYBASE). The type of information available for every assembled sequence includes the consensus sequence, the anchor sequence, the assembly method used, the organism, and the date of assembly. In addition, all contig ESTs are represented in a cluster along with information indicating the libraries from which the ESTs were derived and the orientation and position of the ESTs in the consensus sequence. Finally, site specific annotations associated with component ESTs are also represented in the data model. A flat file representation of an ACE assembled sequence is shown in Figure 3.

By assembling individual ESTs into consensus sequences, it is possible to generate an improved, less redundant view of the EST data. Specifically, the average length of an assembled sequence is increased relative to an EST; this may yield more interpretable sequence for the purpose of database mining. The sequences may also have fewer errors due to the consensus assembly method used by REX. Finally, there are fewer sequences in an assembled EST database to search relative to a database consisting of all component ESTs. If a gene is sufficiently represented by EST data, full-length virtual transcripts can be created by this process.

The relational database representation of the ACE data allows one to make complex queries of the data. By monitoring the libraries from which the component ESTs in each assembly were derived, it is possible to calculate tissue distribution information (i.e., an electronic Northern). It is also possible to identify low information content ESTs by finding those that participate in many ACE assemblies. The identification of low information content ESTs was helpful in refining the repetitive element filter mentioned earlier in this paper. It is also possible to identify rarely expressed messages by looking for assemblies that have few ESTs in their clusters.

Another useful feature of ACE is the ability to measure the novelty of new EST data. After creating a assembly from a new EST, the resulting consensus sequence can be new, a splice variant, or identical to an existing assembly. By monitoring the ratio of novel assemblies to redundant ones, it is possible to determine how effectively novel data is generated by a given sequencing strategy.

Discussion

The sequence assembly problem has long been known to be intractable due to the large space that must be searched to produce optimal solutions. However, faced with a large body of valuable EST data one needs to address the assembly problem. Efforts such as those of TIGR (Sutton et al. 1995) have been valuable in supplying the scientific community with a second generation database of assemblies. This paper has described the REX algorithm, a simple greedy assembly algorithm which is an alternative to tools such as the TIGR assembler and others which precompute a table of all pairwise overlaps. Since no preprocessing is necessary before generating an assembly, REX is particularly amenable to databases that are changing rapidly.

It is a greedy algorithm because as it "walks" through a database of ESTs creating a virtual cDNA, it chooses without lookahead a single EST to perform the extension at each step of recursion. When faced with an anchor sequence which may be unspliced or a variant with respect to some other EST, it tries to create an assembly which represents the same variant as the anchor EST. Thus for a given gene, different anchors may produce assemblies of varying length and content. Also note that there is no guarantee that the consensus sequence produced by REX actually occurs; for example, the sequence produced by REX may merge different parts of two splice variants into a single assembled sequence.

The ACE database is an example of the systematic application of the REX algorithm to large EST databases, analogous to the THC (Tentative Human Consensus) efforts of the TIGR group (Sutton et al. 1995). We have described a process for creating an assembled EST database, and outlined a procedure for maintaining this database as new data is generated. Once created, the assembled EST database can be updated and the full database never has to be recomputed.

As EST databases approach complete coverage of all expressed human genes, the ability to characterize these data becomes critical. By querying a relational database of fully assembled ESTs, it is possible to identify high- and low-information content ESTs, generate full length virtual transcripts, identify ESTs that are poorly represented in EST databases, and generate tissue distribution profiles for well represented genes. Since REX identifies ESTs that are potential splice variants, there is a mechanism for systematically exploring all splice variants of an expressed gene. ACE also allows one to assess the novelty of new data. Fi-

nally, by observing changes in the number of assemblies, it is possible to address the question of how many expressed genes are represented in EST databases.

Acknowledgments

The authors would like to thank Patrick O'Hara for providing a stimulating research environment, Domenick Venezia for keeping the computers alive and databases current, and Jim Holloway for preliminary work on the repetitive element sequence filter. Finally, the authors thank Scott Presnell, Paul Sheppard, and Terry Farrah for useful comments and feedback during the development of REX and ACE.

References

Altschul, S., Gish, W., Miller, W., Myers, E. and Lipman, D. 1990. Basic local alignment search tool. *J. Mol. Biol.*, 215, 403-410.

Bailleul, B., Akerblom, I., and Strosberg, A. D. 1997. The leptin receptor promoter controls expression of a distinct second protein. *Nucleic Acids Research*, 25, 14:2752-2758.

Boguski, M. S., Lowe, T. M., and Tolstoshev, C. M. 1993. dbEST–database for "Expressed Sequence Tags". *Nature Genetics*, 4, 332-333.

Boguski, M. and Schuler, G. 1995. ESTablishing a Human Transcript Map. *Nature Genetics*, 10, 369-371.

Gill, R., Hodgman, T., Littler, C., Oxer, M., Montgomery, D., Taylor, S. and Sanseau, P. 1997. A new dynamic tool to perform assembly of Expressed Sequence Tags. *CABIOS*, 13, 4:453-457.

Lin, J., Poole, J., and Linzer, D. I. 1997. Three new members of the mouse prolactin/growth hormone family are homologous to proteins expressed in the rat. *Endocrinology*, 138, 12:5541-5549.

South African National Bioinformatics Institute 1997. STACK: Sequence Tag Alignment and Consensus Knowledgebase. www.sanbi.ac.za/stack

Sutton, G., White, O., Adams. M. and Kerlavage, A. 1995. TIGR Assembler: A new tool for assembling large shotgun sequencing projects. *Genome Science & Technology*, 1, 1:9-19.

Wolfsberg, T. and Landsman, D. 1997. A comparison of expressed sequence tags (ESTs) to human genomic sequences. *Nucleic Acids Research*, 25, 8:1626-1632.

Wu, Z., Wu, J., Jacinto, E., and Karin, M. 1997. Molecular cloning and characterization of human JNKK2, a novel Jun NH2-terminal kinase-specific kinase. *Mol. Cell. Biol.*, 17, 12:7407-7416.

Yamada, Y., Nezu, J., Shimane, M., and Hirata, Y. 1997. Molecular cloning of a novel vascular endothelial growth factor, VEGF-D. *Genomics*, 42, 3:483-488.