Published in final edited form as:
J Proteome Res. 2014 January 3; 13(1): 21–28. doi:10.1021/pr400294c.

Proteogenomic database construction driven from large scale RNA-seq data

Sunghee Woo[†], **Seong Won Cha**[†], **Gennifer Merrihew**[‡], **Yupeng He**[¶], **Natalie Castellana**[§], **Clark Guest**[†], **Michael MacCoss**[‡], and **Vineet Bafna**[*,§]

[†]Department of Electrical and Computing Engineering, University of California, San Diego

[‡]University of Washington, Department of Genome Sciences, Seattle, WA, USA

[¶]Department of Bioinformatics and Systems Biology, University of California, San Diego

[§]Department of Computer Science, University of California, San Diego

[‖] University of California, San Diego, United States

Abstract

The advent of inexpensive RNA-Seq technologies and other deep sequencing technologies for RNA has the promise to radically improve genomic annotation, providing information on transcribed regions and splicing events in a variety of cellular conditions. Using MS based proteogenomics, many of these events can be confirmed directly at the protein level. However, the integration of large amounts of redundant RNA-seq data and mass spectrometry data poses a challenging problem. Our manuscript addresses this by construction of a compact database that contains all useful information expressed in RNA-seq reads. Applying our method to cumulative *C. elegans* data reduced 496.2GB of aligned RNA-seq SAM files to 410MB of splice graph database written in FASTA format. This corresponds to 1000× compression of data size, without loss of sensitivity. We performed a proteogenomics study using the custom dataset, using a completely automated pipeline and identified a total of 4044 novel events, including 215 novel genes, 808 novel exons, 12 alternative splicings, 618 gene-boundary corrections, 245 exon-boundary changes, 938 frame-shifts, 1166 reverse-strands, and 42 translated UTR. Our results highlight the usefulness of transcript+proteomic integration for improved genome annotations.

INTRODUCTION

With the advent of inexpensive DNA sequencing technologies, researchers finally have the opportunity to sequence thousands of individuals in a population. This presents the scenario that every individual will be sequenced, perhaps multiple times in their lifetimes, providing a comprehensive and unbiased look at genomic variability in the population. A few large scale studies have explored this genomic variability,[1,2] and have shown that the genomes are surprisingly plastic, diverging not only with single nucleotide variations, but include large

[*]**Corresponding Author** Tel: +1 858 822 4978. Fax: +1 858 534 7029. vbafna@cs.ucsd.edu.

Supporting Information
Supporting Information Available: This material is available free of charge via the Internet at http://pubs.acs.org.

structural changes involving deletions, inversions, translocations, and duplications of large portions of the genome. It is only to be expected that these genomic changes also modify the structure, splicing patterns, and the primary sequence of the expressed transcripts and proteins.

Historically, gene finding has been solely the province of the genomics community. In addition to *de novo* signals for coding regions and splicing, gene finding tools also make use of transcript information to identify genic regions, splicing, and other information. The availability of RNA-Seq and other deep sequencing technologies for RNA has the promise to radically improve genomic annotation. ENCODE and other, similar projects have made effective use of RNA-Seq, ChIP-seq, and other technologies to improve the functional annotation of the genome.[3]

Nevertheless, challenges remain, even with simple gene finding. Although RNA-seq provides a deep sampling of expressed genes within the sample, not all genes are expressed at one time. Therefore, RNA-seq data generated from multiple experiments must be used in a cumulative manner. The transcribed portion of the genome appears to greatly exceed the translated portion, and everything that is transcribed may not be translated. Transcriptomes do not provide information on the reading frame, and large amounts of pre-spliced and un-spliced RNA mask true splicing events.

The emerging field of proteogenomics attempts to remedy this by using proteomic information derived using tandem mass-spectrometry to augment the transcript information. For example, we can search MS spectra against a translation of RNA-seq reads, but this is both inefficient and redundant. Typical RNA-seq database sizes match the size of the genome, while only sampling a small fraction (~ 3%) of it. An improvement is to assemble RNA-seq fragments into longer transcripts, and search these reduced databases.[4,5] However, this approach also has many shortcomings. First, information is lost during the assembly, and indeed a wrong call might be made among competing splicing events. A peptide might match multiple isoforms derived from the same set of reads. Information on mutations is often discarded during assembly.

Further, the best sensitivity is obtained by accumulating, and searching RNA data across multiple conditions and cell-types. However, it is technically difficult to assemble multiple RNA-seq data-sets given the huge numbers of experiments. As an extreme example from humans a single project (The Cancer Genome Atlas or TCGA) project lists over 240Tb of RNA-Seq data across multiple cancer sub-types.[6] It is not clear that there is an effective way to search all of these data-sets, even when limited to a specific sub-type. Previous studies such as Wang *et al*. (2012),[7] focused on creating customized proteomic database by reducing the search space using RNA-seq in order to increase the sensitivity of the peptide identification proteomic database. Li *et al*. (2011)[8] focused on encoding SNPs (Single Nucleotide Polymorphisms) to the proteomic Database.

Our study has different goals, focused on finding novel gene events. For this reason, we explicitly search both RNA derived, and 6-frame translated data-sets. Therefore, we focus on maximizing the sensitivity of the database itself with respect to gene features such as

splicing and translated regions. In addition, rather than using matched RNA and proteomic samples, we work on *aggregated RNA* datasets from multiple experiments to maximize sensitivity, and remove the constraint of proteomic and RNA data being from the same sample. To reduce the search space, we construct a non-redundant compact database that contains useful splicing information expressed in RNA-seq reads, along with enough information that any MS search tool can identify peptides. We have developed a tool to construct a splice graph database using RNA-seq fragment mappings. The database encodes a graph G where genomic intervals (exonic regions) correspond to nodes, and edges correspond to pairs of exons that are putatively spliced together. RNA-seq read mappings that split across splice-junctions are used to determine edges. The huge compression in database size comes from the tremendous redundancy of transcript generation and mapping. Mutations, including small insertions and deletions are repeatedly sampled both within a data-set, and also across many data-sets. Unlike transcript assembly, the splice-graph does not have to select between specific splicing paths. Thus, there is no loss of sensitivity, even with the large compression. We have previously used a splice-graph encoding of cDNA sequences (typically from EST projects) for proteogenomic studies.[9,10] Here we modified the tool to deal with very large RNA-seq data-sets and release it for general use. As shown in the results, we can compress large (> 490Gb) of RNA-seq mapping data to a compact database of 0<.4Gb.

While some MS2 identification software can search splice-graphs directly,[9–11] most tools require a FASTA formatted sequence database. To generate a universal database, we also present a tool to convert splice graph data structure into a multi-FASTA formatted file. The naive approach would enumerate all paths in the graph, leading to a large expansion in size. Instead, we exploit the short length of typical 'bottom-up' peptides. Using an adjustable parameter L as the maximum length of a peptide, our tool generates a highly compressed FASTA file that encodes all splice-graph peptides of length $\leq L$. Applying this method reduced 496.2GB of aligned RNA-seq SAM files to a 410MB splice graph database written in FASTA format. This corresponds to 1000× compression of data size, without loss of sensitivity.

We performed a proteogenomics study using our pipeline and identified a total of 4044 novel events (as compared to *C. elegans* gene set version WBcel215.68[12]). The identified events included 215 novel genes, 808 novel exons, 12 alternative splicings, 618 gene-boundary corrections, 245 exon-boundary changes, 938 frame-shifts, 1166 reverse-strands, and 42 translated UTR. Our results highlight the usefulness of transcript+proteomic integration for improved genome annotations.

METHOD

The pipeline has two major parts: splice-graph construction from mapped reads, and splice-graph to FASTA conversion.

Input preparation

RNA-seq data used in our study was generated by the Waterston lab. RNA-seq reads were mapped using the method described in previous studies[13,14] as part of the modEN-CODE

project. RNA-seq read alignments are in SAM format,[15] a column based encoding of the alignment of each read. The alignment itself is stored in a compact CIGAR string of the SAM file. We parse the CIGAR string to obtain the split information in spliced reads. Resulting coordinates of the mappings are converted into GFF, a simple interval-based flexible format for representing genomic intervals. Calculation of the split mapped coordinates from the CIGAR string is described in the Supplementary Material section. While RNA-Seq reads can span multiple splice junctions, we did not find such instances in our data set which had 76bp sequences. However, the software automatically, through its parse of the CIGAR string, identifies, and creates (multiple) splice junctions per read. The functionality of encoding genomic variants such as insertions, deletions, and mutations, is not applied in this study due to the lack of known genomic variant information for this data-set, but it was used in other data not presented here.

Filtering RNA-seq reads

We employed a number of filtering steps to reduce the size of the database. To start, note that the final proteogenomic study involves searching MS/MS spectra against three different databases. These are a database of known proteins, a 6-frame translation of the entire genome, and the splice graph database. In the past, a search of the 6-frame database has been eschewed when transcript data is available. However, typical RNA-seq databases are often the size of the genome (or larger), and when multiple RNA-seq databases are being searched, the burden of searching the 6-frame translation becomes less dominant. Moreover, *all* non-spliced, non-mutated, peptides can be identified in the search. Therefore, as a first filtering step, we discarded all RNA-seq fragments that are not *split-reads*, i.e., they map to the genome without splicing. This simple step removed 71.79% of the total RNA-seq reads.

In a second filtering step, we considered mappings of the split-reads to the genome and compress all the reads sharing the same intron boundary into a single read. As in Figure S.1, if a pair of reads share the same intron coordinates, we merged them into a single read preserving the boundaries of the intron, and extend both ends of the read. In this process, we maintained hash table using intron coordinates as a key value which represents a unique splice junction (intron coordinate pair). Entries of the hash table contain information of expanding exon boundaries on each side, RNA-seq read counts, and original RNA-seq file name. Once all files were processed, we filtered out all putative introns that are not covered by at least c reads, where c is a user-defined parameter with default value $c = 2$. This second filtering stage removed 98.16% of the reads that survived from the first filtering step. After applying the above two steps of filtering, a total of 4,669,116,388 RNA-seq reads were reduced to 517,326 merged RNA-seq components, and the average exon length on each side of the component was 83.42bp. Note that the goal is not to cover the entire exon, but to cover split peptides of maximum length L. In our study, we used 90bp (30 amino-acids) for the value of parameter L.

To further reduce the computational burden, we partitioned the merged data into multiple files, based on mapping coordinates. The construction ensures that splice junctions do not exist between multiple files, and therefore the true splice graph is simply a concatenation of

the splice-graphs from each file. The splice-graph construction was done in parallel for each file.

Constructing the splice-graph

In the splice graph data structure, nodes represent exons, and edges represent splice junctions. The construction is schematically illustrated in Figure 1. Starting with the empty graph, the splice-graph is augmented/updated read by read. (here, a read represents a single merged component of RNA-seq reads which is the output of the previously described filtering stage.)

See Figure 1 for an example. Given RNA-seq read r_1, node s_1 is split into nodes u_1 and u_2, and node u_3 is added. Next, we assign edges for each spliced-read. In Figure 1(c), edge e_4 is added to the current set (e_1-e_3). Finally, we revisit each pair of contiguous nodes, where contiguous means that there is no coordinate gap between the previous and next node. In Figure 1, u_1 and u_2 are contiguous, while u_5 and u_6 are not contiguous since there exists a gap in between. The contiguous nodes are merged if there is no assigned edge between the corresponding pair; otherwise, they are connected by an additional edge. For example in Figure 1(d), u_1 and u_2 are merged since there is no edge between. On the other hand, u_2 and u_3 cannot be merged due to the existence of e_1, and the additional edge e_5 is assigned.

Converting splice graph structure to a FASTA file format—While the splice-graph database is a compact encoding of splice patterns, it cannot be searched directly by standard MS/MS search tools. To overcome this limitation, we developed a tool that generates a FASTA formatted database from the splice-graph.

The generated FASTA database must have certain properties that relate it to the splice-graph database. Following Edwards and Lippert,[16] we say that a FASTA database F is *L-Complete* w.r.t a splice-graph database G if every length L sequence in G is a substring in F. In addition, F is *correct* w.r.t G if every string in F is also a substring in G. Given a splice-graph G, and a user-defined parameter L, our objective is to generate a minimum size (number of amino acids in database) FASTA database F that is correct and L-complete w.r.t G. A naive approach for splice graph to FASTA conversion is to retrieve all possible paths within G and generate a new FASTA sequence. Such a database is complete (for all L), and correct, but will be greatly increased in size, growing exponentially in the average node-degree of G. Similar to Edwards and Lippert,[16] we also describe a novel method which finds a greedy but effective solution of this problem. However, unlike Edwards and Lippert,[16] our method uses a genomic-coordinate based data structure(represented in base pairs) rather than minimizing the amino acid sequence overlap. We claim that for proteogenomic analysis, the coordinate based approach is more appropriate since it can easily reconstruct the original genomic coordinate of the identified peptide.

FASTA conversion strategies

We used three rules to eliminate shared sub-paths.

1. For a pair of paths, xz and yz with a shared string z, we generate two FASTA strings: xz, and $y \cdot \text{pref}_L(z)$, where $\text{pref}_L(z)$ denotes a length $L - 1$ prefix of string z.

2. For a pair of paths, *xz* and *xy* with a shared prefix *x*, we generate two FASTA strings: *xz*, and $\text{suff}_L(x) \cdot y$, where $\text{suff}_L(x)$ denotes a length $L - 1$ suffix of string *x*.

3. For paths *xy* and *yz*, which have a prefix-suffix match with $y \geq L$, generate the FASTA string: *xyz*.

Rules 1 and 2 can be implemented in an enumerating procedure during a depth first search (DFS) traversal of the splice graph. Recall that in a standard DFS search, a node is marked the first time it is visited. Thus if a previously-visited node *v* is revisited, we keep only the length $L - 1$ path from outgoing edges to *v*. Likewise if a traversal touches a node with multiple outgoing edges, we need to only maintain a length $L - 1$ suffix to attach to each of the outgoing paths. (See Figure 2) Implementation of rule 1 and 2 is described in Algorithm 1, and the following rule 3 can be applied separately after the completion of this algorithm.

Rule 3 allows us to combine pairs of sequences that share a prefix and suffix string. First, we identify *overlap-node-pairs* as pairs of *merge* nodes (out degree > 1) and *split* nodes (in-degree > 1) with length l ($L \leq l < 2L$) sequence between the two. If $l < L$ (seq1 and seq2 in Figure S. 2(a)), the generated sequences cannot share an identical prefix and suffix. If $l \geq 2L$, the prefix and suffix of generated sequences will not overlap (Figure S. 2(b)).

For implementation of rule 3, we used a hashing technique to rapidly identify overlap-node-pairs. Traversing the graph in a depth first fashion, we store all the split nodes present in a candidate list. For each split node *u*, we consider the sequence of nodes encompassing the length *L* prefix of *u*, and hash the prefix string using the first 3 nodes as key (Figure S. 3(b)), so that each key contains the list of the paths such that prefix of the paths is the same as the corresponding key. Every time a merge node is encountered in the DFS, we traverse the subsequent path, querying the hash table continuously using 3 node triplets. For example in Figure S. 3(c), key2, key3, and key4 are used to query the hash table. When a match is found (e.g., between key4 and key1), the hash table returns a list of sequences that corresponding paths starting with the appropriate key. (e.g., ('TCG'+'CG'+'GG'+'AAC'+'CCTA'+ 'AATATG')). We search each sequence within the returned sequences, using the remaining suffix of the queried sequence. In our example, the remaining sequence is 'A' which appears right after the key4. We merge the matched sequence with queried sequence, and translate it into three different frames. Finally, we output the 3-frame translated sequences to a FASTA file.

Heuristic constraints to prevent exponential growth—Growth of the final FASTA database is dependant on the complexity of the splice graph. Splice junctions expressed in RNA-seq reads are expressed as edges in the graph. Multiple edges assigned within a small region(less than *L*) will increase the complexity of the splice graph structure. Therefore, the final FASTA file will grow exponentially in the case where many splice junctions are found within a small region. To prevent exponential growth in very complex regions, we added an additional constraint to our conversion strategy. Based on the RefSeq known protein database, we set a proper length parameter *W* as the minimum distance between adjacent splice-junctions. In our implementation, if two splice junctions appear within *W* bp of each other, the FASTA sequence generation selects each splicing independently, but not in combination. We used $W = 20$bp in this study (See Figure S.4 for the description of *W*

parameter). Note that the average length of exons in RefSeq C.elegans database was 207.62bp (s.d. 262bp). Only 1.01% of known exons were shorter than 20bp. The proofs of correctness and completeness in applying all methods above are illustrated in Supplementary material.

Datasets and experimental procedure—RNA-seq data was generated by the Waterston lab as part of the modENCODE project. The dataset used was 111 experiments from multiple Caenorhabditis species and developmental stages. RNA-seq reads were mapped as described in the studies.[13,14,17] Detailed RNA-seq methods are illustrated in Supplementary material.

For the mass spectrometry data, eleven developmental stages of C. elegans were analyzed - N2 embryo, N2 L1, N2 L2, N2 L3, N2 L4, N2 YA, N2 dauer, spe-9 L4, spe-9 YA, spe-9 adult and him- 8. Each developmental stage was grown on agar plates at 20 °C, seeded with the NA22 strain of E. coli.,[18] sucrose floated, lysed in the presence of protease inhibitors (Roche) and centrifuged to separate insoluble and soluble fractions. A 200μg soluble lysate of each developmental stage was reduced with DTT(Sigma) and separated into 15 molecular weight fractions ranging from 3.5 to 500 kDa using the GelFree 8100TM fractionation system (Protein Discovery/Expedeon).[19] Each fraction was alkylated with IAA (Sigma) and trypsin (Promega) digested. SDS was removed with SDS removal columns (Pierce) and salts were removed with MCX columns (Waters). The peptides from each fraction were analyzed using a 35 cm fused silica 75μg column and a 4 cm fused silica Kasil1 (PQ Corporation) frit trap loaded with Jupiter C12 reverse phase resin (Phenomenex) with a 120-minute LC-MS/MS run on a Thermo LTQ-Orbitrap Velos mass spectrometer coupled with an Eksigent nanoLC 2D. A biological and analytical replicate was performed for each sample. Using the constructed splice graph database, we launched a proteogenomics search of C.elegans MS/MS spectra dataset. C.elegans MS/MS spectra is a total of 81 GB in size, consisting of 11,123,595 spectra. The spectra dataset was produced by the MacCoss lab, and comparison to Merrihew *et al*. (2008)[17] is illustrated in Supplementary material. We used MSGFDB (version 20120106)[20] for the database using the following parameters: 30ppm for parent mass tolerance, semi-tryptic search, Carbamidomethylation of C as fixed modification, and Oxidation of M as optional modification. For each spectrum, we selected PSMs with the lowest SpecProb reported by MSGFDB across all database search results (known proteins, 6-frame, and splice-graph-fasta). The reversed decoy database was also searched for all databases to apply the target-decoy approach. The database search resulted in 65,874 peptides better than 1% spectral level FDR cut-off. Among identified peptides, 52,292 corresponded to known peptides, but 13,582 peptides were novel. The novel PSMs were mapped back to their genomic coordinates using automated scripts. The 13,582 novel peptides mapped to 15,205 different genomic locations, giving on average of 1.12 locations per peptide. Among 15,205 locations, 3,484 were identified from the splice graph database, and 11,721 were from the 6-frame database.

Using previously developed tools,[9,10] identified peptides were grouped together into a single *event* in a pairwise fashion if they were located within $\leq$1000bp apart. The novel events were called automatically along with an event probability. We filtered out low quality results by setting the event probability cut-off as 0.998. For further validation, the novel

events were plotted using the UCSC genome browser[21] and verified using comparative genomics (protein level BLAST[22]). A second part of the software not used here, tracks how the novel findings were supported by specific RNA-seq data-sets, allowing for a more accurate correlation between protein and RNA evidence. In future work, we plan to apply this pipeline to compare MS and RNA data acquired on identical biological samples. Our software for splice graph construction and FASTA conversion is available at CCMS web page(http://proteomics.ucsd.edu/Software.html).

RESULTS

A splice graph was created from 496.2GB of aligned RNA-seq SAM files to 410MB of a splice graph database written in FASTA format. Overall statistics of our splice graph data structure is illustrated in Table S.1. The 6-frame translation database was created from the reference genome and also written in 102MB of FASTA formatted data.

Data compression

Figure 3(a) shows the overall increase in database size as a function of accumulating RNA-seq data. On the *x*-axis, the number of data-sets are progressively incremented up to 149 (496GB). The *y*-axis describes (on a log-scale) the growth of the corresponding splice graph and FASTA sequence database.

The 496.2GB RNA-seq data was compressed into a 410MB FASTA database, a 1000× compresion in terms of file size. Most of the gains are due to the filtering of spliced reads. Within the filtering stage, 71.79% reduction was achieved from filtering split mapped reads, and the remainder was from merging identical splice junctions and discarding ambiguously mapped reads. Since most of the size reduction was achieved in the filtering stage, this indicates the strong advantage of aligning RNA-seq reads before database construction, unlike other methods[16] where no coordinate information is used in database creation. Additionally, we observed that the rate of growth of the splice graph decreases after using 45 data-sets due to a saturation in the splicing information. Note that our design choice of filtering out the non-spliced reads works because we also search a 6-frame translation, which is 102MB in size. Thus, the 6-frame translation acts as a compressed version of unspliced RNA reads, which in combination with the splice graph reduces the 496.2GB file to (410 + 102)MB total. Moreover, due to the large variation in transcript abundance, we observe that even the large set of RNA data includes only 91% of known splice junctions, and we expect a similar ratio for known exons. Therefore, the addition of 6-frame translation also improves the sensitivity of the search by capturing all possible non-spliced translations.

The total computation time required for the database creation was 12 CPU-hours for filtering, 2.5 CPU-hours for graph construction, and 300 CPU-seconds for FASTA conversion. This database creation computation was performed on a Desktop PC with Intel Core i7 2.67GHz processor and 9.0 GB of RAM.

Database validation

To validate the splicing information, we compared our Splice Graph database with RefSeq[23] Accession NC_003279. The ideal Splice Graph database should cover all known splicing

junctions. Define the *RefSeq-splice-coverage (RSC)* as the fraction of all RefSeq splicing events covered by the SpliceGraph. We observed that the RSC value saturated after about 45 data-sets (Figure 3(b)), with most (though not all) RefSeq splicings incorporated.

Additional growth in the Splice Graph was due to the incorporation of novel splicings in the RNASeq data, but not in RefSeq. Figure 3(c) plots the number of novel splice junctions in the Splice Graph. Comparing the growths in Figure 3(b) and Figure 3(c), we observed a similar growth curve, with the observed rates being 1.2 × (75.0% to 91.3%) in coverage and 3.37 × (123,670 to 416,176) in number of novel splice junctions. The tremendous growth in novel splicing events, which might not be translated, highlights the ambiguity in locating gene events using RNA data alone, and underscores the importance of protein level validation via proteogenomics. Our proteogenomic search identified 2,126 novel spliced peptide locations from the total 416,176 novel putative splicings encoded in our splice graph database.

Proteogenomics discoveries

The compacted FASTA representation of splicings was used in conjunction with a known protein database in a proteogenomic analysis of a bottom up LC-MS/MS LTQ-Orbitrap data-set generated by the MacCoss lab. We used the existing proteogenomic pipeline, ENOsi[9,10], which automatically searches all of the spectra against the custom databases, accumulates all results, employs FDR calculations to identify Peptide Spectra Matches (PSMs), clusters novel peptides, and calls events automatically. *Event Probability Score* for each novel event was calculated as, $1-\left(\prod_{i\in S}(1-\frac{(1-FDR)}{LocationCount})\right)$ where *S* is the set of peptides assigned in current event, *LocationCount* is the number of genomic locations of the identified peptide, and *FDR* is the calculated FDR value of the corresponding PSM. We only reported novel events with *Event Probability Score* larger than 0.998. The proteogenomic search of 11,123,595 spectra was performed using a cluster server with 125 cores in parallel. 6-frame database search was done in 34.96 wall time hours, splice-graph-fasta database search took 15.31 wall time hours, and known protein database search took 6.54 wall time hours.

We note that the splice-graph FASTA has extensive header information that describes the mapped coordinates of reads and how they are split. The sequence part of the splice graph is 114MB in size, and it still contains some redundant sequence that can be efficiently handled by MSGFDB,[20] which indexes the database using suffix-tree techniques. Therefore, the search time of the splice graph FASTA is less than the six-frame translation.

The search revealed 4,044 events, as shown in Table 1. Figure 4 shows a few examples of the novel findings taken from our result which were further plotted using UCSC genome browser.[21] Red blocks represent identified peptides, and sky-blue blocks represent split mapped RNA-seq reads used in creating splice graph database.

We note that each event must contain at least one uniquely located peptide sequence match (PSM). However, a peptide group contains multiple peptides, a single group can represent multiple events. For example, there could be a group of peptides that extend the end of a

gene by jumping past the stop codon, and adding a new terminal exon. In this case, the peptides support both 'alternative splicing' and 'novel exon'. In another way to parse the solution, we identified 5463 unique novel peptide locations, and 3979 novel clusters.

Novel genes—We identified 215 novel gene events Table 1 where a collection of novel peptides were located ≥3Kbp from any annotated gene, again underscoring the impact of proteomic data on the discovery of new genes. Moreover, even with the extensive RNA information, we identify new genes from the 6-frame search as well. An example is shown in Figure 4(a), with 2 peptides, 'R.SRKSLPRTSQSPSSNFSGFY.V' and 'R.CYRYIIVSDIEKAFHQVRLQKAFR.N', both from 6-frame database search. We looked for comparative evidence using Blastx.[22] A query sequence was extracted from the DNA region 'chrV:19272600-19274335' and searched against the nr protein database.[24] The top blastx hit was 'hypothetical protein CRE-09558 [Caenorhabditis remanei]' with e-value 0.0. As shown in Figure S.5, peptide 'R.CYRYIIVSDIEKAFHQVRLQKAFR.N' was aligned with the protein sequence of Caenorhabditis remanei indicating that a similar CDS region exists in Caenorhabditis remanei which is a positive evidence of protein translation. Furthermore, we also found a supporting evidence from a predicted protein sequence generated from the GeneFinder[17] in the same region containing the peptide sequence 'R.CYRYIIVSDIEKAFHQVRLQKAFR.N'.

Gene corrections—The majority of the events in Table 1 are corrections to existing gene structures, including novel exons, extensions to UTRs, alternative splicing, frame correction, and even reverse strand events. We identified 12 alternative splicing events, with junctions that differed from RefSeq. Figure 4(b) shows 2 novel spliced peptides, 'T.LNVNGQE:IVYSMENEK.L', and 'R.EIKK:QHTSFQVSGPKEEIVYSMENEK.L' in their genomic context. The notation ':' indicates where the splice junctions are located. In peptide 'T.LNVNGQE:IVYSMENEK.L', the splice junction spans the amino acid 'I'. The peptides are well represented on either side, and located uniquely in the genome. Splice junction of peptide 'T.LNVNGQE:IVYSMENEK.L' was supported by 13 split mapped RNA-seq reads, and peptide 'R.EIKK:QHTSFQVSGPKEEIVYSMENEK.L' was supported by 40 reads. The peptides identify a novel splicing in the gene vit-5, part of a 5-member family of vitellogenin genes involved in maternal yolk production.[25]

We identified 938 'frame-shift events', where peptides match to known genes but in a different frame. In Figure 4(c), we identified the peptide 'TIVFTVPLSQCQMVSPMISK.E' (in the gene eef-2), which matches in a different frame. Two neighboring peptides, 'R.FIEPIEDIPSGNIAGLVGVDQYL:S.R', and 'G.HVFEESQVTGTPMFVV:R.L' were identified with 1 bp deletion, that allow for the frame-shift to occur. This region has complex RNA-seq mapping containing many small deletions, implying DNA assembly error, or a high degree of polymorphism in the region.

We measured the distribution of the novel peptides across different developmental stages. Figure S.6 shows the spectral counts of novel peptide spectra related to translated UTR events across different developmental stages. There is a small bias toward early developmental stages relating to translated UTR events. The translated UTR events suggest

new transcription start sites (and alternative regulation) of genes in early developmental ('N2 L1') stage.

To compare peptide versus RNA abundance, we computed a scatter-plot (Supplemental Figure S.7) of RNA-seq read counts mapped to a known gene (x-axis) versus the spectral count of peptides (y) falling within the region. The correlation between two values was calculated as 0.31 which implies that only a weak correlation is observed. However, since we are looking at the statistics of the accumulative data collected from various studies, we may need more detailed information on time specificity and sample consistency in order to study this correlation.

CONCLUSIONS

Our manuscript makes two points. First, cumulative mass spectrometry information (acquired in multiple studies) is a useful data resource for improving genome annotation, and should be applied as a standard part of continuing annotation efforts. Second, incorporating RNA-seq toward genome-annotation remains non-trivial due to high redundancy, large data sizes, but also, the difficulty of assuming translation given transcription information.

At the same time, judicious use of RNA-seq databases can be made by compacting and saving the information non-redundantly and using it to gather proteomic (translation level) information as an aide to genome annotation efforts. On the well-annotated *C. elegans* data, we still succeeded in identifying over seven thousand novel events. In developing our methods, we made many design choices, including 'mapping raw RNA reads' versus transcript assembly, and maintaing a single comprehensive database of all RNA. Our results suggest that this is a better, and more inclusive approach to combining RNA and protein data, and can be reused for all organisms. Our splice graph database construction pipeline produces a conventional FASTA database that can be applied to any kind of proteomics study, while achieving large scale data compression with no loss of useful information.

Utilizing RNA-seq information in proteogenomics database construction has many other benefits, including the computation of sample specific expression, and genomic variation identification. Future improvements of our pipeline will extend to mapping all variations in addition to splicing events, and further the use of proteomic data in genetic studies.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

V. B. and S. W. was supported by a grant from the NIH (P41-RR024851). S. C. was supported in part by the NSF grant (0924023).

References

1. Siva N. Nature biotechnology. 2008; 26:256–256.

2. Via M, Gignoux C, Burchard EG. Genome Med. 2010; 2:3. [PubMed: 20193048]
3. The ENCODE Project Consortium. Science. 2004; 306:636–640. [PubMed: 15499007]
4. Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, van Baren MJ, Salzberg SL, Wold BJ, Pachter L. Nat. Biotechnol. 2010; 28:511–515. [PubMed: 20436464]
5. Grabherr MG, et al. Nat. Biotechnol. 2011; 29:644–652. [PubMed: 21572440]
6. Verhaak RG, et al. Cancer Cell. 2010; 17:98–110. [PubMed: 20129251]
7. Wang X, Slebos RJ, Wang D, Halvey PJ, Tabb DL, Liebler DC, Zhang BJ. Proteome Res. 2012; 11:1009–1017.
8. Li J, Su Z, Ma ZQ, Slebos RJ, Halvey P, Tabb DL, Liebler DC, Pao W, Zhang B. Mol. Cell Proteomics. 2011; 10:M110.006536.
9. Castellana N, Bafna V. J Proteomics. 2010; 73:2124–2135. [PubMed: 20620248]
10. Castellana NE, Payne SH, Shen Z, Stanke M, Bafna V, Briggs SP. Proc. Natl. Acad. Sci. U.S.A. 2008; 105:21034–21038. [PubMed: 19098097]
11. Tanner S, Shen Z, Ng J, Florea L, Guigo R, Briggs SP, Bafna V. Genome Res. 2007; 17:231–239. [PubMed: 17189379]
12. Flicek P, et al. Nucleic Acids Res. 2013; 41:48–55.
13. Hillier LW, Reinke V, Green P, Hirst M, Marra MA, Waterston RH. Genome Res. 2009; 19:657–666. [PubMed: 19181841]
14. Gerstein MB, et al. Science. 2010; 330:1775–1787. [PubMed: 21177976]
15. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. Bioinformatics. 2009; 25:2078–2079. [PubMed: 19505943]
16. Edwards NJ. Mol. Syst. Biol. 2007; 3:102. [PubMed: 17437027]
17. Merrihew GE, Davis C, Ewing B, Williams G, Kall L, Frewen BE, Noble WS, Green P, Thomas JH, MacCoss MJ. Genome Res. 2008; 18:1660–1669. [PubMed: 18653799]
18. Brenner S. Genetics. 1974; 77:71–94. [PubMed: 4366476]
19. Tran JC, Doucette AA. Anal. Chem. 2009; 81:6201–6209. [PubMed: 19572727]
20. Kim S, Mischerikow N, Bandeira N, Navarro JD, Wich L, Mohammed S, Heck AJ, Pevzner PA. Mol. Cell Proteomics. 2010; 9:2840–2852. [PubMed: 20829449]
21. Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. Genome Res. 2002; 12:996–1006. [PubMed: 12045153]
22. Altschul, Stephen F.; Gish, Warren; Miller, Webb; Myers, Eugene W.; Lipman, David J. Journal of molecular biology. 1990; 215:403–410. others. [PubMed: 2231712]
23. Pruitt KD, Tatusova T, Klimke W, Maglott DR. Nucleic Acids Res. 2009; 37:D32–36. [PubMed: 18927115]
24. McEntyre, J, e.; Ostell, J., editors. The NCBI Handbook [Internet]. National Center for Biotechnology Information; 2002. http://www.ncbi.nlm.nih.gov/books/NBK21101/
25. DePina AS, Iser WB, Park SS, Maudsley S, Wilson MA, Wolkow CA. BMC Physiol. 2011; 11:11. [PubMed: 21749693]

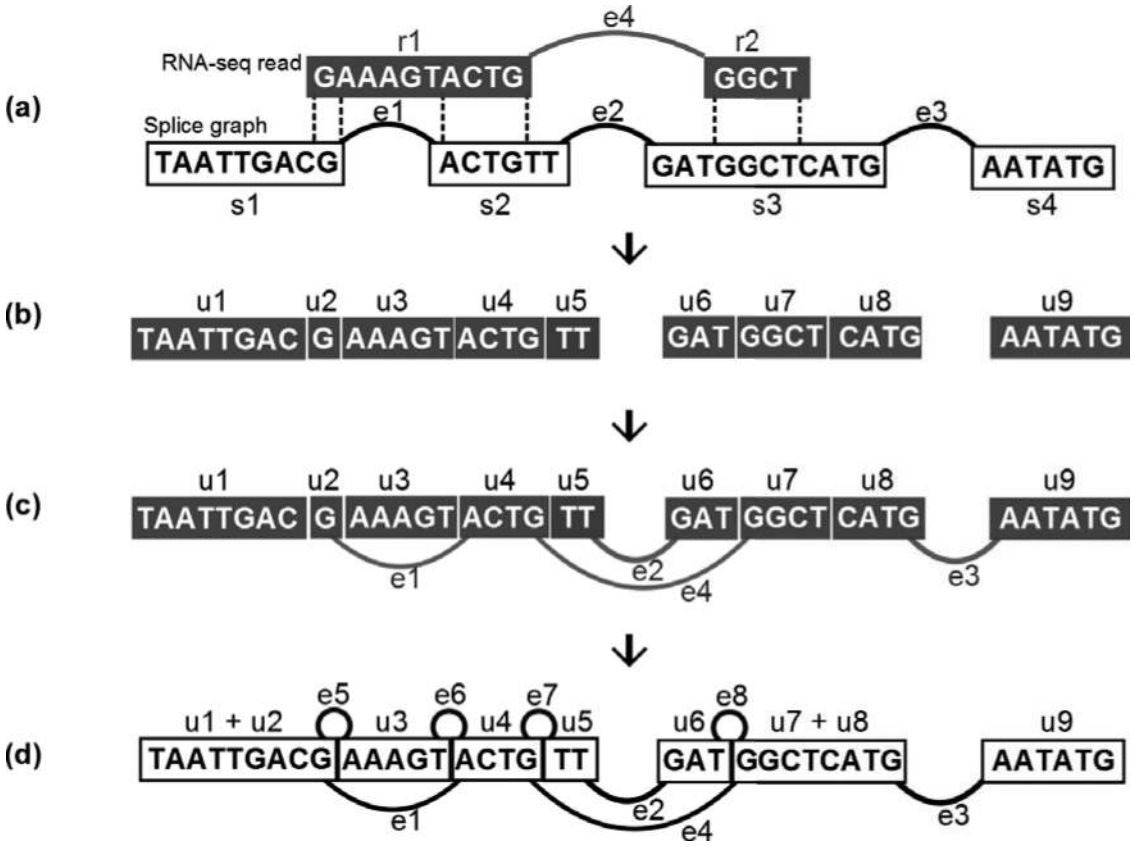


Figure 1.

(a) Given RNA-seq read, find overlapping regions with the existing splice graph. (b) Split and add nodes. (r_1, node s_1 is split into nodes u_1 and u_2, and node u_3 is added.) (c) Assign edges for each spliced-read. (d) Revisit each pair of contiguous nodes. The nodes are merged if there is no edge at the boundaries. (Nodes u_1 and u_2 are merged, while e_5 is added between u_2 and u_3.)

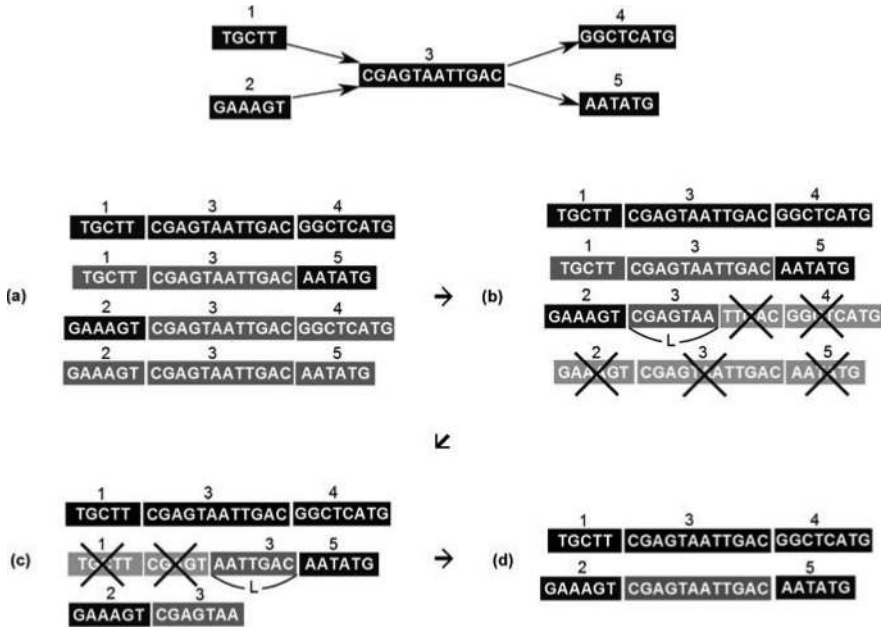
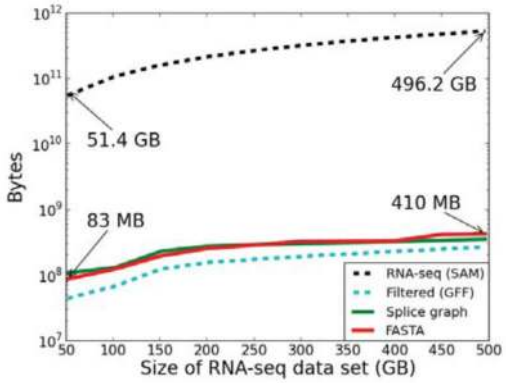
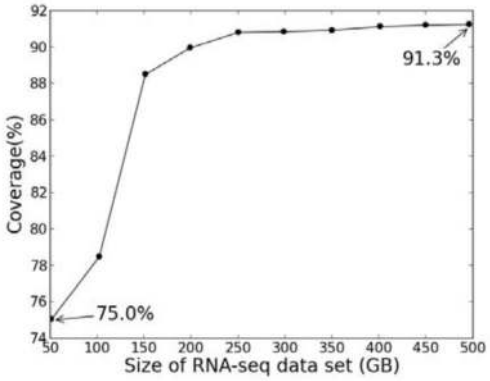


Figure 2.

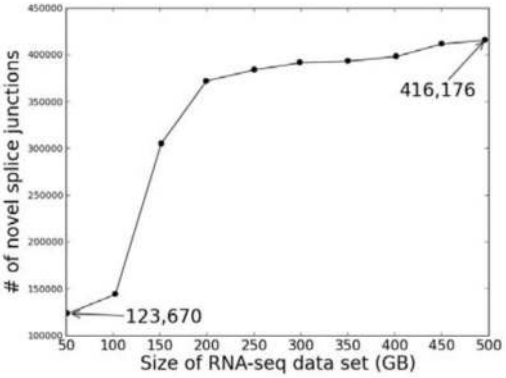
(a) By traversing the graph using a depth first search(DFS), we generate a sequence from the first visited start to end node path. (b) While traversing in DFS, when we encounter an outgoing edge that is already visited, only maintain a length $L-1$ suffix. (c) While traversing in DFS, when we encounter an incoming edge that is already visited, only maintain a length $L-1$ prefix. (d) For a pair of sequences(paths) with a prefix-suffix match, combine two sequences.



(a) Database file size growth



(b) Coverage plot



(c) Number of novel splice junction

Figure 3.

(a) Growth of the database file size(Bytes) while incorporating more RNA-seq data. (b) Increase in the percentage of covered splice junctions compared to RefSeq. (c) Increase in the number of splice junctions expressed in splice graph database which does not exist in RefSeq.

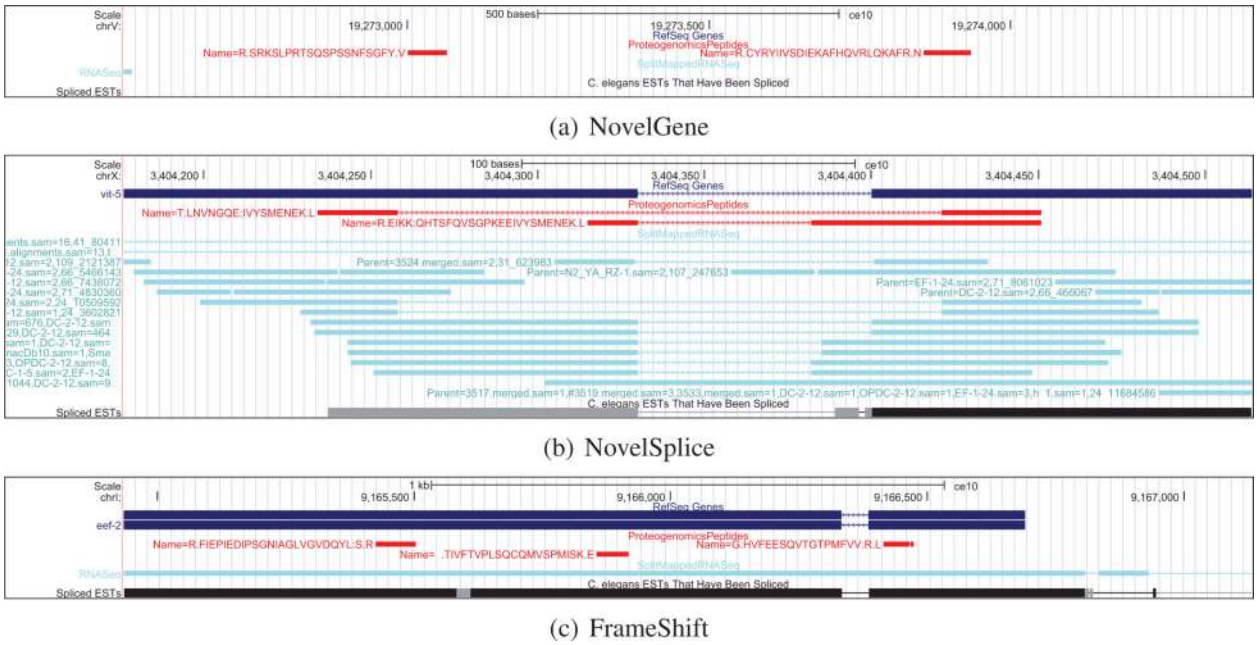


Figure 4.

(a) Shows a novel gene area where two peptides are identified in a non-genomic region. (b) Two peptides with alternative splice junctions. Peptide T.LNVNGQE:IVYSMENEK.L is supported by 13 split mapped RNA-seq reads, and R.EIKK:QHTSFQVSGPKEEIVYSMENEK.L is supported by 40 reads. (c) Peptide 'TIVFTVPLSQCQMVSPMISK.E' matches in a different frame compared to the gene eef-2. Two neighboring peptides, 'R.FIEPIEDIPSGNIAGLVGVDQYL:S.R', and 'G.HVFEESQVTGTPMFVV:R.L' are identified with 1 bp deletion, that allow for the frame-shift to occur.

Algorithm 1

Pseudo code for FASTA conversion

input : Splice graph G

output : A compact FASTA file F that is correct and L-complete with regard to G

FUNCTION main(Graph $G(V, E)$)

begin

  Forall $u \in V$

    If (Not(Visited(u))

      F=DFS(u, *EmptyString*)

  Print(F)

end

FUNCTION DFS(node u, str x)

begin

  If (Visited(u))

    F = DFSFiniteLength(u,x,L)

  Else If (TERMINAL(u))

    Visited(u) = True

      $F = F \cup \{x + \text{str}(u)\}$

  Else

    Visited(u) = True

    $x = x$ + str(u)

    ForAll (outgoing neighbors v)

      If (NotFirstEdge($u \rightarrow v$))

        $x = \text{suffix}_L(x)$

      F=DFS(v,x)

  Return(F)

end

input : Splice graph G, node u, str x, length l

output : All sequences of length l that start from node u

FUNCTION DFSFiniteLength(node u, str x, length l)

begin

  If (length(u) > l)

      $F = F \cup \{x + \text{prefix}_l(\text{str}(u))\}$

  Else If (TERMINAL(u))

      $F = F \cup \{x + \text{str}(u)\}$

  Else

    $l = l$ - length(u)

    $x = x$ + str(u)

    ForAll (outgoing neighbor v)

      If (NotFirstEdge($u \rightarrow v$))

        $x = \text{suffix}_L(x)$

F=DFSFiniteLength(*v,x,l*)

Return(*F*)

end

Table 1

The statistics of novel events identified

Novel Events	# of events
Alternative Splice	12
Novel Exon	808
Novel Gene	215
Exon Boundary	245
Gene Boundary	618
Reverse Strand	1166
Frame Shift	938
Translated UTR	42