

# **Conserved control signals in the transcriptome of higher plants**

**Khanh Tran**

**Thesis submitted for the degree of Doctor of Philosophy**

**May 2010**

**Discipline of Plant and Pest Science  
School of Agriculture, Food, and Wine  
The University of Adelaide**

## **APPENDIX**

## APPENDIX

### 1.1. uORFSCAN source code

The source code of uORFSCAN is written in Java (see below), and the executable uORFSCAN program can be found at <http://www.biomedcentral.com/1471-2164/9/361/additional/>.

```
/**
 * uORFPipeLine.java
 */
import java.util.*;
import java.io.*;

//The four stages of uORFPipeline: 1) initialisation, 2) find mainORF, 3) find
//uORFs, and 4) alignment of uORFs and result output.
public class uORFPipeLine
{
    /**
     * @param args not used.
     */
    public static void main (String[] args)
    {
        int    numOfSeq;
        List   listOfcDNAs;
        List   mainORFs;

        try
        {
            System.out.println("uORFScan v1.0");
            System.out.println("-----");

            System.out.print("Resetting folders...");
            uORFPipeLine.checkDirectory();
            uORFPipeLine.deleteDirectory(new File("uORFs"));
            uORFPipeLine.deleteDirectory(new File("alignments"));
            uORFPipeLine.deleteDirectory(new File("predictions"));
            uORFPipeLine.deleteDirectory(new File("errors"));
            System.out.println("completed!");

            System.out.print("\nEnter the MINIMUM number of species
                that uORFs need to be conserved in: ");
            uORFScan.minSpecieConservation = Keyboard.readInt();
            System.out.print("\nEnter length similarity of uORFs to
                be aligned (e.g. 0.2 for 20%): ");
            uORFScan.lengthSimilarity = Keyboard.readDouble();

            File directory = new File("inputs");
            String[] files = directory.list(new TxtFilter());
            if (files.length == 0)
            {
                System.out.println("\nERROR: No input files
                    detected!");
            }

            for (int i=0; i<files.length; i++)
            {
                System.out.println("\nLOADING FASTA FILE: " +
                    files[i] + "...(Stage 1 of 4)");

                numOfSeq =

```

```

        uORFScan.getNumberOfSequences("inputs/" +
            files[i]);
        listOfcDNAs = uORFScan.parseFASTA("inputs/" +
            files[i],numOfSeq);
        System.out.println("\tcompleted!");

        System.out.println("\nFINDING MAIN ORFs...(Stage
            2 of 4)");
        mainORFs = uORFScan.findMainORF(listOfcDNAs);
        System.out.println("\tcompleted!");

        if (mainORFs != null)
        {
            System.out.println("\nFINDING ALL
                uORFs...(Stage 3 of 4)");
            List orthoClustersFASTA =
                uORFScan.finduORFs(files[i], mainORFs,
                    mainORFs.size());

            //DENOTES CONSERVED uORFs ARE FOUND
            if (orthoClustersFASTA.size() != 0)
            {
                System.out.println("\tcompleted!");
                System.out.println("\nuORF
                    ALIGNMENT AND OUTPUT...(Stage
                        4 of 4)");

                Iterator itr2 = orthoClustersFASTA.listIterator(0);
                while (itr2.hasNext())
                {
                    File file = (File)itr2.next();
                    File aln = Align.align(file);
                    if (!aln.exists())
                        break;

                    String filePath = file.getPath();
                    filePath = filePath.replaceFirst("PEP","ORF");
                    File file2 = new File (filePath);
                    Map table = AlignParser.parseAlignment(aln);

                    double conservation =
                        AlignParser.calculateConservation(table);
                    System.out.println("\t\tAvg. sequence similarity: " +
                        conservation + " %\n");

                    AlignParser.writeResults(table,aln,conservation,file2);
                }

                System.out.println("\t\ncompleted!");
            }
        }
        else System.out.println("\tno uORFs found!");
    }
}
catch (Exception e)
{
    e.printStackTrace();
}
}

/**
 * Deletes all files in directory.
 * @param dir directory file
 */
public static void deleteDirectory(File dir)
{
    if (dir.isDirectory())
    {
        String[] children = dir.list();
        for (int i=0; i<children.length; i++)
        {
            deleteDirectory(new File(dir, children[i]));
        }
    }
    if (dir.getPath() != "uORFs" && dir.getPath() != "alignments" &&
        dir.getPath() != "predictions" && dir.getPath() != "errors")
        dir.delete();
}

```

```

    }
    else dir.delete();
}

/**
 * Creates new directories if they do not exist.
 */
public static void checkDirectory()
{
    List l = new ArrayList();
    l.add("uORFs");
    l.add("predictions");
    l.add("alignments");
    l.add("errors");
    l.add("inputs");

    Iterator itr = l.listIterator(0);
    while(itr.hasNext())
    {
        String str = (String)itr.next();
        File f = new File(str.toString());
        if (!f.exists())
        {
            f.mkdirs();
        }
    }
}

}

//*****
// uORFScan.java
//*****
import java.util.*;
import java.io.*;
import java.util.regex.*;

public class uORFScan
{
    static int minSpecieConservation;
    static double lengthSimilarity;

    /**
     * Count the number of sequences in the input file.
     * @param filename file name of input file.
     */
    public static int getNumberOfSequences(String fileName)
    {
        int counter = 0;
        BufferedReader br = null;
        try
        {
            br = new BufferedReader(new FileReader(fileName));
            String line = br.readLine();

            while (br.readLine() != null)
            {
                if (line.indexOf(">") == 0)
                    counter++;
                line = br.readLine();
            }
        }
        catch (Exception e)
        {
            System.out.println(e);
        }
        return counter;
    }
}

```

```

/**
 * Parses the input file and returns a List of cDNA objects.
 * @param filename file name of input file.
 * @param numberOfSequences number of sequences in input file.
 * @returns a List of cDNA objects.
 */
public static List parseFASTA(String fileName, int numberOfSequences)
{
    List listOfcDNAs = new ArrayList();
    BufferedReader br = null;
    StringBuffer sb = null;
    cDNA c = null;
    String line, id, description, sequence;
    String[] splits = null;

    try
    {
        br = new BufferedReader(new FileReader(fileName));
        line = br.readLine();

        for (int i=0; i<numberOfSequences; i++)
        {
            sb = new StringBuffer();
            splits = line.split("\\s",2);
            id = splits[0];
            id = id.substring(1,splits[0].length());
            if (splits.length == 1)
                description = "";
            else description = splits[1];
            line = br.readLine();
            sequence = line;
            c = new cDNA(id,description,sequence);
            listOfcDNAs.add(c);
            line = br.readLine();
        }
    }
    catch (Exception e)
    {
        System.out.println(e);
    }

    return listOfcDNAs;
}

/**
 * Finds ORFs.
 * @param sequence nucleotide sequence.
 * @param cDNAid identifier of cDNA.
 * @param cDNAdescription description of cDNA.
 * @param cDNAsequence sequence of cDNA.
 * @returns a List of sequences.
 */
public static List findORFs(String sequence, String cDNAid, String
cDNAdescription, String cDNAsequence)
{
    List ORFs = new ArrayList();
    Pattern ATGpattern = Pattern.compile("ATG");
    Matcher matchATG = ATGpattern.matcher(sequence);
    String sequenceContext;

    while (matchATG.find())
    {
        int ATGindex = matchATG.start();

        Pattern stopPattern =
            Pattern.compile("(...)?(TAG|TAA|TGA)");
        Matcher matchStop = stopPattern.matcher(sequence);

        while (matchStop.find(ATGindex))
        {
            String orfSequence = matchStop.group(0);
            int stopIndex = matchStop.end();
            int orfLength = stopIndex - ATGindex;

```

```

        if (orfSequence.startsWith("ATG") &&
            orfLength>=6)
        {
            if (ATGindex-3 <0)
            {
                sequenceContext = cDNASequence.substring(0,
                    ATGindex+7);
            }
            else
            {
                sequenceContext =
                    cDNASequence.substring(ATGindex-3, ATGindex+4);
            }

            ORF orf = new ORF (cDNAId, cDNADescription,
                cDNASequence, ATGindex, stopIndex, orfLength,
                orfSequence,sequence.length(),sequenceContext);

            ORFs.add(orf);
            break;
        }
        else break;
    }
    }
    return ORFs;
}

/**
 * Find main ORF.
 * @param listOfcDNAs a List of cDNA objects.
 * @returns a List of main ORFs.
 */
public static List findMainORF(List listOfcDNAs)
{
    List mainORFs = new ArrayList();
    List ORFs = null;
    cDNA cDNA = null;
    ORF orf = null;

    Iterator itr = listOfcDNAs.listIterator(0);
    while (itr.hasNext())
    {
        cDNA = (cDNA)itr.next();
        ORFs =
uORFScan.findORFs(cDNA.sequence,cDNA.id,cDNA.description,cDNA.sequence);
        Collections.sort(ORFs);
        if (ORFs.size(>0)
        {
            orf = (ORF)ORFs.get(0);
            orf.mORFStartIndex = orf.startIndex;
            orf.intercistronicDist = orf.mORFStartIndex - orf.endIndex;
            mainORFs.add(orf);
            System.out.println(orf.toString3());
        }
    }
    return mainORFs;
}
}

```

```

/**
 * Print main ORFs.
 * @param parentDir parent directory.
 * @param fileName file name.
 * @param mainORFs a list of main ORFs objects.
 */
public static void printMainORFs(String parentDir, String fileName, List
    mainORFs)
{
    BufferedWriter bw = null;

    try
    {
        fileName = fileName.substring(0,fileName.indexOf("."));
        bw = new BufferedWriter(new FileWriter(parentDir + "/" +
            fileName + "_mORF" + ".fasta"));

        Iterator itr = mainORFs.listIterator(0);
        while (itr.hasNext())
        {
            ORF mORF = (ORF)itr.next();
            bw.write(mORF.toString());
        }

        bw.close();

        ORF mORF = (ORF)mainORFs.get(0);
        bw = new BufferedWriter(new
            FileWriter("predictions/mORFContext.fasta",true));
        bw.write(mORF.id + "_mORF@" + mORF.startIndex + "_" +
            mORF.length + "_" + mORF.intercistronicDist + "\t" +
            mORF.sequenceContext + "\n");

        bw.close();
    }
    catch(IOException e)
    {
        e.printStackTrace();
    }
}

/**
 * Find uORFs.
 * @param filename input file name.
 * @param mainORFs a List of main ORF objects.
 * @param numOfMainORFs number of main ORFs
 * @returns a List of input files.
 */
public static List finduORFs(String fileName, List mainORFs, int
    numOfMainORFs)
{
    List fastaFiles = new ArrayList();
    File fasta = null;

    ORF mainORF = (ORF)mainORFs.get(0);
    String leaderSequence =
        mainORF.cDNASequence.substring(0,mainORF.startIndex);
    List uORFs =
        findORFs(leaderSequence,mainORF.id,mainORF.description,mainORF.cDNASequence);

    List conservedUORFs = null;

    Iterator itr1 = uORFs.listIterator(0);
    Iterator itr2 = mainORFs.listIterator(1);
    while (itr1.hasNext()) //iterate through uORFs
    {
        conservedUORFs = new ArrayList();

        ORF uORF = (ORF)itr1.next();
        conservedUORFs.add(uORF);
        while (itr2.hasNext())
        {
            ORF nextcDNA = (ORF)itr2.next();

```



```

        leaderSequence =
            nextcDNA.cDNASequence.substring(0,nextcDNA.startIndex);
        uORFs =
findORFs(leaderSequence,nextcDNA.id,nextcDNA.description,nextcDNA.cDNASequence)
;
        ORF bestORF = findBestMatchedORF(uORF,uORFs);

        if (bestORF != null)
            conservedUORFs.add(bestORF);

    }
    itr2 = mainORFs.listIterator(1);

    if (conservedUORFs.size()>=minSpecieConservation)
    {
        fasta =
uORFScan.writeConservedUORFs(fileName,conservedUORFs,mainORFs);
        fastaFiles.add(fasta);
    }

    return fastaFiles;
}

/**
 * Write conserved uORFs.
 * @param filename input file name.
 * @param conservedUORFs a List of conserved uORF objects.
 * @param mainORFs a List of main ORF objects.
 * @returns a input file.
 */
public static File writeConservedUORFs(String fileName, List
conservedUORFs,List mainORFs)
{
    BufferedWriter bw = null;
    String parentDir = null;
    String outputDir = null;
    String clusterSize = conservedUORFs.size() + "of " +
        mainORFs.size();
    File fasta = null;

    try
    {
        parentDir = "uORFs/" + clusterSize + "/" +
            fileName.replaceFirst(".txt","");
        File f = new File (parentDir);
        f.mkdirs();
        printMainORFs(parentDir,fileName,mainORFs);

        ORF uORF = (ORF)conservedUORFs.get(0);
        outputDir = parentDir + "/" + uORF.id + "_uORF@" +
uORF.startIndex + "_" + uORF.length + "_" + uORF.intercistronicDist + ".fasta";
        bw = new BufferedWriter(new FileWriter(outputDir,true));

        Iterator itr = conservedUORFs.listIterator(0);
        while(itr.hasNext())
        {
            ORF nextuORF = (ORF)itr.next();
            bw.write(nextuORF.toString());
        }
        bw.close();

        outputDir = parentDir + "/" + uORF.id + "_uPEP@" +
uORF.startIndex + "_" + uORF.length + "_" + uORF.intercistronicDist + ".fasta";
        bw = new BufferedWriter(new FileWriter(outputDir,true));
        fasta = new File (outputDir);

        itr = conservedUORFs.listIterator(0);
        while(itr.hasNext())
        {
            ORF nextuORF = (ORF)itr.next();
            bw.write(nextuORF.toString2());
        }
    }
}

```

```

        bw.close();

        outputDir = "predictions/uORFContext.fasta";
        bw = new BufferedWriter(new FileWriter(outputDir,true));
        bw.write(uORF.id + "_uORF@" + uORF.startIndex + "_" +
uORF.length + "_" + uORF.intercistronicDist + "\t" + uORF.sequenceContext +
"\n");

        bw.close();

    }
    catch(IOException e)
    {
        e.printStackTrace();
    }
    return fasta;
}

/**
 * Find best matched ORF.
 * @param queryORF query ORF.
 * @param subjectORFs subject ORFs.
 * @returns a ORF object.
 */
public static ORF findBestMatchedORF(ORF queryORF, List subjectORFs)
{
    List subjectORFsSimQueryLength = new ArrayList();
    Iterator itr = subjectORFs.listIterator(0);
    while (itr.hasNext())
    {
        ORF subjectORF = (ORF)itr.next();
        if (isSimilarLength(queryORF,subjectORF))
            subjectORFsSimQueryLength.add(subjectORF);
    }

    ORF emptyORF = null;
    ORF bestORF = null;

    if (subjectORFsSimQueryLength.size()==0)
    {
        bestORF = null;
    }
    else bestORF = (ORF)subjectORFsSimQueryLength.get(0);

    if (subjectORFsSimQueryLength.size()==0)
    {
        return emptyORF;
    }
    else
    {
        itr = subjectORFsSimQueryLength.listIterator(1);
        while (itr.hasNext())
        {
            ORF subjectORF = (ORF)itr.next();
            int a = Math.abs(bestORF.startIndex -
                queryORF.startIndex);
            int b = Math.abs(subjectORF.startIndex -
                queryORF.startIndex);
            int c = Math.min(a,b);
            if (c != a)
            {
                bestORF = subjectORF;
            }
        }
        return bestORF;
    }
}

/**
 * Returns true if two ORFs are similar in length, otherwise false.
 * @param queryORF query ORF.
 * @param subjectORF subject ORF.
 * @returns boolean.
 */

```

```

        public static boolean isSimilarLength(ORF queryORF, ORF subjectORF)
        {
            if (queryORF.length >= (subjectORF.length -
lengthSimilarity*queryORF.length) && queryORF.length <= (subjectORF.length +
lengthSimilarity*queryORF.length))
                return true;
            else return false;
        }
    }

    /**
     * Counts the number of sequences in the multiple alignment.
     * @param path of alignment file.
     * @returns the number of sequences in the multiple alignment.
     */
    public static int getNumberOfSequences(String path)
    {
        int counter = 0;

        try
        {
            BufferedReader br = new BufferedReader(new
                FileReader(path));

            String line = br.readLine();
            for (int i=0; i<3; i++)
                line = br.readLine();

            while (!line.startsWith(" "))
            {
                counter++;
                line = br.readLine();
            }
            br.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }

        return counter;
    }

    /**
     * Converts CLUSTAL W alignment so that there is no "wrapping".
     * @param aln alignment file.
     * @returns a Map structure of alignment.
     */
    public static Map parseAlignment (File aln)
    {
        String filePath = aln.getPath();
        int numberOfSequences = getNumberOfSequences(filePath);
        int rowNumber = 0;
        String[] subStrings = null;
        Map table = new LinkedHashMap();

        try
        {
            BufferedReader br = new BufferedReader(new
                FileReader(filePath));

            String line = br.readLine();
            for (int i=0; i<2; i++)
                line = br.readLine();

```

```

while (line != null)
{
    int rowLength=0;

    for (int i=0; i<numberOfSequences; i++)
    {
        line = br.readLine();
        subStrings = line.split("\\s",2);
        if (!table.containsKey(subStrings[0]))
        {
            StringBuffer sb = new
                StringBuffer();
            table.put(subStrings[0],
                sb.append(subStrings[1].trim()));
            rowLength =
                subStrings[1].trim().length();
        }
        else
        {
            StringBuffer sb =
                (StringBuffer)table.get(subStrings[0]);
            table.put(subStrings[0],
                sb.append(subStrings[1].trim()));
            rowLength =
                subStrings[1].trim().length();
        }
    }
    line = br.readLine();
    if (!table.containsKey("annotation"))
    {
        rowNumber++;
        StringBuffer sb = new StringBuffer();
        String annotationSubsequence =
            line.substring(line.length()-
                rowLength,line.length());
        table.put("annotation",
            sb.append(annotationSubsequence));
    }
    else
    {
        rowNumber++;
        StringBuffer sb =
            (StringBuffer)table.get("annotation");
        String annotationSubsequence =
            line.substring(line.length()-
                rowLength,line.length());
        table.put("annotation",
            sb.append(annotationSubsequence));
    }
    line = br.readLine();
}
br.close();
}
catch(IOException e)
{
    System.out.println(e);
}
return table;
}

/**
 * Calculate sequence conservation.
 * @param table Map structure of alignment.
 * returns a double value of sequence conservation.
 */
public static double calculateConservation(Map table)
{
    double conservation = 0.0;
    double count = 0.0;
    String annotationString =
        ((StringBuffer)table.get("annotation")).toString();

```

```

        for (int i=0;i<annotationString.length();i++)
        {
            if (annotationString.charAt(i) == '*')
                count++;
        }
        conservation =
            Math.round((count/annotationString.length()*100);

        return conservation;
    }

    /**
     * Output results.
     * @param table Map structure of alignment.
     * @param aln alignment file.
     * @param conservation conservation value.
     * @param file2 output file.
     */
    public static void writeResults (Map table, File aln, double
                                    conservation, File file2)
    {
        try
        {
            String clusterSize = aln.getPath();
            clusterSize =
                clusterSize.substring(clusterSize.indexOf("of")-
                    1,clusterSize.indexOf("of")+3);

            List uORFNTSEQ = new ArrayList();
            BufferedReader br = new BufferedReader(new
                FileReader(file2));

            String line = br.readLine();
            while (line !=null)
            {
                if (line.startsWith("ATG"))
                    uORFNTSEQ.add(line);
                line = br.readLine();
            }
            br.close();

            String outputFile =
                aln.getName().replaceFirst(".aln",".fasta");
            BufferedWriter bw1 = new BufferedWriter (new
                FileWriter(new File ("predictions/" + outputFile )));

            BufferedWriter bw2 = new BufferedWriter (new
                FileWriter(new File ("predictions/summary.txt"),true));
            bw2.write(clusterSize + "\t" + conservation + "\t");

            Set entries = table.entrySet();
            Iterator itr = entries.iterator();

            int counter = 0;
            while (itr.hasNext())
            {
                Map.Entry thisPair = (Map.Entry)itr.next();
                String key = (String)thisPair.getKey();
                StringBuffer value =
                    (StringBuffer)table.get(key);
                StringBuffer ORFSequence = removeGaps(value);

                if (key.indexOf("annotation") == -1)
                {
                    bw1.write(">" + key + "\n");
                    bw1.write(ORFSequence.toString() + "\n" );
                    bw2.write(key + "\t" +
                        ORFSequence.toString() + "\t" +
                        uORFNTSEQ.get(counter) + "\t" );
                    counter++;
                }
            }
            bw2.write("\n");
            bw1.close();
            bw2.close();
        }
    }

```

```

        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }

/**
 * Remove gaps from ORF sequence.
 * @param value value.
 * @returns a stringBuffer of ORF sequence without gaps.
 */
public static StringBuffer removeGaps(StringBuffer value)
{
    StringBuffer ORFSequence = new StringBuffer();
    for (int i=0; i<value.length(); i++)
    {
        if (value.charAt(i) != '-')
            ORFSequence.append(value.charAt(i));
    }
    return ORFSequence;
}

}

/*****
// Align.java
/*****
import java.io.*;
import java.util.*;

public class Align
{

    /**
     * Aligns an orthologue cluster.
     * @param fasta a File representing a fasta file.
     * @return reformatted CLUSTALW alignment file.
     */
    public static File align(File fasta)
    {
        File alignment = null;
        PrintStream ps = null;
        String fileName = null;
        BufferedWriter bw = null;
        BufferedReader br = null;

        try
        {
            FileOutputStream fos = new FileOutputStream(new
                File("errors/errors.txt"),true);
            ps = new PrintStream(fos);
            System.setErr(ps);

            File file1 = fasta.getParentFile();
            File file2 = file1.getParentFile();

            fileName = fasta.getName();
            System.out.println("\t" + fileName);

            String outFileParent =
                fasta.getParent().replace("uORFs", "alignments");
            File file = new File (outFileParent);
            file.mkdirs();

            String outFilePath = outFileParent + "\\\" +
                fileName.replaceFirst(".fasta", ".aln");
            alignment = new File(outFilePath);
            Runtime rt = Runtime.getRuntime();
            Process proc = rt.exec("cmd /c clustalw " +
                fasta.getPath() + " -type=PROTEIN -
                outorder=input -outfile=" + outFilePath);
            InputStream is = proc.getInputStream();
            InputStreamReader isr = new InputStreamReader(is);
            br = new BufferedReader(isr);

```

```

        br.close();
        int exitVal = proc.waitFor();
        if (exitVal == 0)
        {
            System.out.println("\t\taligned successfully");
        }
        else
        {
            System.out.println("\t\tERROR: could not align
file!");
            bw = new BufferedWriter(new FileWriter(new
                File("errors/errors.txt"),true));
            bw.write(fileName + "\t\tCLUSTALW ERROR: could not
                align!\n");
            bw.close();
        }

        System.out.println("\t\tconserved in: " +
            file2.getName());
    }
    catch(Exception e)
    {
        ps.println(e);
        ps.close();
    }
    return alignment;
}
}

}

//*****
// cDNA.java
//*****
class cDNA
{
    String id;
    String description;
    String sequence;

    /**
     * @param id identifier of cDNA
     * @param description description of cDNA
     * @param sequence nucleotide sequence of cDNA
     */
    public cDNA(String id, String description, String sequence)
    {
        this.id = id;
        this.description = description;
        this.sequence = sequence;
    }

    /**
     * Return string description of cDNA
     */
    public String toString()
    {
        return id + " " + description + "\n" + sequence + "\n";
    }
}

//*****
// ORF.java
//*****
import org.biojava.bio.symbol.*;
import org.biojava.bio.seq.*;
import java.util.*;
import java.io.*;

public class ORF implements Comparable
{
    String id;
    String description;
    String cDNASequence;

```

```

String orfSequence;
String protSeq;
String sequenceContext;
int startIndex;
int endIndex;
int length;
int mORFStartIndex;
int intercistronicDist;

/**
 * @param id ORF identifier
 * @param description ORF annotation
 * @param cDNASequence cDNA sequence
 * @param startIndex start index of ORF
 * @param endIndex end index of ORF
 * @param length length of ORF in nucleotides
 * @param orfSequence ORF nucleotide sequence
 * @param mORFStartIndex main ORF start index
 * @param sequenceContext sequence context of ORF
 */
public ORF (String id, String description, String cDNASequence, int
startIndex, int endIndex, int length, String orfSequence, int mORFStartIndex,
String sequenceContext)
{
    try
    {
        this.id = id;
        this.description = description;
        this.cDNASequence = cDNASequence;
        this.startIndex = startIndex;
        this.endIndex = endIndex;
        this.orfSequence = orfSequence;
        this.length = length;
        this.protSeq = null;
        this.mORFStartIndex = mORFStartIndex;
        this.sequenceContext = sequenceContext;
        intercistronicDist = mORFStartIndex - endIndex;
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}

/**
 * Returns ORF that has best possible match in position
 */
public static ORF compareORF(ORF parent, ORF current, ORF temp)
{
    if (current == null)
        return temp;
    else
    {
        int a = Math.abs(current.startIndex - parent.startIndex);
        int b = Math.abs(temp.startIndex - parent.startIndex);
        int c = Math.min(a,b);
        if (c == a)
            return current;
        else return temp;
    }
}

/**
 * Returns boolean true if ORF objects have same identifier;
 * otherwise false
 */
public boolean equals(Object obj)
{
    String[] split1 = id.split("_");
    String[] split2 = ((ORF)obj).id.split("_");

    if (obj instanceof ORF)
        return (split1[1].equals(split2[1]));
    else return false;
}

```



```

    }

    /**
     * Identifier id used as hashcode
     */
    public int hashCode()
    {
        String[] split = id.split("_");
        return split[1].hashCode();
    }

    /**
     * Returns integer 0 if ORF objects have same length; otherwise 1
     */
    public final int compareTo(Object obj)
    {
        return ((ORF)obj).length - this.length;
    }

    /**
     * String description of uORF
     */
    public String toString()
    {
        String result = ">" + id + "_uORF@" + startIndex + "_" + length
            + "_" + intercistronicDist + " " + description + "\n" +
            orfSequence + "\n";
        return result;
    }

    /**
     * String description of upeptide
     */
    public String toString2()
    {
        String result = null;
        try
        {
            org.biojava.bio.seq.Sequence seq =
                DNATools.createDNASequence(orfSequence,
                    "nucleotideSequence");
            SymbolList symL = seq.subList(1,seq.length()-3);
            symL = RNATools.transcribe(symL);
            symL = RNATools.translate(symL);
            orfSequence = symL.seqString();
            protSeq = orfSequence;

            result = ">" + id + "_uPEP@" + startIndex + "_" + length
                + "_" + intercistronicDist + " " + description + "\n"
                + protSeq + "\n";
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        return result;
    }

    /**
     * String description of main ORF
     */
    public String toString3()
    {
        String result = "\t" + id + "_mORF@" + startIndex + "_" +
            endIndex + "_" + mORFstartIndex;
        return result;
    }
}

/*****
// TxtFilter.java
/*****
import java.io.File;
import java.io.FilenameFilter;

```

```

import java.io.IOException;

class TxtFilter implements FilenameFilter
{
    /**
     * Select only *.txt files.
     * @param dir the directory in which the file was found.
     * @param name the name of the file.
     * @returns true if and only if the name should be included in the file list;
     * false otherwise.
     */
    public boolean accept (File dir, String name)
    {
        if (new File(dir, name).isDirectory())
        {
            return false;
        }
        name = name.toLowerCase();
        return name.endsWith(".txt");
    }
}

//*****
// Keyboard.java
//*****

import java.io.*;
import java.util.*;

public class Keyboard
{
    private static boolean printErrors = true;
    private static int errorCount = 0;

    /**
     * Returns the current error count.
     */
    public static int getErrorCount()
    {
        return errorCount;
    }

    /**
     * Resets the current error count to zero.
     */
    public static void resetErrorCount (int count)
    {
        errorCount = 0;
    }

    /**
     * Returns a boolean indicating whether input errors are currently printed
     * to standard output.
     */
    public static boolean getPrintErrors()
    {
        return printErrors;
    }

    /**
     * Sets a boolean indicating whether input errors are to be
     * printed to standard output.
     */
    public static void setPrintErrors (boolean flag)
    {
        printErrors = flag;
    }

    /**
     * Increments the error count and prints the error message if
     * appropriate.
     */
    private static void error (String str)

```

```

{
    errorCount++;
    if (printErrors)
        System.out.println (str);
}

private static String current_token = null;
private static StringTokenizer reader;

private static BufferedReader in = new BufferedReader
    (new InputStreamReader(System.in));

/**
 * Gets the next input token assuming it may be on subsequent
 * input lines.
 */
private static String getNextToken()
{
    return getNextToken (true);
}

/**
 * Gets the next input token, which may already have been read.
 */
private static String getNextToken (boolean skip)
{
    String token;

    if (current_token == null)
        token = getNextInputToken (skip);
    else
    {
        token = current_token;
        current_token = null;
    }

    return token;
}

/**
 * Gets the next token from the input, which may come from the
 * current input line or a subsequent one. The parameter
 * determines if subsequent lines are used.
 */
private static String getNextInputToken (boolean skip)
{
    final String delimiters = " \t\n\r\f";
    String token = null;

    try
    {
        if (reader == null)
            reader = new StringTokenizer
                (in.readLine(), delimiters, true);

        while (token == null ||
            ((delimiters.indexOf (token) >= 0) && skip))
        {
            while (!reader.hasMoreTokens())
                reader = new StringTokenizer
                    (in.readLine(), delimiters, true);

            token = reader.nextToken();
        }
    }
    catch (Exception exception)
    {
        token = null;
    }

    return token;
}

/**

```

```

* Returns true if there are no more tokens to read on the
* current input line.
*/
public static boolean endOfLine()
{
    return !reader.hasMoreTokens();
}

/**
* Returns a string read from standard input.
*/
public static String readString()
{
    String str;

    try
    {
        str = getNextToken(false);
        while (! endOfLine())
        {
            str = str + getNextToken(false);
        }
    }
    catch (Exception exception)
    {
        error ("Error reading String data, null value returned.");
        str = null;
    }
    return str;
}

/**
* Returns a space-delimited substring (a word) read from
* standard input.
*/
public static String readWord()
{
    String token;
    try
    {
        token = getNextToken();
    }
    catch (Exception exception)
    {
        error ("Error reading String data, null value returned.");
        token = null;
    }
    return token;
}

/**
* Returns a boolean read from standard input.
*/
public static boolean readBoolean()
{
    String token = getNextToken();
    boolean bool;
    try
    {
        if (token.toLowerCase().equals("true"))
            bool = true;
        else if (token.toLowerCase().equals("false"))
            bool = false;
        else
        {
            error ("Error reading boolean data, false value returned.");
            bool = false;
        }
    }
    catch (Exception exception)
    {
        error ("Error reading boolean data, false value returned.");
        bool = false;
    }
}

```

```

    return bool;
}

/**
 * Returns a character read from standard input.
 */
public static char readChar()
{
    String token = getNextToken(false);
    char value;
    try
    {
        if (token.length() > 1)
        {
            current_token = token.substring (1, token.length());
        } else
            current_token = null;

        value = token.charAt (0);
    }
    catch (Exception exception)
    {
        error ("Error reading char data, MIN_VALUE value returned.");
        value = Character.MIN_VALUE;
    }

    return value;
}

/**
 * Returns an integer read from standard input.
 */
public static int readInt()
{
    String token = getNextToken();
    int value;
    try
    {
        value = Integer.parseInt (token);
    }
    catch (Exception exception)
    {
        error ("Error reading int data, MIN_VALUE value returned.");
        value = Integer.MIN_VALUE;
    }
    return value;
}

/**
 * Returns a long integer read from standard input.
 */
public static long readLong()
{
    String token = getNextToken();
    long value;
    try
    {
        value = Long.parseLong (token);
    }
    catch (Exception exception)
    {
        error ("Error reading long data, MIN_VALUE value returned.");
        value = Long.MIN_VALUE;
    }
    return value;
}

/**
 * Returns a float read from standard input.
 */
public static float readFloat()
{
    String token = getNextToken();
    float value;

```

```

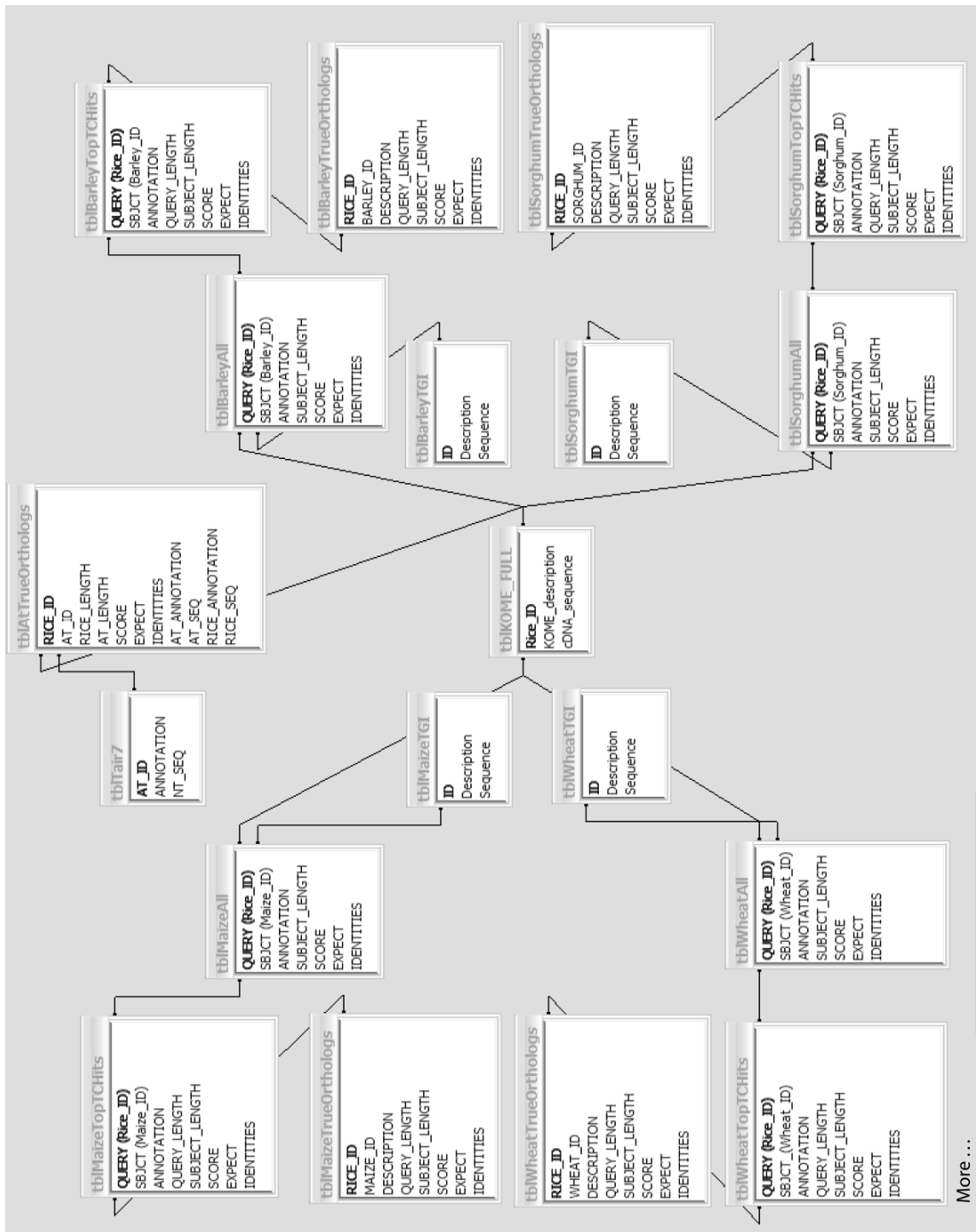
    try
    {
        value = (new Float(token)).floatValue();
    }
    catch (Exception exception)
    {
        error ("Error reading float data, NaN value returned.");
        value = Float.NaN;
    }
    return value;
}

/**
 * Returns a double read from standard input.
 */
public static double readDouble()
{
    String token = getNextToken();
    double value;
    try
    {
        value = (new Double(token)).doubleValue();
    }
    catch (Exception exception)
    {
        error ("Error reading double data, NaN value returned.");
        value = Double.NaN;
    }
    return value;
}
}

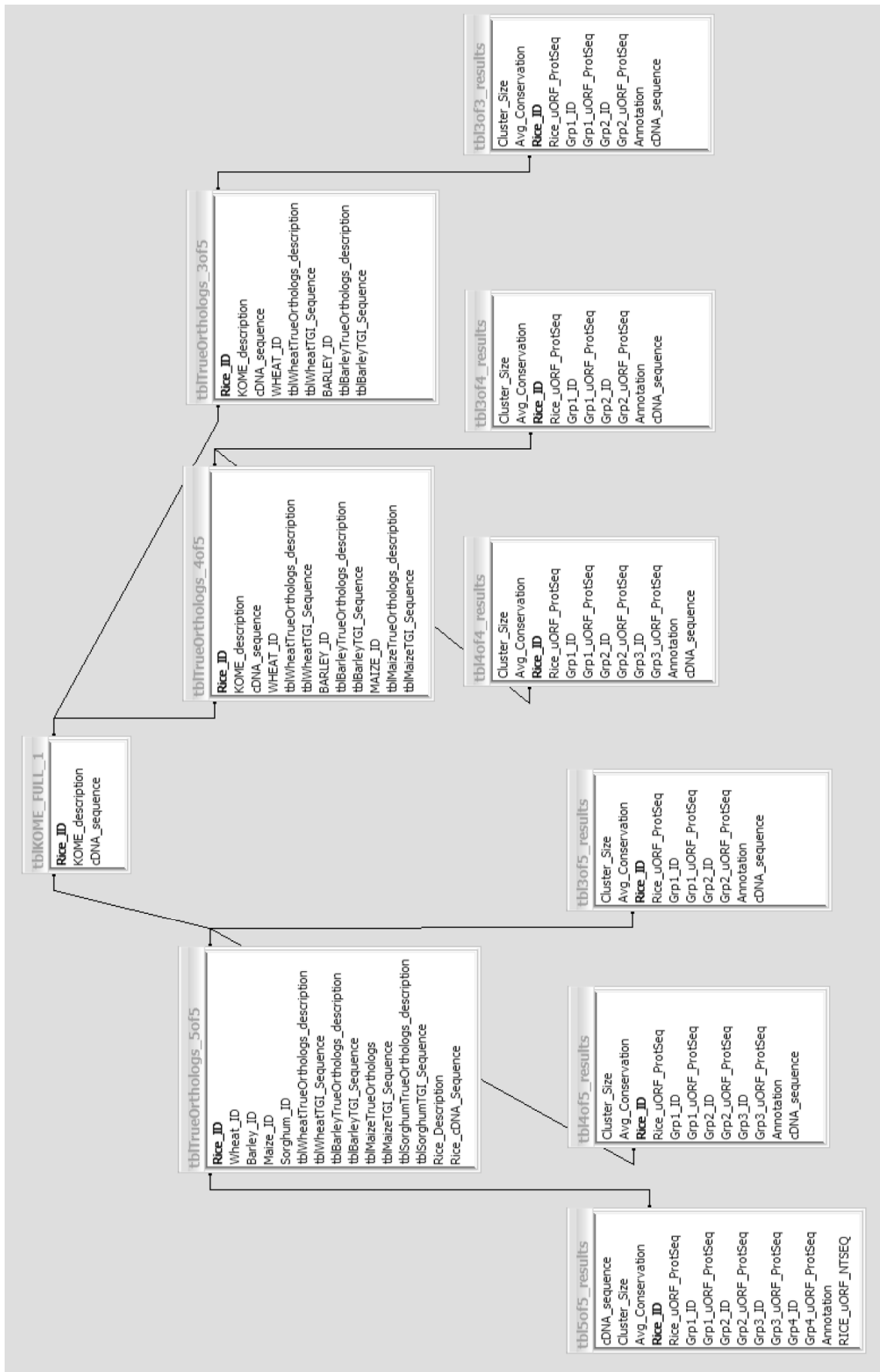
```

## 1.2. Database schema used to manage the data generated by uORFSCAN

Figure 1A Database schema used to manage the data generated by uORFSCAN. Image (right) is generated by Microsoft Access 2003, which is the database management system employed to create and manage the physical database. Tables are prefixed with “tbl” to differentiate them from both queries and reports (not shown). Primary keys used to relate data from one table to another are indicated in bold. Each table contains a list of field names, but the data type for each field name is not shown in this particular view (right). However, identifiers (ID) and expect are set to data type “text”; sequences (SEQ) and annotations/descriptions are set to data type “memo”; length, score, and identities are set to data type “number”.







### 1.3. Overview of CMPSCAN



Figure 2 Flow chart of CMPSCAN algorithm.

## BIBLIOGRAPHY

- Adams, M.D., J.M. Kelley, J.D. Gocayne, M. Dubnick, M.H. Polymeropoulos, H. Xiao, C.R. Merril, A. Wu, B. Olde, R.F. Moreno, and et al. 1991. Complementary DNA sequencing: expressed sequence tags and human genome project. *Science* **252**: 1651-1656.
- Alberts, B., A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. 2002. *Molecular biology of the cell*. Garland Science, New York.
- Alonso, J.M., A.N. Stepanova, T.J. Leisse, C.J. Kim, H. Chen, P. Shinn, D.K. Stevenson, J. Zimmerman, P. Barajas, R. Cheuk, C. Gadrinab, C. Heller, A. Jeske, E. Koesema, C.C. Meyers, H. Parker, L. Prednis, Y. Ansari, N. Choy, H. Deen, M. Geralt, N. Hazari, E. Hom, M. Karnes, C. Mulholland, R. Ndubaku, I. Schmidt, P. Guzman, L. Aguilar-Henonin, M. Schmid, D. Weigel, D.E. Carter, T. Marchand, E. Risseuw, D. Brogden, A. Zeko, W.L. Crosby, C.C. Berry, and J.R. Ecker. 2003. Genome-wide insertional mutagenesis of *Arabidopsis thaliana*. *Science* **301**: 653-657.
- Altschul, S.F., W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. 1990. Basic local alignment search tool. *J Mol Biol* **215**: 403-410.
- Amsterdam, A., R.M. Nissen, Z. Sun, E.C. Swindell, S. Farrington, and N. Hopkins. 2004. Identification of 315 genes essential for early zebrafish development. *Proc Natl Acad Sci U S A* **101**: 12792-12797.
- Anderson, L. and J. Seilhamer. 1997. A comparison of selected mRNA and protein abundances in human liver. *Electrophoresis* **18**: 533-537.
- Arnaud, N., K. Ravet, A. Borlotti, B. Touraine, J. Boucherez, C. Fizames, J.F. Briat, F. Cellier, and F. Gaymard. 2007. The iron-responsive element (IRE)/iron-regulatory protein 1 (IRP1)-cytosolic aconitase iron-regulatory switch does not operate in plants. *Biochem J* **405**: 523-531.
- Aylon, Y. and M. Kupiec. 2004. New insights into the mechanism of homologous recombination in yeast. *Mutat Res* **566**: 231-248.
- Bailey-Serres, J., R. Sorenson, and P. Juntawong. 2009. Getting the message across: cytoplasmic ribonucleoprotein complexes. *Trends Plant Sci* **14**: 443-453.
- Baim, S.B. and F. Sherman. 1988. mRNA structures influencing translation in the yeast *Saccharomyces cerevisiae*. *Mol Cell Biol* **8**: 1591-1601.

- Batey, R.T., R.P. Rambo, and J.A. Doudna. 1999. Tertiary Motifs in RNA Structure and Folding. *Angew Chem Int Ed Engl* **38**: 2326-2343.
- Baumann, U. and J. Juttner. 2002. Plant thioredoxins: the multiplicity conundrum. *Cell Mol Life Sci* **59**: 1042-1057.
- Bayer, T.S., L.N. Booth, S.M. Knudsen, and A.D. Ellington. 2005. Arginine-rich motifs present multiple interfaces for specific binding by RNA. *Rna* **11**: 1848-1857.
- Benson, D.A., I. Karsch-Mizrachi, D.J. Lipman, J. Ostell, and E.W. Sayers. 2009. GenBank. *Nucleic Acids Res* **37**: D26-31.
- Bi, X. and Y.S. Rong. 2003. Genome manipulation by homologous recombination in *Drosophila*. *Brief Funct Genomic Proteomic* **2**: 142-146.
- Bologna, N.G., J.L. Mateos, E.G. Bresso, and J.F. Palatnik. 2009. A loop-to-base processing mechanism underlies the biogenesis of plant microRNAs miR319 and miR159. *EMBO J*.
- Bork, P., T. Dandekar, Y. Diaz-Lazcoz, F. Eisenhaber, M. Huynen, and Y. Yuan. 1998. Predicting function: from genes to genomes and back. *J Mol Biol* **283**: 707-725.
- Brierley, I., R.J. Gilbert, and S. Pennell. 2008. RNA pseudoknots and the regulation of protein synthesis. *Biochem Soc Trans* **36**: 684-689.
- Brunel, C., R. Marquet, P. Romby, and C. Ehresmann. 2002. RNA loop-loop interactions as dynamic functional motifs. *Biochimie* **84**: 925-944.
- Byrne, M.E. 2009. A role for the ribosome in development. *Trends Plant Sci* **14**: 512-519.
- Cannell, I.G., Y.W. Kong, and M. Bushell. 2008. How do microRNAs regulate gene expression? *Biochem Soc Trans* **36**: 1224-1231.
- Canzoneri, J.C. and A.K. Oyelere. 2008. Interaction of anthracyclines with iron responsive element mRNAs. *Nucleic Acids Res* **36**: 6825-6834.
- Castrignano, T., A. Canali, G. Grillo, S. Liuni, F. Mignone, and G. Pesole. 2004. CSTminer: a web tool for the identification of coding and noncoding conserved sequence tags through cross-species genome comparison. *Nucleic Acids Res* **32**: W624-627.

- Castrignano, T., P.D. De Meo, G. Grillo, S. Liuni, F. Mignone, I.G. Talamo, and G. Pesole. 2006. GenoMiner: a tool for genome-wide search of coding and non-coding conserved sequence tags. *Bioinformatics* **22**: 497-499.
- Chang, K.S., S.H. Lee, S.B. Hwang, and K.Y. Park. 2000. Characterization and translational regulation of the arginine decarboxylase gene in carnation (*Dianthus caryophyllus* L.). *Plant J* **24**: 45-56.
- Chapple, C.E., R. Guigo, and A. Krol. 2009. SECISaln, a web-based tool for the creation of structure-based alignments of eukaryotic SECIS elements. *Bioinformatics* **25**: 674-675.
- Chastain, M. and I. Tinoco, Jr. 1991. Structural elements in RNA. *Prog Nucleic Acid Res Mol Biol* **41**: 131-177.
- Chaudhuri, K. and R. Chatterjee. 2007. MicroRNA detection and target prediction: integration of computational and experimental approaches. *DNA Cell Biol* **26**: 321-337.
- Chen, H.H., D. Castanotto, J.M. LeBon, J.J. Rossi, and A.D. Riggs. 2000a. *In vivo*, high-resolution analysis of yeast and mammalian RNA-protein interactions, RNA structure, RNA splicing and ribozyme cleavage by use of terminal transferase-dependent PCR. *Nucleic Acids Res* **28**: 1656-1664.
- Chen, H.H., J. Lebon, and A.D. Riggs. 2008. Analysis of RNA structure and RNA-protein interactions in mammalian cells by use of terminal transferase-dependent PCR. *Methods Mol Biol* **488**: 319-341.
- Chen, J.H., S.Y. Le, and J.V. Maizel. 2000b. Prediction of common secondary structures of RNAs: a genetic algorithm approach. *Nucleic Acids Res* **28**: 991-999.
- Chen, S.J. 2008. RNA folding: conformational statistics, folding kinetics, and ion electrostatics. *Annu Rev Biophys* **37**: 197-214.
- Chen, S.J. and K.A. Dill. 2000. RNA folding energy landscapes. *Proc Natl Acad Sci U S A* **97**: 646-651.
- Child, S.J., M.K. Miller, and A.P. Geballe. 1999. Translational control by an upstream open reading frame in the *HER-2/neu* transcript. *J Biol Chem* **274**: 24335-24341.

- Chumpolkulwong, N., K. Sakamoto, A. Hayashi, F. Iraha, N. Shinya, N. Matsuda, D. Kiga, A. Urushibata, M. Shirouzu, K. Oki, T. Kigawa, and S. Yokoyama. 2006. Translation of 'rare' codons in a cell-free protein synthesis system from *Escherichia coli*. *J Struct Funct Genomics* **7**: 31-36.
- Clote, P. 2005. RNALOSS: a web server for RNA locally optimal secondary structures. *Nucleic Acids Res* **33**: W600-604.
- Cok, S.J., S.J. Acton, A.E. Sexton, and A.R. Morrison. 2004. Identification of RNA-binding proteins in RAW 264.7 cells that recognize a lipopolysaccharide-responsive element in the 3-untranslated region of the murine cyclooxygenase-2 mRNA. *J Biol Chem* **279**: 8196-8205.
- Col, B., S. Oltean, and R. Banerjee. 2007. Translational regulation of human methionine synthase by upstream open reading frames. *Biochim Biophys Acta* **1769**: 532-540.
- Combiere, J.P., F. de Billy, P. Gamas, A. Niebel, and S. Rivas. 2008. Trans-regulation of the expression of the transcription factor *MtHAP2-1* by a uORF controls root nodule development. *Genes Dev* **22**: 1549-1559.
- Combiere, J.P., F. Frugier, F. de Billy, A. Boualem, F. El-Yahyaoui, S. Moreau, T. Vernie, T. Ott, P. Gamas, M. Crespi, and A. Niebel. 2006. *MtHAP2-1* is a key transcriptional regulator of symbiotic nodule development regulated by microRNA169 in *Medicago truncatula*. *Genes Dev* **20**: 3084-3088.
- Crooks, G.E., G. Hon, J.M. Chandonia, and S.E. Brenner. 2004. WebLogo: a sequence logo generator. *Genome Res* **14**: 1188-1190.
- Crowe, M.L., X.Q. Wang, and J.A. Rothnagel. 2006. Evidence for conservation and selection of upstream open reading frames suggests probable encoding of bioactive peptides. *BMC Genomics* **7**: 16.
- Cvijovic, M., D. Dalevi, E. Bilslund, G.J. Kemp, and P. Sunnerhagen. 2007. Identification of putative regulatory upstream ORFs in the yeast genome using heuristics and evolutionary conservation. *BMC Bioinformatics* **8**: 295.
- Czar, M.J., J.C. Anderson, J.S. Bader, and J. Peccoud. 2009. Gene synthesis demystified. *Trends Biotechnol* **27**: 63-72.

- Damiani, R.D., Jr. and S.R. Wessler. 1993. An upstream open reading frame represses expression of *Lc*, a member of the R/B family of maize transcriptional activators. *Proc Natl Acad Sci U S A* **90**: 8244-8248.
- Daou-Chabo, R. and C. Condon. 2009. RNase J1 endonuclease activity as a probe of RNA secondary structure. *RNA* **15**: 1417-1425.
- Dausse, E., C. Belair, C. Staedel, and J.J. Toulme. 2008. Regulating mRNA translation with a kiss. *Nucleic Acids Symp Ser (Oxf)*: 711-712.
- David-Assael, O., H. Saul, V. Saul, T. Mizrachy-Dagri, I. Berezin, E. Brook, and O. Shaul. 2005. Expression of *AtMHX*, an Arabidopsis vacuolar metal transporter, is repressed by the 5' untranslated region of its gene. *J Exp Bot* **56**: 1039-1047.
- Davis, A.R. and B.M. Znosko. 2007. Thermodynamic characterization of single mismatches found in naturally occurring RNA. *Biochemistry* **46**: 13425-13436.
- Delbecq, P., M. Werner, A. Feller, R.K. Filipkowski, F. Messenguy, and A. Pierard. 1994. A segment of mRNA encoding the leader peptide of the *CPA1* gene confers repression by arginine on a heterologous yeast gene transcript. *Mol Cell Biol* **14**: 2378-2390.
- Ditzer, A. and D. Bartels. 2006. Identification of a dehydration and ABA-responsive promoter regulon and isolation of corresponding DNA binding proteins for the group 4 LEA gene *CpC2* from *C. plantagineum*. *Plant Mol Biol* **61**: 643-663.
- dos Santos, C.O., L.C. Dore, E. Valentine, S.G. Shelat, R.C. Hardison, M. Ghosh, W. Wang, R.S. Eisenstein, F.F. Costa, and M.J. Weiss. 2008. An iron responsive element-like stem-loop regulates alpha-hemoglobin-stabilizing protein mRNA. *J Biol Chem* **283**: 26956-26964.
- Du, T. and P.D. Zamore. 2005. microPrimer: the biogenesis and function of microRNA. *Development* **132**: 4645-4652.
- Duconge, F. and J.J. Toulme. 1999. *In vitro* selection identifies key determinants for loop-loop interactions: RNA aptamers selective for the TAR RNA element of HIV-1. *Rna* **5**: 1605-1614.
- Ebhardt, H.A. and P.J. Unrau. 2009. Characterizing multiple exogenous and endogenous small RNA populations in parallel with subfemtomolar sensitivity using a streptavidin gel-shift assay. *RNA* **15**: 724-731.

- Eddy, S.R. and R. Durbin. 1994. RNA sequence analysis using covariance models. *Nucleic Acids Res* **22**: 2079-2088.
- Esakova, O., A. Perederina, C. Quan, M.E. Schmitt, and A.S. Krasilnikov. 2008. Footprinting analysis demonstrates extensive similarity between eukaryotic RNase P and RNase MRP holoenzymes. *RNA* **14**: 1558-1567.
- Fang, P., C.C. Spevak, C. Wu, and M.S. Sachs. 2004. A nascent polypeptide domain that can regulate translation elongation. *Proc Natl Acad Sci U S A* **101**: 4059-4064.
- Fernandez, J., I. Yaman, C. Huang, H. Liu, A.B. Lopez, A.A. Komar, M.G. Caprara, W.C. Merrick, M.D. Snider, R.J. Kaufman, W.H. Lamers, and M. Hatzoglou. 2005. Ribosome stalling regulates IRES-mediated translation in eukaryotes, a parallel to prokaryotic attenuation. *Mol Cell* **17**: 405-416.
- Ferre-D'Amare, A.R. and J.A. Doudna. 2001. Methods to crystallize RNA. *Curr Protoc Nucleic Acid Chem* **Chapter 7**: Unit 7 6.
- Ferre, F., Y. Ponty, W.A. Lorenz, and P. Clote. 2007. DIAL: a web server for the pairwise alignment of two RNA three-dimensional structures using nucleotide, dihedral angle and base-pairing similarities. *Nucleic Acids Res* **35**: W659-668.
- Floris, R., H. Mahgoub, E. Lanet, C. Robaglia, and B. Menand. 2009. Post-transcriptional Regulation of Gene Expression in Plants during Abiotic Stress. *Int J Mol Sci* **10**: 3168-3185.
- Franceschetti, M., C. Hanfrey, S. Scaramagli, P. Torrigiani, N. Bagni, D. Burtin, and A.J. Michael. 2001. Characterization of monocot and dicot plant S-adenosyl-l-methionine decarboxylase gene families including identification in the mRNA of a highly conserved pair of upstream overlapping open reading frames. *Biochem J* **353**: 403-409.
- Freeling, M. and S. Subramaniam. 2009. Conserved noncoding sequences (CNSs) in higher plants. *Curr Opin Plant Biol* **12**: 126-132.
- Frith, M.C., A.R. Forrest, E. Nourbakhsh, K.C. Pang, C. Kai, J. Kawai, P. Carninci, Y. Hayashizaki, T.L. Bailey, and S.M. Grimmond. 2006. The abundance of short proteins in the mammalian proteome. *PLoS Genet* **2**: e52.



- Fulton, D.L., Y.Y. Li, M.R. Laird, B.G. Horsman, F.M. Roche, and F.S. Brinkman. 2006. Improving the specificity of high-throughput ortholog prediction. *BMC Bioinformatics* **7**: 270.
- Furtig, B., J. Buck, V. Manoharan, W. Bermel, A. Jaschke, P. Wenter, S. Pitsch, and H. Schwalbe. 2007. Time-resolved NMR studies of RNA folding. *Biopolymers* **86**: 360-383.
- Furtig, B., C. Richter, J. Wohnert, and H. Schwalbe. 2003. NMR spectroscopy of RNA. *Chembiochem* **4**: 936-962.
- Gaba, A., Z. Wang, T. Krishnamoorthy, A.G. Hinnebusch, and M.S. Sachs. 2001. Physical evidence for distinct mechanisms of translational control by upstream open reading frames. *EMBO J* **20**: 6453-6463.
- Gautheret, D., D. Konings, and R.R. Gutell. 1995. G.U base pairing motifs in ribosomal RNA. *Rna* **1**: 807-814.
- Giedroc, D.P. and P.V. Cornish. 2009. Frameshifting RNA pseudoknots: structure and mechanism. *Virus Res* **139**: 193-208.
- Giedroc, D.P., C.A. Theimer, and P.L. Nixon. 2000. Structure, stability and function of RNA pseudoknots involved in stimulating ribosomal frameshifting. *J Mol Biol* **298**: 167-185.
- Gorodkin, J., L.J. Heyer, and G.D. Stormo. 1997. Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Res* **25**: 3724-3732.
- Gorodkin, J., S.L. Stricklin, and G.D. Stormo. 2001. Discovering common stem-loop motifs in unaligned RNA sequences. *Nucleic Acids Res* **29**: 2135-2144.
- Gutell, R.R., N. Larsen, and C.R. Woese. 1994. Lessons from an evolving rRNA: 16S and 23S rRNA structures from a comparative perspective. *Microbiol Rev* **58**: 10-26.
- Gygi, S.P., Y. Rochon, B.R. Franza, and R. Aebersold. 1999. Correlation between protein and mRNA abundance in yeast. *Mol Cell Biol* **19**: 1720-1730.
- Hanfrey, C., K.A. Elliott, M. Franceschetti, M.J. Mayer, C. Illingworth, and A.J. Michael. 2005. A dual upstream open reading frame-based autoregulatory circuit controlling polyamine-responsive translation. *J Biol Chem* **280**: 39229-39237.

- Hanfrey, C., M. Franceschetti, M.J. Mayer, C. Illingworth, and A.J. Michael. 2002. Abrogation of upstream open reading frame-mediated translational control of a plant S-adenosylmethionine decarboxylase results in polyamine disruption and growth perturbations. *J Biol Chem* **277**: 44131-44139.
- Hao, Y.J., Z. Zhang, H. Kitashiba, C. Honda, B. Ubi, M. Kita, and T. Moriguchi. 2005. Molecular cloning and functional characterization of two apple S-adenosylmethionine decarboxylase genes and their different expression in fruit development, cell growth and stress responses. *Gene* **350**: 41-50.
- Hayden, C.A. and G. Bosco. 2008. Comparative genomic analysis of novel conserved peptide upstream open reading frames in *Drosophila melanogaster* and other dipteran species. *BMC Genomics* **9**: 61.
- Hayden, C.A. and R.A. Jorgensen. 2007. Identification of novel conserved peptide uORF homology groups in Arabidopsis and rice reveals ancient eukaryotic origin of select groups and preferential association with transcription factor-encoding genes. *BMC Biol* **5**: 32.
- Hemmings-Mieszczak, M., T. Hohn, and T. Preiss. 2000. Termination and peptide release at the upstream open reading frame are required for downstream translation on synthetic shunt-competent mRNA leaders. *Mol Cell Biol* **20**: 6212-6223.
- Hertz, G.Z., G.W. Hartzell, 3rd, and G.D. Stormo. 1990. Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Comput Appl Biosci* **6**: 81-92.
- Hilbers, C.W., P.J. Michiels, and H.A. Heus. 1998. New developments in structure determination of pseudoknots. *Biopolymers* **48**: 137-153.
- Hirosawa-Takamori, M., D. Ossipov, S.V. Novoselov, A.A. Turanov, Y. Zhang, V.N. Gladyshev, A. Krol, G. Vorbruggen, and H. Jackle. 2008. A novel stem loop control element-dependent UGA read-through system without translational selenocysteine incorporation in *Drosophila*. *FASEB J*.
- Hofacker, I.L., W. Fontana, P.F. Stadler, L.S. Bonhoeffer, M. Tacker, and P. Schuster. 1994. Fast Folding and Comparison of RNA Secondary Structures. *Monatsh.Chem* **125**: 167-188.

- Holcik, M., N. Sonenberg, and R.G. Korneluk. 2000. Internal ribosome initiation of translation and the control of cell death. *Trends Genet* **16**: 469-473.
- Hu, W.W., H. Gong, and E.C. Pua. 2005. The pivotal roles of the plant S-adenosylmethionine decarboxylase 5' untranslated leader sequence in regulation of gene expression at the transcriptional and posttranscriptional levels. *Plant Physiol* **138**: 276-286.
- Hu, Y.J. 2002. Prediction of consensus structural motifs in a family of coregulated RNA sequences. *Nucleic Acids Res* **30**: 3886-3893.
- Hu, Y.J. 2003. GPRM: A genetic programming approach to finding common RNA secondary structure elements. *Nucleic Acids Res* **31**: 3446-3449.
- Hulzink, R.J., P.F. de Groot, A.F. Croes, W. Quaedvlieg, D. Twell, G.J. Wullems, and M.M. Van Herpen. 2002. The 5'-untranslated region of the *ntp303* gene strongly enhances translation during pollen tube growth, but not during pollen maturation. *Plant Physiol* **129**: 342-353.
- Iacono, M., F. Mignone, and G. Pesole. 2005. uAUG and uORFs in human and rodent 5'untranslated mRNAs. *Gene* **349**: 97-105.
- Imai, A., Y. Hanzawa, M. Komura, K.T. Yamamoto, Y. Komeda, and T. Takahashi. 2006. The dwarf phenotype of the Arabidopsis *acl5* mutant is suppressed by a mutation in an upstream ORF of a *bHLH* gene. *Development* **133**: 3575-3585.
- Iseli, C., C.V. Jongeneel, and P. Bucher. 1999. ESTScan: a program for detecting, evaluating, and reconstructing potential coding regions in EST sequences. *Proc Int Conf Intell Syst Mol Biol*: 138-148.
- James, B.D., G.J. Olsen, and N.R. Pace. 1989. Phylogenetic comparative analysis of RNA secondary structure. *Methods Enzymol* **180**: 227-239.
- Ji, Y., X. Xu, and G.D. Stormo. 2004. A graph theoretical approach for predicting common RNA secondary structure motifs including pseudoknots in unaligned sequences. *Bioinformatics* **20**: 1591-1602.
- Jiao, Y., J.L. Riechmann, and E.M. Meyerowitz. 2008. Transcriptome-wide analysis of uncapped mRNAs in Arabidopsis reveals regulation of mRNA degradation. *Plant Cell* **20**: 2571-2585.

- Jin, X., E. Turcott, S. Englehardt, G.J. Mize, and D.R. Morris. 2003. The two upstream open reading frames of oncogene *mdm2* have different translational regulatory properties. *J Biol Chem* **278**: 25716-25721.
- Jordan, I.K., I.B. Rogozin, Y.I. Wolf, and E.V. Koonin. 2002. Essential genes are more evolutionarily conserved than are nonessential genes in bacteria. *Genome Res* **12**: 962-968.
- Joshi, C.P., H. Zhou, X. Huang, and V.L. Chiang. 1997. Context sequences of translation initiation codon in plants. *Plant Mol Biol* **35**: 993-1001.
- Juttner, J., D. Olde, P. Langridge, and U. Baumann. 2000. Cloning and expression of a distinct subclass of plant thioredoxins. *Eur J Biochem* **267**: 7109-7117.
- Kawaguchi, R. and J. Bailey-Serres. 2002. Regulation of translational initiation in plants. *Curr Opin Plant Biol* **5**: 460-465.
- Kawaguchi, R. and J. Bailey-Serres. 2005. mRNA sequence features that contribute to translational regulation in Arabidopsis. *Nucleic Acids Res* **33**: 955-965.
- Ke, Y., H. Sierzputowska-Gracz, Z. Gdaniec, and E.C. Theil. 2000. Internal loop/bulge and hairpin loop of the iron-responsive element of ferritin mRNA contribute to maximal iron regulatory protein 2 binding and translational regulation in the iso-iron-responsive element/iso-iron regulatory protein family. *Biochemistry* **39**: 6235-6242.
- Ke, Y., J. Wu, E.A. Leibold, W.E. Walden, and E.C. Theil. 1998. Loops and bulge/loops in iron-responsive element isoforms influence iron regulatory protein binding. Fine-tuning of mRNA regulation? *J Biol Chem* **273**: 23637-23640.
- Kikuchi, S., K. Satoh, T. Nagata, N. Kawagashira, K. Doi, N. Kishimoto, J. Yazaki, M. Ishikawa, H. Yamada, H. Ooka, I. Hotta, K. Kojima, T. Namiki, E. Ohneda, W. Yahagi, K. Suzuki, C.J. Li, K. Ohtsuki, T. Shishiki, Y. Otomo, K. Murakami, Y. Iida, S. Sugano, T. Fujimura, Y. Suzuki, Y. Tsunoda, T. Kurosaki, T. Kodama, H. Masuda, M. Kobayashi, Q. Xie, M. Lu, R. Narikawa, A. Sugiyama, K. Mizuno, S. Yokomizo, J. Niikura, R. Ikeda, J. Ishibiki, M. Kawamata, A. Yoshimura, J. Miura, T. Kusumegi, M. Oka, R. Ryu, M. Ueda, K. Matsubara, J. Kawai, P. Carninci, J. Adachi, K. Aizawa, T. Arakawa, S. Fukuda, A. Hara, W. Hashizume, N. Hayatsu, K. Imotani, Y. Ishii, M. Itoh, I. Kagawa, S. Kondo, H. Konno, A. Miyazaki, N. Osato, Y. Ota, R. Saito, D. Sasaki, K. Sato, K. Shibata, A. Shinagawa, T. Shiraki, M. Yoshino, Y. Hayashizaki, and A. Yasunishi. 2003. Collection, mapping,

and annotation of over 28,000 cDNA clones from japonica rice. *Science* **301**: 376-379.

Klinkert, B., I. Elles, and J. Nickelsen. 2006. Translation of chloroplast *psbD* mRNA in *Chlamydomonas* is controlled by a secondary RNA structure blocking the AUG start codon. *Nucleic Acids Res* **34**: 386-394.

Kochetov, A.V., S. Ahmad, V. Ivanisenko, O.A. Volkova, N.A. Kolchanov, and A. Sarai. 2008. uORFs, reinitiation and alternative translation start sites in human mRNAs. *FEBS Lett* **582**: 1293-1297.

Kozak, M. 1987a. An analysis of 5'-noncoding sequences from 699 vertebrate messenger RNAs. *Nucleic Acids Res* **15**: 8125-8148.

Kozak, M. 1987b. At least six nucleotides preceding the AUG initiator codon enhance translation in mammalian cells. *J Mol Biol* **196**: 947-950.

Kozak, M. 1989. Circumstances and mechanisms of inhibition of translation by secondary structure in eucaryotic mRNAs. *Mol Cell Biol* **9**: 5134-5142.

Kozak, M. 2005. Regulation of translation via mRNA structure in prokaryotes and eukaryotes. *Gene* **361**: 13-37.

Kozak, U.C., J. Kopecky, J. Teisinger, S. Enerback, B. Boyer, and L.P. Kozak. 1994. An upstream enhancer regulating brown-fat-specific expression of the mitochondrial uncoupling protein gene. *Mol Cell Biol* **14**: 59-67.

Kvaratskhelia, M. and S.F. Grice. 2008. Structural analysis of protein-RNA interactions with mass spectrometry. *Methods Mol Biol* **488**: 213-219.

Kwak, S.H. and S.H. Lee. 2001. The regulation of ornithine decarboxylase gene expression by sucrose and small upstream open reading frame in tomato (*Lycopersicon esculentum* Mill). *Plant Cell Physiol* **42**: 314-323.

Lai, E.C., P. Tomancak, R.W. Williams, and G.M. Rubin. 2003. Computational identification of *Drosophila* microRNA genes. *Genome Biol* **4**: R42.

Landers, J.E., S.L. Cassel, and D.L. George. 1997. Translational enhancement of *mdm2* oncogene expression in human tumor cells containing a stabilized wild-type p53 protein. *Cancer Res* **57**: 3562-3568.

Lawrence, C.E., S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, and J.C. Wootton. 1993. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* **262**: 208-214.

- Le Roch, K.G., J.R. Johnson, L. Florens, Y. Zhou, A. Santrosyan, M. Grainger, S.F. Yan, K.C. Williamson, A.A. Holder, D.J. Carucci, J.R. Yates, 3rd, and E.A. Winzeler. 2004. Global analysis of transcript and protein levels across the *Plasmodium falciparum* life cycle. *Genome Res* **14**: 2308-2318.
- Lee, M.M., S.H. Lee, and K.Y. Park. 1997. Characterization and expression of two members of the S-adenosylmethionine decarboxylase gene family in carnation flower. *Plant Mol Biol* **34**: 371-382.
- Lee, R.C., R.L. Feinbaum, and V. Ambros. 1993. The *C. elegans* heterochronic gene *lin-4* encodes small RNAs with antisense complementarity to *lin-14*. *Cell* **75**: 843-854.
- Lee, Y., R. Sultana, G. Pertea, J. Cho, S. Karamycheva, J. Tsai, B. Parvizi, F. Cheung, V. Antonescu, J. White, I. Holt, F. Liang, and J. Quackenbush. 2002. Cross-referencing eukaryotic genomes: TIGR Orthologous Gene Alignments (TOGA). *Genome Res* **12**: 493-502.
- Lee, Y., J. Tsai, S. Sunkara, S. Karamycheva, G. Pertea, R. Sultana, V. Antonescu, A. Chan, F. Cheung, and J. Quackenbush. 2005. The TIGR Gene Indices: clustering and assembling EST and known genes and integration with eukaryotic genomes. *Nucleic Acids Res* **33**: D71-74.
- Lemieux, S. and F. Major. 2002. RNA canonical and non-canonical base pairing types: a recognition method and complete repertoire. *Nucleic Acids Res* **30**: 4250-4263.
- Lewis, S., M. Ashburner, and M.G. Reese. 2000. Annotating eukaryote genomes. *Curr Opin Struct Biol* **10**: 349-354.
- Licatalosi, D.D., A. Mele, J.J. Fak, J. Ule, M. Kayikci, S.W. Chi, T.A. Clark, A.C. Schweitzer, J.E. Blume, X. Wang, J.C. Darnell, and R.B. Darnell. 2008. HITS-CLIP yields genome-wide insights into brain alternative RNA processing. *Nature* **456**: 464-469.
- Lincoln, J.E. and R.L. Fischer. 1988. Diverse mechanisms for the regulation of ethylene-inducible gene expression. *Mol Gen Genet* **212**: 71-75.
- Locatelli, F., E. Magnani, C. Vighi, C. LanzaNova, and I. Coraggio. 2002. Inhibitory effect of *myb7* uORF on downstream gene expression in homologous (rice) and heterologous (tobacco) systems. *Plant Mol Biol* **48**: 309-318.

- Lohmer, S., M. Maddaloni, M. Motto, F. Salamini, and R.D. Thompson. 1993. Translation of the mRNA of the maize transcriptional activator *Opaque-2* is inhibited by upstream open reading frames present in the leader sequence. *Plant Cell* **5**: 65-73.
- Lovett, P.S. and E.J. Rogers. 1996. Ribosome regulation by the nascent peptide. *Microbiol Rev* **60**: 366-385.
- Lu, G. and E.N. Moriyama. 2004. Vector NTI, a balanced all-in-one sequence analysis suite. *Brief Bioinform* **5**: 378-388.
- Luo, Z. and M.S. Sachs. 1996. Role of an upstream open reading frame in mediating arginine-specific translational control in *Neurospora crassa*. *J Bacteriol* **178**: 2172-2177.
- Mad Arif, S.A., M.A. Taylor, L.A. George, A.R. Butler, L.R. Burch, H.V. Davies, M.J. Stark, and A. Kumar. 1994. Characterisation of the S-adenosylmethionine decarboxylase (SAMDC) gene of potato. *Plant Mol Biol* **26**: 327-338.
- Mangone, M., P. Macmenamin, C. Zegar, F. Piano, and K.C. Gunsalus. 2008. UTRome.org: a platform for 3'UTR biology in *C. elegans*. *Nucleic Acids Res* **36**: D57-62.
- Manuell, A.L., J. Quispe, and S.P. Mayfield. 2007. Structure of the chloroplast ribosome: novel domains for translation regulation. *PLoS Biol* **5**: e209.
- Marin-Navarro, J., A.L. Manuell, J. Wu, and P.M. S. 2007. Chloroplast translation regulation. *Photosynth Res* **94**: 359-374.
- Martin, G.W., 3rd, J.W. Harney, and M.J. Berry. 1996. Selenocysteine incorporation in eukaryotes: insights into mechanism and efficiency from sequence, structure, and spacing proximity studies of the type 1 deiodinase SECIS element. *RNA* **2**: 171-182.
- Mathews, D.H. 2005. Predicting a set of minimal free energy RNA secondary structures common to two sequences. *Bioinformatics* **21**: 2246-2253.
- Mathews, D.H. and D.H. Turner. 2002. Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. *J Mol Biol* **317**: 191-203.
- Mathews, D.H. and D.H. Turner. 2006. Prediction of RNA secondary structure by free energy minimization. *Curr Opin Struct Biol* **16**: 270-278.

- Meijer, H.A. and A.A. Thomas. 2002. Control of eukaryotic protein synthesis by upstream open reading frames in the 5'-untranslated region of an mRNA. *Biochem J* **367**: 1-11.
- Meijer, H.A. and A.A. Thomas. 2003. Ribosomes stalling on uORF1 in the *Xenopus Cx41* 5' UTR inhibit downstream translation initiation. *Nucleic Acids Res* **31**: 3174-3184.
- Mignone, F., A. Anselmo, G. Donvito, G.P. Maggi, G. Grillo, and G. Pesole. 2008. Genome-wide identification of coding and non-coding conserved sequence tags in human and mouse genomes. *BMC Genomics* **9**: 277.
- Mignone, F., C. Gissi, S. Liuni, and G. Pesole. 2002. Untranslated regions of mRNAs. *Genome Biol* **3**: reviews0004.0001–reviews0004.0010.
- Mochida, K., T. Yoshida, T. Sakurai, Y. Ogihara, and K. Shinozaki. 2009. TriFLDB: a database of clustered full-length coding sequences from Triticeae with applications to comparative grass genomics. *Plant Physiol* **150**: 1135-1146.
- Monde, R.A., J.C. Greene, and D.B. Stern. 2000. The sequence and secondary structure of the 3'-UTR affect 3'-end maturation, RNA accumulation, and translation in tobacco chloroplasts. *Plant Mol Biol* **44**: 529-542.
- Mooers, B.H. 2009. Crystallographic studies of DNA and RNA. *Methods* **47**: 168-176.
- Moreno-Hagelsieb, G. and K. Latimer. 2008. Choosing BLAST options for better detection of orthologs as reciprocal best hits. *Bioinformatics* **24**: 319-324.
- Muckenthaler, M., N.K. Gray, and M.W. Hentze. 1998. IRP-1 binding to ferritin mRNA prevents the recruitment of the small ribosomal subunit by the cap-binding complex eIF4F. *Mol Cell* **2**: 383-388.
- Munroe, S.H. 2004. Diversity of antisense regulation in eukaryotes: Multiple mechanisms, emerging patterns. *J Cell Biochem* **93**: 664-671.
- Nadershahi, A., S.C. Fahrenkrug, and L.B. Ellis. 2004. Comparison of computational methods for identifying translation initiation sites in EST data. *BMC Bioinformatics* **5**: 14.
- Ng, D.W., M.B. Chandrasekharan, and T.C. Hall. 2004. The 5' UTR negatively regulates quantitative and spatial expression from the ABI3 promoter. *Plant Mol Biol* **54**: 25-38.



- Niepel, M., J. Ling, and D.R. Gallie. 1999. Secondary structure in the 5'-leader or 3'-untranslated region reduces protein yield but does not affect the functional interaction between the 5'-cap and the poly(A) tail. *FEBS Lett* **462**: 79-84.
- Nishimura, T., T. Wada, K.T. Yamamoto, and K. Okada. 2005. The Arabidopsis STV1 protein, responsible for translation reinitiation, is required for auxin-mediated gynoecium patterning. *Plant Cell* **17**: 2940-2953.
- Nyiko, T., B. Sonkoly, Z. Merai, A.H. Benkovics, and D. Silhavy. 2009. Plant upstream ORFs can trigger nonsense-mediated mRNA decay in a size-dependent manner. *Plant Mol Biol* **71**: 367-378.
- Orso, F., E. Cottone, M.D. Hasleton, J.C. Ibbitt, P. Sismondi, H.C. Hurst, and M. De Bortoli. 2004. Activator protein-2gamma (AP-2gamma) expression is specifically induced by oestrogens through binding of the oestrogen receptor to a canonical element within the 5'-untranslated region. *Biochem J* **377**: 429-438.
- Pavesi, G., G. Mauri, M. Stefani, and G. Pesole. 2004. RNAProfile: an algorithm for finding conserved secondary structure motifs in unaligned RNA sequences. *Nucleic Acids Res* **32**: 3258-3269.
- Pavesi, G., P. Mereghetti, F. Zambelli, M. Stefani, G. Mauri, and G. Pesole. 2006. MoD Tools: regulatory motif discovery in nucleotide sequences from co-regulated or homologous genes. *Nucleic Acids Res* **34**: W566-570.
- Pavesi, G., F. Zambelli, and G. Pesole. 2007. WeederH: an algorithm for finding conserved regulatory motifs and regions in homologous sequences. *BMC Bioinformatics* **8**: 46.
- Perriquet, O., H. Touzet, and M. Dauchet. 2003. Finding the common structure shared by two homologous RNAs. *Bioinformatics* **19**: 108-116.
- Pesole, G., C. Gissi, G. Grillo, F. Licciulli, S. Liuni, and C. Saccone. 2000. Analysis of oligonucleotide AUG start codon context in eukaryotic mRNAs. *Gene* **261**: 85-91.
- Petros, A.M. and S.W. Fesik. 1994. Nuclear magnetic resonance methods for studying protein-ligand complexes. *Methods Enzymol* **239**: 717-739.

- Pickering, B.M. and A.E. Willis. 2005. The implications of structured 5' untranslated regions on translation and disease. *Semin Cell Dev Biol* **16**: 39-47.
- Picoult-Newberg, L., T.E. Ideker, M.G. Pohl, S.L. Taylor, M.A. Donaldson, D.A. Nickerson, and M. Boyce-Jacino. 1999. Mining SNPs from EST databases. *Genome Res* **9**: 167-174.
- Pikovskaya, O., A.A. Serganov, A. Polonskaia, A. Serganov, and D.J. Patel. 2009. Preparation and crystallization of riboswitch-ligand complexes. *Methods Mol Biol* **540**: 115-128.
- Poyry, T.A., A. Kaminski, and R.J. Jackson. 2004. What determines whether mammalian ribosomes resume scanning after translation of a short upstream open reading frame? *Genes Dev* **18**: 62-75.
- Proudhon, D., J. Wei, J. Briat, and E.C. Theil. 1996. Ferritin gene organization: differences between plants and animals suggest possible kingdom-specific selective constraints. *J Mol Evol* **42**: 325-336.
- Puchta, H. 2002. Gene replacement by homologous recombination in plants. *Plant Mol Biol* **48**: 173-182.
- Rabani, M., M. Kertesz, and E. Segal. 2008. Computational prediction of RNA structural motifs involved in posttranscriptional regulatory processes. *Proc Natl Acad Sci U S A* **105**: 14885-14890.
- Rahmani, F., M. Hummel, J. Schuurmans, A. Wiese-Klinkenberg, S. Smeeckens, and J. Hanson. 2009. Sucrose control of translation mediated by an upstream open reading frame-encoded peptide. *Plant Physiol* **150**: 1356-1367.
- Raney, A., A.C. Baron, G.J. Mize, G.L. Law, and D.R. Morris. 2000. *In vitro* translation of the upstream open reading frame in the mammalian mRNA encoding S-adenosylmethionine decarboxylase. *J Biol Chem* **275**: 24444-24450.
- Rogers, J.T., J.D. Randall, C.M. Cahill, P.S. Eder, X. Huang, H. Gunshin, L. Leiter, J. McPhee, S.S. Sarang, T. Utsuki, N.H. Greig, D.K. Lahiri, R.E. Tanzi, A.I. Bush, T. Giordano, and S.R. Gullans. 2002. An iron-responsive element type II in the 5'-untranslated region of the Alzheimer's amyloid precursor protein transcript. *J Biol Chem* **277**: 45518-45528.

- Rowland, L.J. and J.N. Strommer. 1986. Anaerobic treatment of maize roots affects transcription of *Adh1* and transcript stability. *Mol Cell Biol* **6**: 3368-3372.
- Ruiz-Echevarria, M.J. and S.W. Peltz. 2000. The RNA binding protein Pub1 modulates the stability of transcripts containing upstream open reading frames. *Cell* **101**: 741-751.
- Russo, A., C. Cirulli, A. Amoresano, P. Pucci, C. Pietropaolo, and G. Russo. 2008. cis-acting sequences and trans-acting factors in the localization of mRNA for mitochondrial ribosomal proteins. *Biochim Biophys Acta* **1779**: 820-829.
- Sankoff, D. 1985. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math* **45**: 810-825.
- SantaLucia, J., Jr. and D.H. Turner. 1997. Measuring the thermodynamics of RNA secondary structure formation. *Biopolymers* **44**: 309-319.
- Savochkina, L., V. Alekseenkova, T. Belyanko, N. Dobrynina, and R. Beabealashvili. 2008. RNase footprinting demonstrates antigenomic hepatitis delta virus ribozyme structural rearrangement as a result of self-cleavage reaction. *BMC Res Notes* **1**: 15.
- Schenborn, E.T. and R.C. Mierendorf, Jr. 1985. A novel transcription property of SP6 and T7 RNA polymerases: dependence on template structure. *Nucleic Acids Res* **13**: 6223-6236.
- Selpi, C.H. Bryant, G.J.L. Kemp, and M. Cvijovic. 2006. A first step towards learning which uORFs regulate gene expression. *J Int Bioinformatics* **3**: 31-44.
- Shapiro, B.A., Y.G. Yingling, W. Kasprzak, and E. Bindewald. 2007. Bridging the gap in RNA structure prediction. *Curr Opin Struct Biol* **17**: 157-165.
- Shen, Y., A. Danon, and D.A. Christopher. 2001. RNA binding-proteins interact specifically with the Arabidopsis chloroplast *psbA* mRNA 5' untranslated region in a redox-dependent manner. *Plant Cell Physiol* **42**: 1071-1078.
- Shirley, B.W. and R.B. Meagher. 1990. A potential role for RNA turnover in the light regulation of plant gene expression: ribulose-1,5-bisphosphate carboxylase small subunit in soybean. *Nucleic Acids Res* **18**: 3377-3385.

- Short, J.D. and C.M. Pfarr. 2002. Translational regulation of the *JunD* messenger RNA. *J Biol Chem* **277**: 32697-32705.
- Shu, P., H. Dai, W. Gao, and E. Goldman. 2006. Inhibition of translation by consecutive rare leucine codons in *E. coli*: absence of effect of varying mRNA stability. *Gene Expr* **13**: 97-106.
- Smith, E., T.E. Meyerrose, T. Kohler, M. Namdar-Attar, N. Bab, O. Lahat, T. Noh, J. Li, M.W. Karaman, J.G. Hacia, T.T. Chen, J.A. Nolte, R. Muller, I. Bab, and B. Frenkel. 2005. Leaky ribosomal scanning in mammalian genomes: significance of histone H4 alternative translation in vivo. *Nucleic Acids Res* **33**: 1298-1308.
- Spevak, C.C., E.H. Park, A.P. Geballe, J. Pelletier, and M.S. Sachs. 2006. *her-2* upstream open reading frame effects on the use of downstream initiation codons. *Biochem Biophys Res Commun* **350**: 834-841.
- Stormo, G.D. 2000. DNA binding sites: representation and discovery. *Bioinformatics* **16**: 16-23.
- Tabuchi, T., T. Okada, T. Azuma, T. Nanmori, and T. Yasuda. 2006. Posttranscriptional regulation by the upstream open reading frame of the phosphoethanolamine N-methyltransferase gene. *Biosci Biotechnol Biochem* **70**: 2330-2334.
- Tassoni, A., M. Franceschetti, G. Tasco, R. Casadio, and N. Bagni. 2007. Cloning, functional identification and structural modelling of *Vitis vinifera* S-adenosylmethionine decarboxylase. *J Plant Physiol* **164**: 1208-1219.
- Tatusov, R.L., E.V. Koonin, and D.J. Lipman. 1997. A genomic perspective on protein families. *Science* **278**: 631-637.
- Theil, E.C. 1993. The IRE (iron regulatory element) family: structures which regulate mRNA translation or stability. *Biofactors* **4**: 87-93.
- Thompson, J.D., D.G. Higgins, and T.J. Gibson. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res* **22**: 4673-4680.
- Touzet, H. 2007. Comparative analysis of RNA genes: the caRNAC software. *Methods Mol Biol* **395**: 465-474.

- Touzet, H. and O. Perriquet. 2004. CARNAC: folding families of related RNAs. *Nucleic Acids Res* **32**: W142-145.
- Tran, M.K., C.J. Schultz, and U. Baumann. 2008. Conserved upstream open reading frames in higher plants. *BMC Genomics* **9**: 361-378.
- Urlaub, H., E. Kuhn-Holsken, and R. Luhrmann. 2008. Analyzing RNA-protein crosslinking sites in unlabeled ribonucleoprotein complexes by mass spectrometry. *Methods Mol Biol* **488**: 221-245.
- Varani, G. and W.H. McClain. 2000. The G x U wobble base pair. A fundamental building block of RNA structure crucial to RNA function in diverse biological systems. *EMBO Rep* **1**: 18-23.
- Vilela, C. and J.E. McCarthy. 2003. Regulation of fungal gene expression via short open reading frames in the mRNA 5'untranslated region. *Mol Microbiol* **49**: 859-867.
- Vilela, C., C.V. Ramirez, B. Linz, C. Rodrigues-Pousada, and J.E. McCarthy. 1999. Post-termination ribosome interactions with the 5'UTR modulate yeast mRNA stability. *EMBO J* **18**: 3139-3152.
- Walker, S.C., F.H. Scott, C. Srisawat, and D.R. Engelke. 2008. RNA affinity tags for the rapid purification and investigation of RNAs and RNA-protein complexes. *Methods Mol Biol* **488**: 23-40.
- Wall, D.P. and T. Deluca. 2007. Ortholog detection using the reciprocal smallest distance algorithm. *Methods Mol Biol* **396**: 95-110.
- Wall, D.P., H.B. Fraser, and A.E. Hirsh. 2003. Detecting putative orthologs. *Bioinformatics* **19**: 1710-1711.
- Walling, L., G.N. Drews, and R.B. Goldberg. 1986. Transcriptional and Post-Transcriptional Regulation of Soybean Seed Protein mRNA Levels. *PNAS* **83**: 2123-2127.
- Wang, L. and S.R. Wessler. 1998. Inefficient reinitiation is responsible for upstream open reading frame-mediated translational repression of the maize *R* gene. *Plant Cell* **10**: 1733-1746.
- Wang, L. and S.R. Wessler. 2001. Role of mRNA secondary structure in translational repression of the maize transcriptional activator *Lc(1,2)*. *Plant Physiol* **125**: 1380-1387.

- Wang, M.L., G. Panasyuk, J. Gwalter, I. Nemazanyy, T. Fenton, V. Filonenko, and I. Gout. 2008. Regulation of ribosomal protein S6 kinases by ubiquitination. *Biochem Biophys Res Commun* **369**: 382-387.
- Wang, X. and C.G. Proud. 2007. A novel mechanism for the control of translation initiation by amino acids, mediated by phosphorylation of eukaryotic initiation factor eIF2B. *Mol Cell Biol*.
- Wang, X.Q. and J.A. Rothnagel. 2004. 5'-untranslated regions with multiple upstream AUG codons can support low-level translation via leaky scanning and reinitiation. *Nucleic Acids Res* **32**: 1382-1391.
- Wang, Z., M. Gerstein, and M. Snyder. 2009. RNA-Seq: a revolutionary tool for transcriptomics. *Nat Rev Genet* **10**: 57-63.
- Weil, C.F. 2009. TILLING in grass species. *Plant Physiol* **149**: 158-164.
- Wiese, A., N. Elzinga, B. Wobbes, and S. Smekens. 2004. A conserved upstream open reading frame mediates sucrose-induced repression of translation. *Plant Cell* **16**: 1717-1729.
- Wiese, A., N. Elzinga, B. Wobbes, and S. Smekens. 2005. Sucrose-induced translational repression of plant bZIP-type transcription factors. *Biochem Soc Trans* **33**: 272-275.
- Wightman, B., I. Ha, and G. Ruvkun. 1993. Posttranscriptional regulation of the heterochronic gene *lin-14* by *lin-4* mediates temporal pattern formation in *C. elegans*. *Cell* **75**: 855-862.
- Wilkins, M.R., E. Gasteiger, A. Bairoch, J.C. Sanchez, K.L. Williams, R.D. Appel, and D.F. Hochstrasser. 1999. Protein identification and analysis tools in the ExPASy server. *Methods Mol Biol* **112**: 531-552.
- Wu, C., N. Amrani, A. Jacobson, and M.S. Sachs. 2007. The use of fungal *in vitro* systems for studying translational regulation. *Methods Enzymol* **429**: 203-225.
- Wuyts, J., G. Perriere, and Y. Van De Peer. 2004. The European ribosomal RNA database. *Nucleic Acids Res* **32**: D101-103.
- Xia, T., J. SantaLucia, Jr., M.E. Burkard, R. Kierzek, S.J. Schroeder, X. Jiao, C. Cox, and D.H. Turner. 1998. Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry* **37**: 14719-14735.

- Yaman, I., J. Fernandez, H. Liu, M. Caprara, A.A. Komar, A.E. Koromilas, L. Zhou, M.D. Snider, D. Scheuner, R.J. Kaufman, and M. Hatzoglou. 2003. The zipper model of translational control: a small upstream ORF is the switch that controls structural remodeling of an mRNA leader. *Cell* **113**: 519-531.
- Yao, Z., Z. Weinberg, and W.L. Ruzzo. 2006. CMfinder: a covariance model based RNA motif finding algorithm. *Bioinformatics* **22**: 445-452.
- Yoon, S. and G. De Micheli. 2006. Computational identification of microRNAs and their targets. *Birth Defects Res C Embryo Today* **78**: 118-128.
- Yu, J., S. Hu, J. Wang, G.K. Wong, S. Li, B. Liu, Y. Deng, L. Dai, Y. Zhou, X. Zhang, M. Cao, J. Liu, J. Sun, J. Tang, Y. Chen, X. Huang, W. Lin, C. Ye, W. Tong, L. Cong, J. Geng, Y. Han, L. Li, W. Li, G. Hu, X. Huang, W. Li, J. Li, Z. Liu, L. Li, J. Liu, Q. Qi, J. Liu, L. Li, T. Li, X. Wang, H. Lu, T. Wu, M. Zhu, P. Ni, H. Han, W. Dong, X. Ren, X. Feng, P. Cui, X. Li, H. Wang, X. Xu, W. Zhai, Z. Xu, J. Zhang, S. He, J. Zhang, J. Xu, K. Zhang, X. Zheng, J. Dong, W. Zeng, L. Tao, J. Ye, J. Tan, X. Ren, X. Chen, J. He, D. Liu, W. Tian, C. Tian, H. Xia, Q. Bao, G. Li, H. Gao, T. Cao, J. Wang, W. Zhao, P. Li, W. Chen, X. Wang, Y. Zhang, J. Hu, J. Wang, S. Liu, J. Yang, G. Zhang, Y. Xiong, Z. Li, L. Mao, C. Zhou, Z. Zhu, R. Chen, B. Hao, W. Zheng, S. Chen, W. Guo, G. Li, S. Liu, M. Tao, J. Wang, L. Zhu, L. Yuan, and H. Yang. 2002. A draft sequence of the rice genome (*Oryza sativa* L. ssp. *indica*). *Science* **296**: 79-92.
- Zaher, H.S. and P.J. Unrau. 2004. T7 RNA polymerase mediates fast promoter-independent extension of unstable nucleic acid complexes. *Biochemistry* **43**: 7873-7880.
- Zhang, B., Q. Wang, and X. Pan. 2007. MicroRNAs and their regulatory roles in animals and plants. *J Cell Physiol* **210**: 279-289.
- Zhang, L. and W.H. Li. 2004. Mammalian housekeeping genes evolve more slowly than tissue-specific genes. *Mol Biol Evol* **21**: 236-239.
- Zhang, Z. and F.S. Dietrich. 2005. Identification and characterization of upstream open reading frames (uORF) in the 5' untranslated regions (UTR) of genes in *Saccharomyces cerevisiae*. *Curr Genet* **48**: 77-87.
- Zhu, T., S. Pooyan, Z. Wei, M.J. Leibowitz, and S. Stein. 1996. Gel shift assay: demonstration of enhanced binding of oligo(delta)-L-ornithine-oligodeoxynucleotide conjugates to complementary DNA and RNA. *Antisense Nucleic Acid Drug Dev* **6**: 69-74.

- Zou, Z., C. Eibl, and H.U. Koop. 2003. The stem-loop region of the tobacco *psbA* 5'UTR is an important determinant of mRNA stability and translation efficiency. *Mol Genet Genomics* **269**: 340-349.
- Zuker, M. 1989. On finding all suboptimal foldings of an RNA molecule. *Science* **244**: 48-52.
- Zuker, M. 2003. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res* **31**: 3406-3415.
- Zuker, M., J.A. Jaeger, and D.H. Turner. 1991. A comparison of optimal and suboptimal RNA secondary structures predicted by free energy minimization with structures determined by phylogenetic comparison. *Nucleic Acids Res* **19**: 2707-2714.
- Zuker, M. and P. Stiegler. 1981. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res* **9**: 133-148.