

**Freie Universität Berlin, Fachbereich Mathematik und Informatik
Institut für Informatik, Studiengang Informatik**

2 März, 2006

Praktikumsbericht

von

Petko Fiziev

3808785
fiziev@inf.fu-berlin.de

über ein Praktikum bei

Center for Bioinformatics, Department of Genetics at the School of
Medicine of the [University of Pennsylvania](#) in Philadelphia, USA

von 15 Oktober, 2002 bis 15 April, 2003

Betreuer: [Dr. Artemis Hatzigeorgiou](#), agh@pcbi.upenn.edu

Inhaltsverzeichnis:

1.	Praktikumsstelle	3
2.	Aufgaben und Tätigkeiten.....	3
2.1.	Tätigkeitsumfeld	3
2.2.	Aufgabe und Ziele.....	4
2.3.	Tätigkeiten und Arbeitsergebnisse	4
3.	Einsichten und Fazit	6
3.1.	Technik	6
3.2.	Methodik	7
3.3.	Fazit	8
3.4.	Danksagung.....	8
3.5.	Referenzen	8

3143 Wörter [[Aktualisieren mit: Markieren, F9]]

Am Anfang dieses Berichtes möchte ich sagen, daß meine Arbeit in UPenn nicht als Praktikum gedacht war. Ich habe sie davor mit niemanden von der FU besprochen und erst einige Jahre später habe ich erfahren, daß es möglich ist, sie als Praxisphase anerkennen zu lassen. Obwohl das nicht empfehlenswert ist, hat die Anerkennung problemlos funktioniert.

1. Praktikumsstelle

Mein Praktikum habe ich als Research Scholar im Center for Bioinformatics, Department of Genetics at the School of Medicine of the [University of Pennsylvania](http://www.upenn.edu) in Philadelphia, USA (kurz PCBI) gemacht. Es handelt sich um eine wissenschaftliche Abteilung im Bereich der Bioinformatik. Mein Betreuer war Dr. Artemis Hatzigeorgiou.

Die Abteilung bestand aus rund 20-30 Leuten, vor allem Informatikern, aber auch vielen Statistikern und einigen Biologen. Es gab auch ungefähr 4 oder 5 Doktoranden und 1-2 Masters-Studenten. Im großen und ganzen haben die Leute in enger Kooperation mit Molekularbiologen von verschiedenen Laboren (nicht nur von UPenn) gearbeitet. Es wurden die experimentellen Daten mit algorithmischen Ansätzen analysiert und dann zusammen diskutiert. Ziel waren wissenschaftliche Publikationen. Andere Projekte in PCBI waren allgenes.org - eine Datenbank für assemblierte EST-Sequenzen, unterschiedliche Software für Gen- und Signalpeptidenvorhersage, die auch über das Web verfügbar sind, und viele mehr. Webadresse: <http://www.pcbi.upenn.edu/>

Ich habe in der Gruppe von Dr. Artemis Hatzigeorgiou gearbeitet. Sie wurde ein Jahr früher als Assistant Professor angestellt und hatte noch sehr wenige Leute in ihren Projekten engagiert. Außer mir waren noch ein Postdoc, Andrei Kouranov und ein PhD Student, Axel Bernal, teilweise unter ihrer Leitung.

Meine Betreuerin habe ich in der Firma, in der ich davor in Berlin gearbeitet habe (metaGen Pharmaceuticals GmbH), kennengelernt. Dort war ich als studentische Hilfskraft in ihren Projekten beschäftigt. Als sie die Position an der UPenn bekommen hat, hat sie mich nach Philadelphia für 6 Monate eingeladen.

Mein Aufenthalt in den USA wurde vollständig (inkl. Flug) von UPenn bezahlt. Während der Zeit bekam ich ein monatliches Gehalt von 1800\$ brutto, das mir die alltäglichen Kosten sowie die Miete und die Krankenversicherungsbeiträge ausreichend gedeckt hat.

Ich hatte keine festen Arbeitszeiten. Jedoch wurde erwartet, daß ich meine Aufgaben richtig und zeitgenau erfülle. Diese beide Faktoren führten dazu, daß ich praktisch monatelang um 11 oder 12 Uhr zur Arbeit kam und erst um 2-3 Uhr nachts nach Hause ging. Ich habe auch mehrmals in der Abteilung übernachtet. Es gab eine Küche und Couch, auch ein Badezimmer mit Dusche (es ist kein Scherz!). Zu dieser Zeit habe ich gelernt, wie wichtig es ist, die Arbeit als Programmierer richtig zu planen und den Aufwand abschätzen zu können.

2. Aufgaben und Tätigkeiten

2.1. Tätigkeitsumfeld

Ich wurde Teil von einem Projekt, dessen Ziel war, die Rolle von micro-RNAs in Säugetieren (Mensch und Maus) zu erforschen. Zu dieser Zeit war das ein neues und unbekanntes Feld, zu dem es eine Handvoll von experimentellen Daten und viele Vermutungen gab. Unsere Aufgabe war, neue Theorien anhand von algorithmischen

Ansätzen zu formulieren, die später im Labor getestet werden können. Wir haben in enger Kooperation mit der Gruppe von Zissimos Mourelatos vom Dept. of Pathology & Laboratory Medicine in UPenn gearbeitet. Wir hatten jede zwei – max. drei Wochen ein Meeting und haben dort die Fortschritte bzw. Probleme diskutiert. Mit meiner Betreuerin, Artemis, habe ich praktisch jeden Tag zu unterschiedlichen Zeiten gesprochen. Unsere Zimmer waren ca. 10 Meter voneinander entfernt. Während meines Aufenthalts habe ich auch viele wissenschaftliche Vorträge besucht, was auch oft einen Einfluß auf meine Arbeit hatte.

2.2. Aufgabe und Ziele

Ziel des Projekts war, außer die Menschheit mit neuem Wissen zu erleuchten, definitiv, unsere Arbeit als wissenschaftliche Publikation zu realisieren. Es gab auch Pläne, sie bei ISMB (einer der großen Konferenzen für Bioinformatik) zu präsentieren, die leider wegen des zeitlichen Drucks nicht erfüllt werden konnten.

Die Ziele des Projekts haben sich mit meinen persönlichen Wünschen allgemein gut ergänzt. Ich habe viel Erfahrung gesammelt, obwohl das nicht meine erste Arbeit als Programmierer war. Ich habe auch viele Kontakte mit Leuten aus UPenn gemacht, was mich nicht nur von der professionellen Seite bereichert hat. Es war auch mein erstes Mal in den USA.

2.3. Tätigkeiten und Arbeitsergebnisse

Als ich in UPenn ankam, war das Projekt noch nicht angefangen. Es gab mehrere konkrete Ziele, die grob in zwei Aufgaben aufgeteilt werden können. Erstens wollten wir die menschlichen Gene finden, die von micro-RNAs reguliert werden und zweitens wollten wir die Gene erforschen, die selbst für micro-RNAs kodieren. Wir haben beide Richtungen parallel angefangen.

Hier werde ich das erste Problem aus informatischer Sicht kurz beschreiben. Micro-RNAs (miRNAs) sind kleine RNA Moleküle in der Zelle, die als Strings mit einem Alphabet aus vier Buchstaben (A,C,T und G), die den vier Nukleinsäuren entsprechen, und Länge – ca. 20 Zeichen, dargestellt werden können. Es wurde vermutet, daß sie durch eine komplementäre Bindung an das 3' untranslatierten Ende (3'UTR) von der messenger-RNAs (mRNA) einigen Genen ihre Translation unterdrücken, und auch, daß diese sog. Targetgene eine kurze DNA-Sequenz (MRE) in diesem Abschnitt besitzen, die von einem Micro-Ribonucleoprotein-Komplex (miRNP) erkannt wird. Formal gesehen sind die messenger-RNAs der Gene auch Strings aus dem gleichen Alphabet, die nur länger sind. Um ein MRE zu finden, muß man, grob gesagt, eine String-Suche durchführen, in der man die Sequenz der micro-RNA in der Sequenz des 3'UTRs der mRNA finden soll.

Der zweite Teil des Projekts bestand darin, neue micro-RNAs durch algorithmische Methoden zu finden, die später im Labor nachgewiesen werden können. Da ich diese Richtung nach den ersten zwei Monaten völlig verlassen habe, werde ich nur kurz am Ende dieses Berichts darauf eingehen.

In Jahre 2002 war noch vieles über die Mechanismen von micro-RNA-mRNA Interaktionen unbekannt. Es wurde jedoch klar, daß, es um die MREs zu finden, nicht ausreichend ist, den reverse-complement der micro-RNA-Sequenz in die Sequenz des 3'UTRs einer mRNA exakt zu suchen. Es gab ein paar Versuche, deren Ergebnisse zeigten, daß die micro-RNA eine nicht-exakte Bindung mit der mRNA macht. Aus informatischer Sicht heißt das, daß man einen Alignment mit Gaps und Mismatches zwischen den beiden Strings ausrechnen soll. Dafür sind Algorithmen wie Smith-Waterman und Needleman-Wunsch schon lange bekannt. Man hat eine $A \times A$

Kostenmatrix (wobei A die Größe des Alphabets ist), die für alle zwei Paaren von Buchstaben eine Zahl angibt, die den Match bzw. Mismatch zwischen ihnen bewertet. Man hat auch eine sog. Gap-Kosten-Funktion, die angibt, wie Inserts und Deletions (kurz Indels) bewertet werden. Im allgemeinen besteht das Problem darin, bei zwei gegebenen Strings, das Alignment mit den geringsten Kosten zu finden. Da es im Prinzip exponentiell viele Alignments zwischen zwei Strings gibt, wird das klassische Paradigma vom dynamischen Programmieren angewendet. Es wird eine $n \times m$ Matrix (wobei n und m die Längen der zwei Strings sind) von oben links nach unten rechts ausgefüllt, wobei bei jedem Schritt immer der optimale Wert notiert wird.

Für unsere Zwecke haben wir eine modifizierte Version des Zuker-Algorithmus implementiert. Meine Betreuerin hatte den Code früher auf andere Probleme angewendet und ich mußte ihn einbauen und teilweise verändern. Ich mußte um eine C-Methode, die das dynamische Programmieren implementiert, ein ganzes Programm schreiben, was als Input eine Datei mit micro-RNA-Sequenzen und eine Datei mit mRNA-Sequenzen nimmt und als Output alle gegen alle optimal aliniert.

Ich werde hier den Algorithmus kurz skizzieren. Man gibt zwei Strings an: M (micro-RNA) und G (3'UTR der mRNA eines Gens), jeweils mit Länge m und g . Gegeben sind auch eine $A \times A$ Kostenmatrix K und eine Gap-Funktion $GAP(k)$, die die Kosten für ein Gap der Länge k zurückgibt. Man initialisiert dann die $n \times g$ Matrix, DP, mit 0 für alle Zellen und füllt sie wie folgt aus:

$$DP_{i,j} = \begin{cases} \min(DP_{r,c} + K_{M_i,G_j} + GAP(i-r) + GAP(j-c)), & \text{falls } K_{M_i,G_j} < 0 \text{ und } DP_{r,c} \neq \text{IMPOSSIBLE} \\ \text{IMPOSSIBLE,} & \text{andernfalls} \end{cases}$$

für $\text{IMPOSSIBLE} \in \mathbb{N}$ und gross genug

$c, r \in \mathbb{N}$ und $i - \text{MAX_L} \leq r \leq i$ und $j - \text{MAX_L} \leq c \leq j$.

In dieser Form findet der Algorithmus das Alignment mit den niedrigsten Kosten. Da die Kostenmatrix K auf der Änderung der freien Energie zwischen zwei Paaren von Nukleotiden basiert, ergibt die optimale Lösung die Bindung zwischen der micro-RNA und dem 3'UTR der mRNA, deren Stabilität am höchsten ist.

Gleichzeitig werden zwei andere Matrizen ausgefüllt, die die Positionen der momentan optimalen Lösung in DP halten. Wenn DP vollständig ausgefüllt ist, wird mit ihr ein Backtracking gemacht und so das Alignment rekonstruiert. Alignments, deren Kosten unter einen bestimmten Schwellwert liegen, werden als biologisch signifikant angezeigt.

Zusätzlich habe ich einige Heuristiken eingebaut, damit der Algorithmus schneller funktioniert. Es gab z.B. ein Preprocessing-Schritt, bei dem die Sequenz des 3'UTRs in einer Hashtabelle gepackt wird, so daß man perfekte Matches der Länge 3 und 4 in konstanter Zeit finden kann. Erst dann wird ein Fenster mit ausreichender Länge genommen und gegen die micro-RNA aliniert. Diese Technik ist bekannt als Seed-And-Extend und wird auch bei vielen anderen Programmen zur Beschleunigung benutzt (z.B. BLAST, BLAT usw).

Außerdem habe ich einen Filter für biologisch relevante Lösungen geschrieben. Nach langen Diskussionen mit den Leuten vom Labor sind wir zu einer Liste von strukturellen Regeln gekommen, die bestimmen, ob ein Alignment mit niedriger freier Energie tatsächlich für eine negative Kontrolle der Proteintranslation spricht. Der Filter wurde als Postprocessing-Schritt eingebaut.

Damit man die statistische Signifikanz der so gefundenen MREs abschätzen kann, habe ich dementsprechend die ganze Prozedur auf einem Set von künstlich randomisierten micro-RNA und mRNA-Sequenzen angewendet. Es kam immer noch heraus, daß der Algorithmus nicht besonders sensitiv ist. Es gab nur ca. 50% weniger Hits bei diesem Hintergrundmodell.

Am Ende habe ich eine parallele Version geschrieben, die die Arbeit auf einen Cluster von mehr als 100 Dualprozessor-Rechnern verteilt.

Im Laufe der Arbeit wurden einige von den gefundenen MREs im Labor getestet. Die meisten waren allerdings falsch, so zeigten die Ergebnisse der Versuche. Trotzdem wurden ungefähr 5-6 positiv nachgewiesen, was ein großer Erfolg war. Das hat später auch zu einer Publikation geführt [1].

Im Großen und Ganzen bin ich mit meiner Arbeit in UPenn zufrieden. Was ich dort erreicht habe, hat meine Erwartungen deutlich übertroffen. Allerdings muß ich sagen, daß es alles andere als ideal abgelaufen ist. Ich habe monatelang jeden Tag von 11 Uhr morgens bis 2-3 Uhr, und auch oftmals später, in der Nacht gearbeitet. Auch nicht so selten habe ich dort übernachtet. Da die Gruppe von Dr. Artemis Hatzigeorgiou sehr jung war, gab es auch viele organisatorische Probleme. Die Diskussionen mit ihr waren oft lang, heftig und emotional, aber im Endeffekt produktiv. Manchmal hat mir auch der Kontakt zu den Biologen gefehlt, obwohl sich ihr Labor im Nachbar-Gebäude befand. Ich habe verstanden und praktisch gesehen, wie wichtig es ist, daß man mit seinen „Kunden“ regelmäßig sprechen soll.

3. Einsichten und Fazit

3.1. Technik

Die Algorithmen für Alignments und die Heuristiken habe ich in C implementiert. Diese Sprache ist dafür bekannt, daß es oft schwierig ist, Memory-Leaks zu finden und zu entfernen. Das Problem ist, daß C keine Mechanismen gegen lokal unerlaubte Speicherzugriffe anbietet, und daß man sich als Programmierer selbst darum kümmern soll. Da das Programm keine Meldungen (wie Exceptions in Java o.ä.) ausgibt, übersieht man leicht einen Pufferüberlauf oder einen Null-Pointer. Die einzigen Möglichkeiten wären dann, ein Segmentation Fault zu beobachten, oder am Output zu erkennen, daß etwas nicht in Ordnung ist. Beide Varianten sind äußerst unattraktiv aus der Sicht des Debuggings. Selbst dann ist es schwierig, die genaue Stelle im Code zu finden, die es verursacht hat. Ich habe fast nie ein Debugger (wie gdb) benutzt, da ich es für zu umständlich halte, und immer mit printf's die Fehler gesucht. Manchmal hat das ganze Nächte gedauert. Es liegt aber irgendwie in der Natur der Sprache C, daß man mehr Aufwand damit hat. Man muß das immer berücksichtigen.

Den Filter und alle Wrappers habe ich in Perl geschrieben, die eine wesentlich höhere Programmiersprache bezüglich der Abstraktion und Readability ist. Sie stellt Konzepte wie Garbage Collection, Hashes usw. zur Verfügung und die Syntax ist extrem flexibel. Das letzte steigert enorm die Readability vom eigenen Code, aber macht es manchmal wirklich schwierig, bis man versteht, was jemand anders geschrieben hat. Der Compiler bietet auch die Möglichkeit an, Variablen zu benutzen, ohne daß sie davor deklariert wurden. Diese Option ist dafür gedacht, daß man sehr kurze Skripte leicht schreiben kann und nichts mehr. Man sollte immer „use strict“ und „use warnings“ benutzen. Ansonsten ist die Gefahr sehr groß, daß das Programm mit uninitialisierten Variablen arbeitet. Das habe ich auf die harte Art gelernt.

Die parallele Version vom Algorithmus bestand darin, daß ich praktisch das C-Programm auf mehreren Rechnern über „ssh“ gestartet habe. Dann sollte man regelmäßig prüfen, ob die Berechnungen fertig sind, und am Ende mußte man alle Outputs in eine Datei umleiten und weiter mit dem Filter verarbeiten. Diese Pipeline habe ich mit einem Perl-Skript automatisiert. Dabei gab es auch viele kleine Schwierigkeiten. Ich habe die Interface, die für die Verteilung von Jobs gedacht ist, nicht benutzt, weil ich sie zu kompliziert fand, und habe meine eigene geschrieben. Am Anfang dachte ich, daß es wirklich schneller wird, ein Perl-Skript zu schreiben, das in einer Schleife „ssh“ zu den verschiedenen Knoten des Clusters ausführt. Als ich aber diesen Fehler begriffen habe, war es schon zu spät, die Skripte zu ändern. Das war Anfang April und ich mußte zurück nach Deutschland verreisen.

3.2. Methodik

Während der Zeit, die ich in UPenn verbracht habe, habe ich ziemlich viele Eindrücke gesammelt, wie eine kleine Gruppe mit einer anderen zusammenarbeiten kann. Mit meiner Betreuerin hatte ich jeden Tag mindestens eine Besprechung, in der wir die aktuellen Problemen und die zukünftigen Ideen diskutiert haben. Es gab keine Vorschriften, daher hat sich das Projekt mehr aus sich selbst herausentwickelt. Den stärksten Einfluß darauf hatten sicherlich die Ergebnisse der Experimente und die Meetings mit den Biologen. Es war ein ständiges Hin und Her, bis klar wurde, was man genau braucht. Sie haben auch ihre Theorien nach den Vorhersagen unserer Programme gerichtet. Es hat mich überrascht, wie lange die Experimente im Labor dauern. Manchmal ein oder mehrere Monate. Was aber absolut merkwürdig für mich war, war, daß wir ungefähr genauso viel Zeit brauchen, um den Algorithmus zu ändern, testen und eine neue Vorhersage zu machen. Allgemein habe ich die Zeit, die man fürs Debugging und Testen wirklich braucht, deutlich unterschätzt. Man kann sagen, das Codieren braucht nur circa ein Drittel vom Ganzen. Eine schlechte Planung kann diese Zeiten auch enorm verlängern. Zum Beispiel lohnt sich es nie, über Nacht bei der Arbeit zu bleiben. Immer habe ich am nächsten Tag gesehen, daß ich trotz allen Mühen irgendwo noch am Anfang einen Fehler begangen habe, und daß die Ergebnisse im Endeffekt Schrott waren. Es ist wirklich viel sinnvoller, um 6 oder 7 Uhr abends nach Hause zu fahren und es dann am Morgen fortzusetzen. Oft kommen die Ideen und man sieht, wo die Fehler sind, gerade wenn man entspannt zu Hause sein Abendbrot ißt oder wenn man sich morgens die Zähne putzt. Das ist kein Scherz! Bei mir ist es wirklich so gewesen.

Eine Sache ist noch zu sagen. Ich habe mich oft sehr überschätzt, dahingehend, daß ich gleichzeitig mehrere Sachen parallel angefangen habe. Das ist eine schlechte Praxis. Man soll eher sequentiell arbeiten, wenn man programmiert. Ansonsten verliert man den Faden. Es ist schwierig, selbst nach einer Stunde den Code fortzusetzen, den man wegen einer Besprechung unterbrochen hat, die nichts damit zu tun hatte. Es ist sehr wichtig, sich zu konzentrieren und den Code bis zu einer Ebene zu bringen, wo es klar ist, was arbeitet und was nicht und was noch zu machen ist - eine Art Snapshot. Oft habe ich gedacht, daß, wenn ich zurück bin, ich mich sofort daran erinnern werde, wo ich stehengeblieben bin. Großer Fehler. Aus diesem Grund habe ich auch die zweite Richtung des Projekts nach den ersten zwei Monaten verlassen.

Das Ganze ist explodiert, als ich zurück nach Deutschland kam und Telefonanrufe aus Philadelphia auf mein Handy bekommen habe. Es ist wirklich kein Spaß, abends in einer Kneipe in Friedrichshain zu sitzen, dabei schnell rauslaufen zu müssen und draußen endlose Gespräche über Perl-Skripte zu führen. Die Leute wollten es weiter entwickeln, den Code ändern und es mit neuen Inputs laufen lassen und bei ihnen war es gerade Mittag um diese Zeit. Es war die Hölle, es zu erklären. Das schlimmste war, daß ich mich selbst nicht erinnerte, was ich genau gemacht habe und das war natürlich peinlich. Ich habe eine sehr knappe Dokumentation zurückgelassen und praktisch keine Kommentare im Code. Ein sehr oft gemachter Fehler, den zu korrigieren schwerfällt.

Obwohl ich UPenn im April 2003 verlassen habe, hat es praktisch noch mehr als ein Jahr gedauert, bis die Publikation akzeptiert wurde[1]. Ich mußte viele Kleinigkeiten remote über ssh korrigieren und einige neue Läufe vom Algorithmus mit anderen Datensätzen machen. Während dieser Zeit mußte ich mich auch natürlich um mein Hautstudium an der FU parallel kümmern. Das ist wirklich nichts, was ich jemandem empfehlen will.

3.3. Fazit

Ich empfinde das Praktikum allgemein als einen großen Erfolg. Wir haben unsere Ziele erreicht und ich habe sehr viel gelernt, was man praktisch in einer Vorlesung an der Uni nicht lernen kann. Seitdem habe ich einige meiner Gewohnheiten geändert. Nicht nur habe ich angefangen, „use strict“ und „use warnings“ zu benutzen. Ich habe mir auch mehr Zeit genommen, um die Sachen zu planen (z.B. die Hausaufgaben an der Uni usw). Die Zeit für Testen und Debugging versuche auch ich immer in Sicht zu halten.

Seitens des Programmierens habe ich mich nicht viel verbessert. Ich denke, eine Programmiersprache zu beherrschen, ist nicht etwas schwieriges. Es ist auch oft egal, ob man in C, C++ oder Java schreibt. Man sollte es allerdings vor dem Anfang überlegen. Im ganzen Prozess der Softwareentwicklung ist das Codieren oft der einzige Teil, den man wirklich alleine macht (mit Ausnahme von Pairprogramming, aber das wird sowieso selten praktiziert). Deswegen ist es das, was am schnellsten geht. Zu langem Testen und Debugging bin ich oft erst dann gekommen, als sich jemand beschwert hat, oder ich irgendwo gemerkt habe, daß die Ergebnisse nicht korrekt sind. Man muß niemals diesen Aufwand unterschätzen.

Von der sozialen Seite hat mir die Arbeit auch viel gebracht. Es ist sehr interessant zu sehen, wie Leute aus zwei verschiedenen Welten – Informatik und Biologie - zusammenarbeiten. Ein Teil davon zu sein ist auch eine unersetzbar wertvolle Erfahrung.

Zum Schluß muß ich sagen, daß der geographische Standort auch nicht zu vernachlässigen ist. Die USA sind ein Land, wo viele Sachen anders als in Deutschland ablaufen. Ein Praktikum dort zu machen, ist daher nur zu empfehlen.

3.4. Danksagung

Ich möchte hier meiner Betreuerin Dr. Artemis Hatzigeorgiou herzlich danken, daß sie mir die Möglichkeit gegeben hat, ein Teil von ihrer Gruppe für sechs Monate zu werden. Ich möchte auch Axel Bernal und alle anderen Mitarbeitern in PCBI danken, die mit mir nicht einmal meine Probleme diskutiert haben. Ich möchte auch UPenn danken, daß sie meinen Aufenthalt vollständig gedeckt haben.

Ich möchte auch Prof. Prechelt und insbesondere Frau Gesine Milde für die Verbesserung dieses Berichtes danken.

3.5. Referenzen

[1] Kiriakidou M, Nelson PT, Kouranov A, Fitziev P, Bouyioukos C, Mourelatos Z, Hatzigeorgiou A.: [A combined computational-experimental approach predicts human microRNA targets](#). Genes Dev. 2004 May 15;18(10):1165-78. Epub 2004 May 06.