The *Vmatch* large scale sequence analysis software

Stefan Kurtz

January 27, 2010

This is the web-site for *Vmatch*, a versatile software tool for efficiently solving large scale sequence matching tasks. *Vmatch* subsumes the software tool REPuter, but is much more general, with a very flexible user interface, and improved space and time requirements.

Features of Vmatch

The *Vmatch*-manual gives many examples on how to use *Vmatch*. Here are the program's most important features.

Persistent index

Usually, in a large scale matching problem, extensive portions of the sequences under consideration are static, i.e. they do not change much over time. Therefore it makes sense to preprocess this static data to extract information from it and to store this in a structured manner, allowing efficient searches. *Vmatch* does exactly this: it preprocesses a set of sequences into an index structure. This is stored as a collection of several files constituting the persistent index. The index efficiently represents all substrings of the preprocessed sequences and, unlike many other sequence comparison tools, allows matching tasks to be solved in time, *independent* of the size of the index. Different matching tasks require different parts of the index, but only the required parts of the index are accessed during the matching process.

Alphabet independency

Most software tools for sequence analysis are restricted to DNA and/or protein sequences. In contrast, *Vmatch* can process sequences over any user defined alphabet not larger than 250 symbols. *Vmatch* fully implements the concept of *symbol mappings*, denoting alphabet transformations. These allow the user to specify that different characters in the input sequences should be

considered identical in the matching process. This feature is used to group similar amino acids, for example.

Versatility

Vmatch allows a multitude of different matching tasks to be solved using the persistent index. Every matching task is basically characterized by (1) the *kind of sequences* to be matched, (2) the *kind of matches* sought, (3) additional *constraints* on the matches, and (4) the *kind of post-processing* to be done with the matches.

In the standard case, *Vmatch* matches sequences over the same alphabet. Additionally, DNA sequences can be matched against a protein sequence index in all six reading frames. Finally, DNA sequences can be transformed in all six reading frames and compared against itself.

Where appropriate, *Vmatch* can compute the following kinds of matches, using state-of-the-art algorithms:

- maximal repeats using the algorithm of [2].
- branching tandem repeats using the algorithm of [1],
- supermaximal repeats using the algorithm of [2].
- maximal substring matches using the algorithm of [26].
- maximal unique substring matches using the algorithm of [26].
- complete matches using the algorithms of [31] and [32]

To compute degenerate substring matches or degenerate repeats, each kind of match (with the exception of tandem repeats and complete matches) can be taken as an exact seed and extended by either of two different strategies:

- the *maximum error* extension strategy, as described in [27] for repeat detection,
- the *greedy* extension strategy of [49].

Matches can be selected according to their length, their E-value, their identity value, or match score.

In the standard case, a match is displayed as an alignment including positional information. Alternatively, a match can directly be postprocessed in different ways:

- *inverse output*, i.e. reporting of substrings *not* covered by a match.
- *masking* of substrings covered by a match.

- *clustering* of sequences according to the matches found.
- *chaining* of matches, i.e. finding optimal subsets of matches which do not cross, using the algorithms described in [3].
- *clustering* of matches according to pairwise sequence similarities computed by the dynamic programming algorithm of [46].
- *clustering* of matches according to the positions where they occur, following the approach of [48].

Efficient algorithms and data structures

Vmatch is based on enhanced suffix arrays described in [2]. This data structure has been shown to be as powerful as suffix trees, with the advantage of a reduced space requirement and reduced processing time. Careful implementation of the algorithms and data structures incorporated in *Vmatch* have led to exceedingly fast and robust software, allowing very large sequence sets to be processed quickly. The 32-bit version of *Vmatch* can process up to 400 million symbols, if enough memory is available. For large server class machines (e.g. SUN-Sparc/Solaris, Intel Xeon/Linux, Compaq-Alpha/Tru64) *Vmatch* is available as a 64 bit version, enabling gigabytes of sequences to be processed.

Flexible input format

The most common formats for input sequences (Fasta, Genbank, EMBL, and SWISSPROT) are accepted. The user does not have to specify the input format. It is automatically recognized. All input files can contain an arbitrary number of sequences. Gzipped compressed inputs are accepted.

Customized output and match selection

Vmatch's output can be parsed by other programs easily. Furthermore, several options allow for its customization. XML output is available and new output formats can easily be incorporated without changing *Vmatch*'s program code. Certain matches can easily be selected by user defined criteria, without intermediate output and subsequent parsing.

The parts of Vmatch

Up until now we have referred to *Vmatch* as a collection of programs. In the following we use the same name, `vmatch` (in typewriter font), for the most important program in this collection. Besides `vmatch`, there are the following programs available:

1. `mkvtree` constructs the persistent index and stores it on files.
2. `mkdna6idx` constructs an index for a DNA sequence after translating this in all six reading frames.
3. `vseqinfo` delivers information about indexed database sequences.
4. `vstree2tex` outputs a representation of the index in LaTeX-format. It can be used, for example, for educational or debugging purposes.
5. `vseqselect` selects indexed sequences satisfying specific criteria.
6. `vsubseqselect` selects substrings of a specified length range from an index.
7. `vmigrate.sh` converts an index from big endian to little endian architectures, or vice versa.
8. `vmatchselect` sort and selects matches delivered by `vmatch`.
9. `chain2dim` computes optimal chains of matches from files in *Vmatch*-format.
10. `matchcluster` computes clusters of matches from files in *Vmatch*-format.

Related tools

There are several tools which are based on the persistent index of *Vmatch*:

Genalyzer is a state of the art graphical user interface to visualize the output of *Vmatch* in form of a match graph. For details see [8].

MGA is a program to compute multiple alignments of complete genomes. For details see [22].

Multimat is a program to compute multiple exact matches between three or more genome size sequences.

PossumSearch Is a program to search for position specific scoring matrices. For details, see [5].

GenomeThreader is a software tool to compute gene structure predictions. The gene structure predictions are calculated using a similarity-based approach where additional cDNA/EST and/or protein sequences are used to predict gene structures via spliced alignments. *GenomeThreader* uses the matching capabilities of *Vmatch* to efficiently map the reference sequence to a genomic sequence. For details, see [19].

Current Usages

Following is a list of completed and ongoing projects in which *Vmatch* has been successfully used:

1. The KPATH system [17, 42], developed at the Lawrence Livermore National Laboratories, uses *Vmatch* to detect unique substrings in large collection of DNA sequences. These unique substrings serve as signatures allowing for rapid and accurate diagnostics to identify pathogen bacteria and viruses. A similar application is reported in [18].

2. In [6], *Vmatch* is used to a compute a non-redundant set from a large collection of protein sequences from Zea-Maize. Similar applications are used in the [13].

3. For the development of the Barley1 GeneChip *Vmatch* is used to search against probes.

4. The latest assembly of the Arabidopsis thaliana genome (GenBank entries of 2/19/04) contains vector sequence contaminations. For example, region 3,617,880 to 3,625,027 of chromosome II is a cloning vector. *Vmatch* was used to detect the vector contamination, see here

5. The RSA-tools [47] developed by Jacques van Helden use *Vmatch* to purge sequences before computing sequence statistics. Similar applications are reported in [23, 41, 40].

6. The program SpliceNest [9] computes gene indices and uses *Vmatch* to map clustered sequences to large genomes.

7. The oligo design program Promide [36] developed by Sven Rahmann is based on the persistent index structure of *Vmatch*. Promide uses *mkvtree* for generating the index.

8. e2g is a web-based server which efficiently maps large EST and cDNA data sets to genomic DNA. The use of *Vmatch* allows to significantly extend the size of data that can be mapped in reasonable time. e2g is available as a web service and hosts large collections of EST sequences (e.g. 4.1 million mouse ESTs of 1.87 Gbp) in a precomputed persistent index. For details see [24].

9. PlantGDB [12] provides a service called PatternSearch@PlantGDB for genome wide pattern searches in plant sequences. The service is based on *Vmatch*.

10. The Bielefeld Bioinformatics Server provides the REPuter web-service to compute repeats in complete genomes. The service is based on *Vmatch*.

11. The Mu Transposon Information Resource, used *Vmatch* to (1) match 130,861 vector-trimmed sequences against the maize repeat database, and (2) to cluster near-identical sequences. See [15] for details.

12. In [7] *Vmatch* was used to reveal long repeats inside human chromosome 1 and long similar regions between human chromosome 1 and all other human chromosomes.

13. In [38] *Vmatch* was used to cluster 317,242 EST and cDNA sequences from *Xenopus laevis*. *Vmatch* was chosen for the following reasons:

    - At first, there was no clustering tool available which could handle large data sets efficiently, and which was documented well enough to allow a detailed b replication and evaluation of existing clusters.
    - Second, *Vmatch* identifies similarities between sequences rapidly, and it provides additional options to cluster a set of sequences based on these matches. Furthermore, the *Vmatch* output provides information about how the clusters were derived. Due to the efficiency of *Vmatch*, it was possible to perform the clustering for a wide variety of parameters on the complete sequence set. This allows to study the effect of the parameter choice on the clustering.

14. In [30] *Vmatch* was used for three different tasks:

    - Searching spliced mRNA in the Arabidopsis genome to detect micromatches of length at least 20 with maximum 2 mismatches.
    - Finding matches of length at least 15 long with at most one mismatch between predicted mature miRNA-sequences and a set of ESTs as well as sequences from the Arabidopsis Small RNA Project (ASRP).
    - Aligning and performing single linkage clustering of the predicted mature miRNA sequences. Candidate pairs aligning over at least 17 bases, allowing an edit distance of 1 were grouped in the same family.

15. CrossLink [11] is a versatile computational tool which aids in visualizing relationships between RNA sequences (particularly between ncRNAs and their putative target transcripts) in an intuitive and accessible way. Besides BLAST, CrossLink uses *Vmatch* to reveal the sequence relationships to be visualized.

16. The Similarity matrix of Proteins (SIMAP) web-service [4] uses *Vmatch* to locate the sequences in SIMAP which are similar to a given query. This is much faster than running BLAST.

17. In [16], *Vmatch* is used to compute similarities between genomes, which are then visualized by the program DNAVis.

18. In [35, 45], *Vmatch* is used to search and compare repeated elements in different chloroplast DNA.

19. In [44], *Vmatch* is used to cluster EST-sequences of *Xenopus laevis*.

20. In [14] *Vmatch* is used to search exact repeats in the Macronuclear Genome Sequence of the Ciliate *Tetrahymena thermophila*.

21. PlantGDB provides a Web Service named VMatchForArabidopsis based on *Vmatch*. It allows to search sequences from *Arabidopsis Thaliana*.

22. The DOE Joint Genome Institute used *Vmatch* to identify and mask all continuous non-unique sequence fragments over 500 bp in *Frankia sp.* and *Shewanella oneidensis*.

23. In [39], Seidel et. al. describe methods for creating web-services and give examples which, among other tools, also integrate *Vmatch*.

24. In [34], Pobigaylo et. al. use *Vmatch* to map signature tags to the genome of *S. meliloti*.

25. In [28], Liang et et. al. use *Vmatch* for Vector screening.

26. The CRISPRFinder-program and the CRISPRdatabase [21, 20] make use of *Vmatch* to efficiently find maximal repeats, as a first step in localizing Clustered regularly interspaced short palindromic repeats (CRISPRs).

27. The programm *Gepard* [25] uses *mkvtree* to compute enhanced suffix arrays.

28. The *MIPSPlantsDB* database [43] uses *Vmatch* to cluster large sequence sets.

29. In [37], *Vmatch* was used to compare target genes of the tomato Chs RNAi to a tomato gene index.

30. In [29], *Vmatch* was used to search different plant genomes for matches of length at least 20 with maximum of 2 mismatches. Here the fact that *Vmatch* is an exhaustive search is important.

31. In [33], *Vmatch* was used to map millions of short sequence reads to the *A. Thaliana* genome. Up to four mismatches and up to three indels were allowed in the matching process. The seed size was chosen to be 0. The reads were aligned using the best match strategy by iteratively increasing the the allowed number of mismatches and gaps at each round.

32. In [10], *Vmatch* was used to map millions of short sequence reads to the *A. Thaliana* genome. *Vmatch* was part of a multi-step pipeline, combining a fast matching algorithm (*Vmatch*) for initial read mapping and an optimal alignment algorithm based on dynamic programming (QPALMA) for high quality detection of splice sites.

Availability

Vmatch is available in executable format for the following platforms:

- 32 bit Linux (Redhat, SuSe) for Intel and AMD architectures
- 64 bit Linux (SuSe) for Intel and AMD architectures
- 32-bit and 64-bit Solaris for the SUN/Sparc architecture
- 32-bit Solaris for Intel and AMD architectures
- Mac OSX for Apple PowerPC and Apple Intel.

If you need *Vmatch* for an additional platform, then please contact Stefan Kurtz. If you want to use *Vmatch* for academic research, educational and demonstration purposes you may obtain a free of charge non-commercial license as follows: download the license agreement form, read it, sign it, and fax it to the number given in the agreement. If you want to obtain a commercial license for *Vmatch*, then please directly contact Stefan Kurtz

Developer

Vmatch was developed since May 2000 by Stefan Kurtz, a professor of Computer Science at the Center for Bioinformatics, University of Hamburg, Germany.

References

[1] M.I. Abouelhoda, S. Kurtz, and E. Ohlebusch. The enhanced suffix array and its applications to genome analysis. In *Proceedings of the Second Workshop on Algorithms in Bioinformatics*, pages 449–463. Lecture Notes in Computer Science 2452, Springer-Verlag, 2002.

[2] M.I. Abouelhoda, S. Kurtz, and E. Ohlebusch. Replacing suffix trees with enhanced suffix arrays. *Journal of Discrete Algorithms*, 2:53–86, 2004.

[3] M.I. Abouelhoda and E. Ohlebusch. A Local Chaining Algorithm and its Applications in Comparative Genomics. In *Proc. 3rd Worksh. Algorithms in Bioinformatics (WABI 2003)*, number 2812 in Lecture Notes in Bioinformatics, pages 1–16. Springer-Verlag, 2003.

[4] R. Arnold, T. Rattei, P. Tischler, M.-D. Truong, V. Stümpflen, and H.W. Mewes. SIMAP - The similarity matrix of proteins. *Bioinformatics*, 21(Suppl. 2):ii42–ii46, 2005.

[5] M. Beckstette, R. Homann, R. Giegerich, and S. Kurtz. Fast index based algorithms and software for matching position specific scoring matrices. *BMC Bioinformatics*, 7:389, 2006.

[6] V. Brendel, S. Kurtz, and V. Walbot. Comparative genomics of Arabidopsis and Maize: Prospects and limitations. *Genome Biology*, 3(3):reviews1005.1–1005.6, 2002.

[7] P.G. Buckley, C. Jarbo, U. Menzel, T. Mathiesen, C. Scott, S.G. Gregory, C.F. Langford, and J.P. Dumanski. Comprehensive DNA Copy Number Profiling of Meningioma Using a Chromosome 1 Tiling Path Microarray identifies Novel Candidate Tumor Surpressor Loci. *Cancer Res.*, 65(7):2653–2661, 2005.

[8] J.V. Choudhuri, C. Schleiermacher, S. Kurtz, and R. Giegerich. Genalyzer: Interactive visualization of sequence similarities between entire genomes. *Bioinformatics*, 20:1964–1965, 2004.

[9] E. Coward, S.A. Haas, and M. Vingron. SpliceNest: Visualization of Gene Structure and Alternative Splicing Based on EST Clusters. *Trends Genet.*, 18(1):53–55, 2002.

[10] F. De Bona, S. Ossowski, K. Schneeberger, and G. Ratsch. Optimal spliced alignments of short sequence reads. *Bioinformatics*, 24(16):i174–180, 2008.

[11] T. Dezulian, M. Schaefer, R. Wiese, D. Weigel, and D.H. Huson. CrossLink: visualization and exploration of sequence relationships between (micro) RNAs. *Nucleic Acids Res.*, 34(Web Server Issue):W400–W404, 200.

[12] Q. Dong, C.J. Lawrence, S.D. Schlueter, M.D. Wilkerson, S. Kurtz, C. Lushbough, and V. Brendel. Comparative Plant Genomics Resources at PlantGDB. *Plant Physiology, Plant Database Focus Issue*, 2005.

[13] Q. Dong, L. Roy, M. Freeling, V. Walbot, and V. Brendel. ZmDB, an integrated Database for Maize Genome Research. *Nucleic Acids Res.*, 31:244–247, 2003.

[14] J.A. Eisen, R.S. Coyne, M. Wu, D. Wu, M. Thiagarajan, J.R. Wortman, J.H. Badger, Q. Ren, P. Amedeo, and K.M. Jones et al. Macronuclear Genome Sequence of the Ciliate Tetrahymena thermophila, a Model Eukaryote. *PLoS Biology*, 4(9):e286, 2006.

[15] J. Fernandes, Q. Dong, B. Schneider, D.J. Morrow, G.-L. Nan, V. Brendel, and V. Walbot. Genome-wide mutagenesis of Zea mays L. using RescueMu transposons. *Genome Biology*, 5(10):R82, 2004.

[16] Fiers, M.W.E.J. and Van de Wetering, H. and Peeters, T.H.J.M. and van Wijk, J.J. and Nap, J-P. DNAVis: interactive visualization of comparative genome annotations. *Bioinformatics*, 22(3):354–355, 2005.

[17] J.P. Fitch, S.N. Gardner, T.A. Kuczmarski, S. Kurtz, R. Myers, L.L. Ott, T.R. Slezak, E.A. Vitalis, A.T. Zemla, and P.M. McCready. Rapid development of nucleic acid diagnostics. *Proceedings of the IEEE*, 90(11):1708–1721, 2002.

[18] S.N. Gardner, T.A. Kuczmarski, E.A. Vitalis, and T.R. Slezak. Limitations of TaqMan PCR for Detecting Viral Pathogens I llustrated by Hepatitis A, B, C, and E Viruses and Human Immunodeficiency Virus. *J. of Clinical Microbiology*, 41(6):2417–2427, 2003.

[19] G. Gremme, V. Brendel, M.E. Sparks, and S. Kurtz. Engineering a software tool for gene prediction in higher organisms. *Information and Software Technology*, 47(15):965–978, 2005.

[20] I. Grissa, G. Vergnaud, and C. Pourcel. The CRISPRdb database and tools to display CRISPRs and to generate dictionaries of spacers and repeats. *BMC Bioinformatics*, 8:172, 2007.

[21] I. Grissa, G. Vergnaud, and C. Pourcel. CRISPRFinder: a web tool to identify clustered regularly interspaced short palindromic repeats. *Nucleic Acids Res*, 35(Web Server issue):W52–7, 2007.

[22] M. Höhl, S. Kurtz, and E. Ohlebusch. Efficient multiple genome alignment. *Bioinformatics*, 18(Suppl. 1):S312–S320, 2002.

[23] R.J.M. Hulzink, H. Weerdesteyn, A.F. Croes, M.M.A. Gerats, T. van Herpen, and J. van Helden. In Silico Identification of Putative Regulatory Sequence Elements in the 5'-Untranslated Region of Genes That Are Expressed during Male Gametogenesis Gene Coregulation. *Plant Physiol.*, 132:75–83, 2003.

[24] J. Krüger, A. Sczyrba, S. Kurtz, and R. Giegerich. e2g: An interactive web-based server for efficiently mapping large EST and cDNA sets to genomic sequences. *Nucleic Acids Res.*, 32:W301–W304, 2004.

[25] J. Krumsiek, R. Arnold, and T. Rattei. Gepard: a rapid and sensitive tool for creating dotplots on genome scale. *Bioinformatics*, 23(8):1026–8, 2007.

[26] S. Kurtz. A Time and Space Efficient Algorithm for the Substring Matching Problem, 2002.

[27] S. Kurtz, J.V. Choudhuri, E. Ohlebusch, C. Schleiermacher, J. Stoye, and R. Giegerich. REPuter: The manifold applications of repeat analysis on a genomic scale. *Nucleic Acids Res.*, 29(22):4633–4642, 2001.

[28] Liang, C. and Wang, G. and Liu, L. and Ji, G. and Liu, Y. and Chen, J. and Webb, J.S. and Reese, G. and Dean, J.F.D.

[29] M. Lindow, A. Jacobsen, S. Nygaard, Y. Mang, and A. Krogh. Intragenomic matching reveals a huge potential for mirna-mediated regulation in plants. *PLOS Comput. Biol*, 3(11):e238, 2007.

[30] M. Lindow and A. Krogh. Computational evidence for hundreds of non-conserved plant microrna. *BMC Genomics*, 6(1):119, 2005.

[31] U. Manber and E.W. Myers. Suffix Arrays: A New Method for On-Line String Searches. *SIAM Journal on Computing*, 22(5):935–948, 1993.

[32] G. Myers. A Fast Bit-Vector Algorithm for Approximate String Matching Based on Dynamic Programming. *Journal of the ACM*, 46:395–415, 1999.

[33] S. Ossowski, K. Schneeberger, R.M. Clark, C. Lanz, N. Warthmann, and D. Weigel. Sequencing of natural strains of Arabidopsis thaliana with short reads. *Genome Res.*, 18:2024–2033, 2008.

[34] N. Pobigaylo, D. Wetter, S. Szymczak, U. Schiller, S. Kurtz, F. Meyer, T.W. Nattkemper, and Becker A. Construction of a large signature-tagged mini-Tn5 transposon library and its application to mutagenesis of *Sinorhizobium meliloti*. *Appl Environ Microbiol.*, 72(6):4329–4337, 2006.

[35] J.-F. Pombert, C. Lemieux, and M. Turmel. The complete chloroplast DNA sequence of the green alga Oltmannsiellopsis viridis reveals a distinctive quadripartite architecture in the chloroplast genome of early diverging ulvophytes. *BMC Biology*, 4:3, 2006.

[36] S. Rahmann. Rapid Large-Scale Selection of Oligonucleotides for Microarrays. In *Proceedings of the First IEEE Computer Society Bioinformatics Conference (CSB 2002)*, pages 54–63. IEEE-Press, 2002.

[37] E.G.W.M. Schijlen, C.H. Ric de Vos, S. Martens, H.H. Jonker, F.M. Rosin, J.W. Molthoff, Y.M. Tikunov, G.C. Angenent, A.J. van Tunen, and A.G. Bovy. RNA interference silencing of chalcone synthase, the first step in the flavonoid biosynthesis pathway, leads to parthenocarpic tomato fruits. *Plant Physiol*, 144(3):1520–30, 2007.

[38] A. Sczyrba, M. Beckstette, A.H. Brivanlou, R. Giegerich, and C.R. Altmann. Xendb: Full length cDNA prediction and cross species mapping in *xenopus laevis*. *BMC Genomics*, 2005.

[39] P.N. Seibel, J. Krüger, S. Hartmeier, K. Schwarzer, K. Löwenthal, H. Mersch, T. Dandekar, and R. Giegerich. XML schemas for common bioinformatic data types and their application in workflow systems. *BMC Bioinformatics*, 7:490, 2006.

[40] N. Simonis, J. van Helden, G.N. Cohen, and S.J. Wodak. Transcriptional regulation of protein complexes in yeast. *Genome Biology*, 5:R33, 2004.

[41] N. Simonis, S.J. Wodak, G.N. Cohen, and J van Helden. Combining Pattern Discovery and Discriminant Analysis to Predict Gene Co-regulation. *Bioinformatics*, 20:2370–2379, 2004.

[42] T. Slezak, T. Kuczmarski, L. Ott, C. Torres, D. Medeiros, J. Smith, B. Truitt, N. Mulakken, M. Lam, E. Vitalis, A. Zemla, C.E. Zhou, and S. Gardner. Comparative Genomics Tools Applied to Bioterrorism Defense. *Briefings in Bioinformatics*, 4(2):133–149, 2003.

[43] M. Spannagl, O. Noubibou, D. Haase, L. Yang, H. Gundlach, T. Hindemitt, K. Klee, G. Haberer, H. Schoof, and K.F.X. Mayer. MIPSPlantsDB–plant database resource for integrative and comparative plant genome research. *Nucleic Acids Res*, 35(Database issue):D834–40, 2007.

[44] M. Spitzer, S. Lorkowski, P. Cullen, A. Sczyrba, and G. Fuellen. Distinguishing isoforms and paralogs on the protein level. *BMC Bioinformatics*, 7:110, 2006.

[45] M. Turmel, C. Otis, and C. Lemieux. The Chloroplast Genome Sequence of Chara vulgaris Sheds New Light into the Closest Green Algal Relatives of Land Plants. *Molecular Biology and Evolution*, 23:1324–1338, 2006.

[46] E. Ukkonen. Algorithms for Approximate String Matching. *Information and Control*, 64:100–118, 1985.

[47] J. van Helden, A.F. Rios, and J. Collado-Vides. Discovering Regulatory Elements in Non-Coding Sequences by Analysis of Spaced Dyads. *Nucleic Acids Res.*, 28(8):1808–1818, 2000.

[48] N. Volfovsky, B.J. Haas, and S.L. Salzberg. A Clustering Method for Repeat Analysis in DNA Sequences. *Genome Biology*, 2(8):research0027.1–0027.11, 2001.

[49] Z. Zhang, S. Schwartz, L. Wagner, and W. Miller. A Greedy Algorithm for Aligning DNA Sequences. *J. Comp. Biol.*, 7(1/2):203–214, 2000.