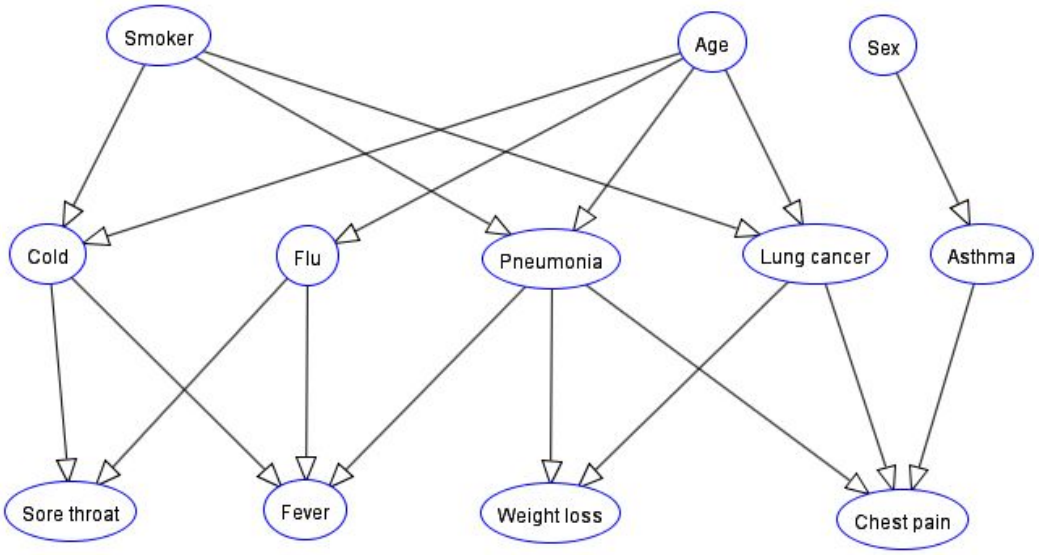
Inferența prin enumerare în rețele bayesiene, aplicată pe un model al bolilor respiratorii

Profesor coordonator:
Prof. dr. ing. Florin LEON
Studenți:
Paul-Cristian BRÎNZĂ
Andrei-Iulius COVAȘ
Mircea-Constantin DOBREANU

Problema considerată

Aplicaţia îşi propune determinarea probabilităţii ca un pacient să aibă o anumită boală respiratorie, date fiind anumite simptome şi/sau factori de risc (vârstă, sex, fumător). În realizarea scopului propus anterior, ne vom folosi de inferenţa prin enumerare în reţele bayesiene. Datele medicale vor fi preluate în principal de pe site-ul CDC-ului american sau alte site-uri de specialitate. Modelul propus de noi este ilustrat mai jos:



Aspecte teoretice privind algoritmul

Probabilitatea este fracţiunea care cuantifică aşteptarea ca un eveniment să aibă loc. Acestea poate lua valori reale cuprinse între 0 şi 1.

Probabilitatea lui A, condiţionat de B, reprezintă fracţiunea de lumi în care B este adevărată şi atunci şi A este adevărată. Probabilitatea condiţionată este măsura numerică a şansei întâmplării unui eveniment după ce s-au luat în cosiderare anumite evidenţe, dovezi.

Inferenţa este o operaţie logică de trecere de la premise la consecinţe logice.

O reţea bayesiană este un graf orientat aciclic în care toate nodurile pot fi descrise prin informaţii probabilistice. Mai riguros, o reţea bayesiană are următoarele proprietăţi:

- Oricare nod corespunde unei variabile aleatorii (fie ea discretă sau continuă).
- Dacă reţeaua conţine o muchie de la nodul X la nodul Y spunem că X este părintele (sau cauza) lui Y. Vom spune că Y este copilul (sau efectul) lui X. Reţeaua nu conţine cicluri.
- Toate variabilele din reţea (nodurile) au asociate o distribuţie de probabilitate condiţionată de stările părinţilor de forma $P(X_i | \, parents(X_i))$.

Sarcina de bază pentru orice sistem inferenţial probabilistic este să determine distribuţia de probabilităţi condiţionate a unui eveniment dat fiind anumite dovezi. Prin dovezi, ne referim la faptul că valorile altor evenimente au fost deja setate.

Modalitatea de rezolvare

În rezolvarea problemei, ne vom folosi de inferenţa prin enumerare (o metodă de inferenţă exactă). Inferenţa prin enumerare se bazează pe următoarea ecuaţie:

$$P(X|\, e) \;=\; \alpha\, P(X,\; e) \;=\; \alpha \sum_{y} P(X,\; e,\; y)$$

Cu alte cuvinte, orice probabilitate condiţionată poate fi aflată prin sumarea unor produse de probabilităţi condiţionate deja cunoscute. Pseudocodul pentru a face sumările:

function ENUMERATION-ASK(X, **e**, bn) **returns** a distribution over X
  inputs: X, the query variable
    e, observed values for variables **E**
    bn, a Bayes net with variables $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$ /* **Y** = *hidden variables* */

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL(bn.VARS, $\mathbf{e}_{x_i}$)
      where $\mathbf{e}_{x_i}$ is **e** extended with $X = x_i$
  return NORMALIZE($\mathbf{Q}(X)$)

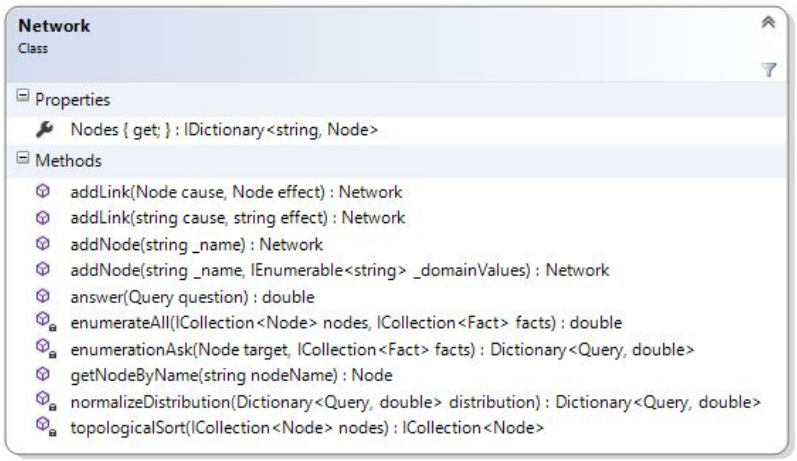
function ENUMERATE-ALL($vars$, **e**) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $Y \leftarrow$ FIRST($vars$)
  if Y has value y in **e**
    then return $P(y \mid parents(Y)) \times$ ENUMERATE-ALL(REST($vars$), **e**)
    else return $\sum_y P(y \mid parents(Y)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_y$)
      where $\mathbf{e}_y$ is **e** extended with $Y = y$

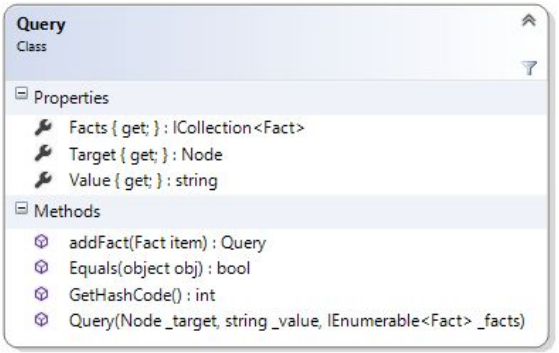
Implementarea

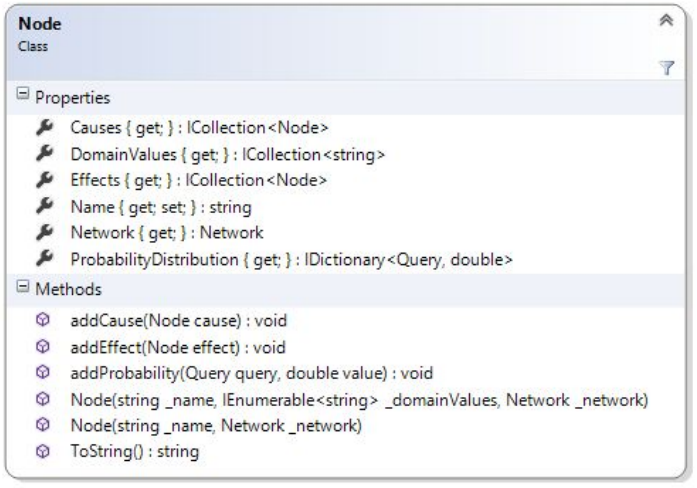
Am implementat aplicaţia în limbajul C#, iar modelul reţelei bayesiene este reprezentat într-un fişier XML. Soluţia cuprinde mai multe proiecte:

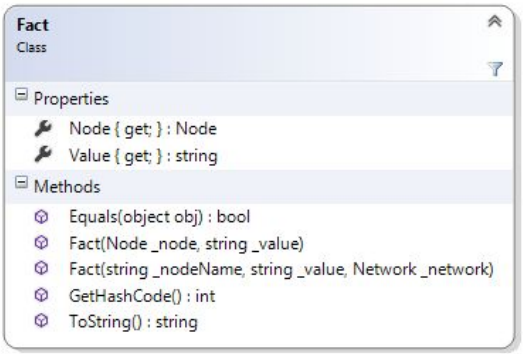
- **BayesianNetwork** - modul librărie care se ocupă cu inferenţa propriu-zisă
- **BayesianNetwork.Tests** - modul de unit tests pentru modulul de inferenţă
- **NetworkParser** - modul librărie care se ocupă de parsarea fişierelor XML
- **NetworkParser.Tests** - modul de unit tests pentru modulul de parsare
- **UserInterface** - aplicaţia WinForms care expune utilizatorului modelul propus

Diagrama de mai jos, reprezintă diagrama de clase generată de mediul de programare Visual Studio 2015 pentru modulul de inferenţă.









În continuare, este prezentat codul sursă pentru părţile relevante ale modulului, regăsite în implementarea clasei Network:

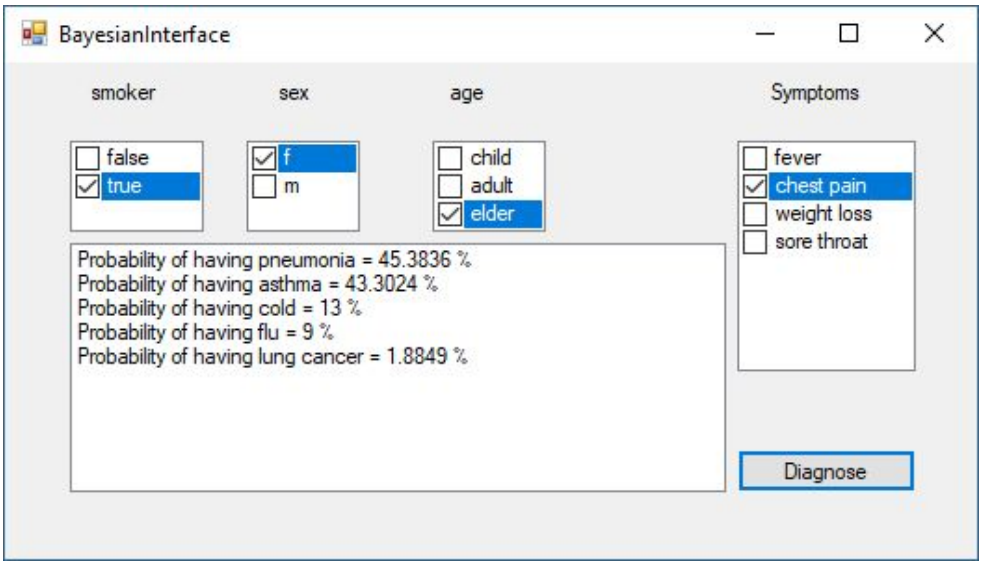
```
/**
 * Metoda wrapper pentru enumerationAsk
 */
public double answer(Query question)
{
    var distribution = enumerationAsk(question.Target, question.Facts);
    return distribution[question];
}

/**
 * Calculează distribuţia unei variabile date fiind nişte dovezi
 */
private Dictionary<Query, double> enumerationAsk(Node target, ICollection<Fact> facts)
{
    var distribution = new Dictionary<Query, double>();
    var nodes = topologicalSort(target.Network.Nodes.Values);

    foreach (var possibleValue in target.DomainValues)
    {
         var newFacts = new List<Fact>(facts);
         newFacts.Add(new Fact(target, possibleValue));

         distribution.Add(new Query(target, possibleValue, facts), enumerateAll(nodes, newFacts));
```

```
    }

    return normalizeDistribution(distribution);
}
/**
* Calculeaza probabilitatea date fiind anumite fapte. Acest fapt se realizează prin recursie, la
* fiecare pas adăugându-se fapte noi în care a fost setată o altă variabilă aleatorie
*/
private double enumerateAll(ICollection<Node> nodes, ICollection<Fact> facts)
{

    if (nodes.Count == 0)
    {
         return 1.0;
    }
    else
    {
         var head = nodes.First();

         var nodesTail = new List<Node>(nodes);
         nodesTail.Remove(head);

         bool isHeadSet = facts.Any(fact => fact.Node == head);
         if (isHeadSet)
         {
                 string headValue = facts.First(fact => fact.Node == head).Value;
                 var causalFacts = facts.Where(fact => head.Causes.Any(node => node == fact.Node));
                 double probabilityOfTarget = head.ProbabilityDistribution[new Query(head,
headValue, causalFacts)];

                 return probabilityOfTarget * enumerateAll(nodesTail, facts);
         }
         else
         {
                 double probabilityOfTarget = 0.0;
                 foreach (string headValue in head.DomainValues)
                 {
                         var causalFacts = facts.Where(fact => head.Causes.Any(node => node ==
fact.Node));
                         double probabilityOfTargetCase = head.ProbabilityDistribution[new
Query(head, headValue, causalFacts)];

                         ICollection<Fact> newFacts = new List<Fact>(facts);
                         newFacts.Add(new Fact(head, headValue));

                         probabilityOfTarget += probabilityOfTargetCase * enumerateAll(nodesTail,
newFacts);
                 }

                 return probabilityOfTarget;
         }
    }
}
/*
 * Această funcţie scalează probabilităţile obţinute fiindcă ele nu mai au suma 1 (ci mai mică)
 */
private Dictionary<Query, double> normalizeDistribution(Dictionary<Query, double> distribution)
{
    var normalized = new Dictionary<Query, double>();
    double sumOfOldProbabilities = distribution.Values.Sum();
    double alpha = 1 / sumOfOldProbabilities;
```

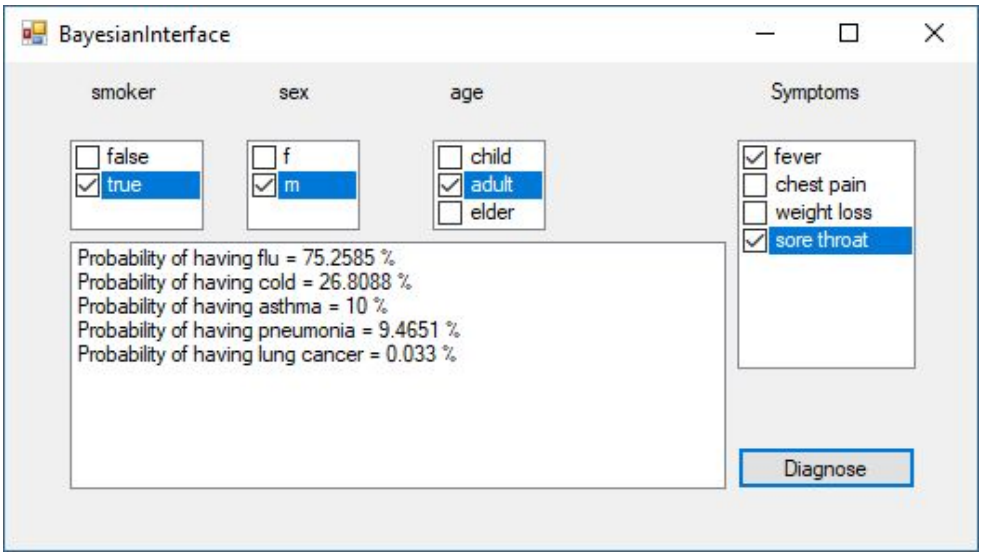
```
    foreach (var kvp in distribution)
    {
        normalized[kvp.Key] = alpha * kvp.Value;
    }

    return normalized;
}
```

Rezultate obținute

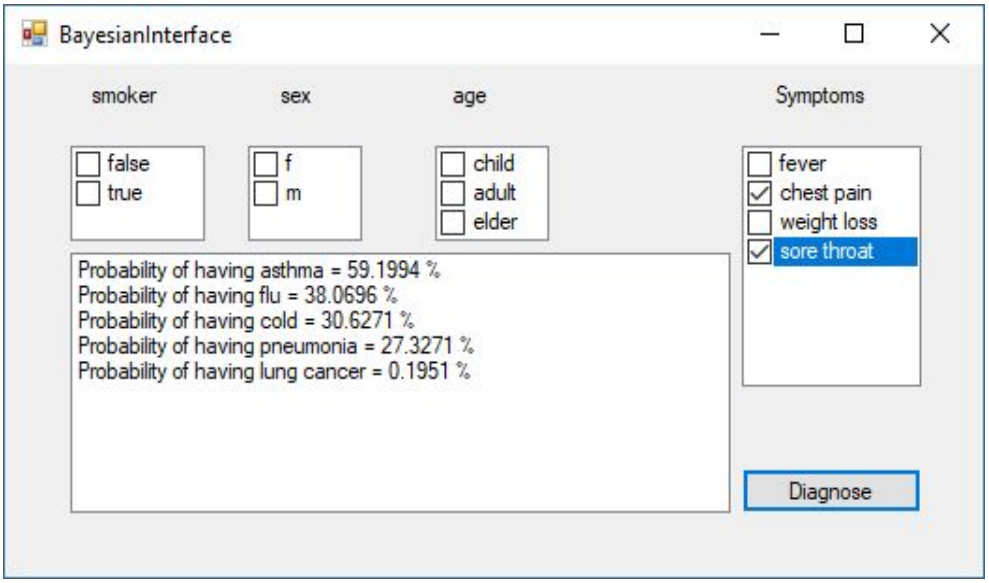
- Pentru cazul în care pacientul prezintă simptome de durere în piept(*chest pain*), este de gen feminin, face parte din categoria oamenilor în vârstă şi este fumător, rezultatele sunt următoarele:



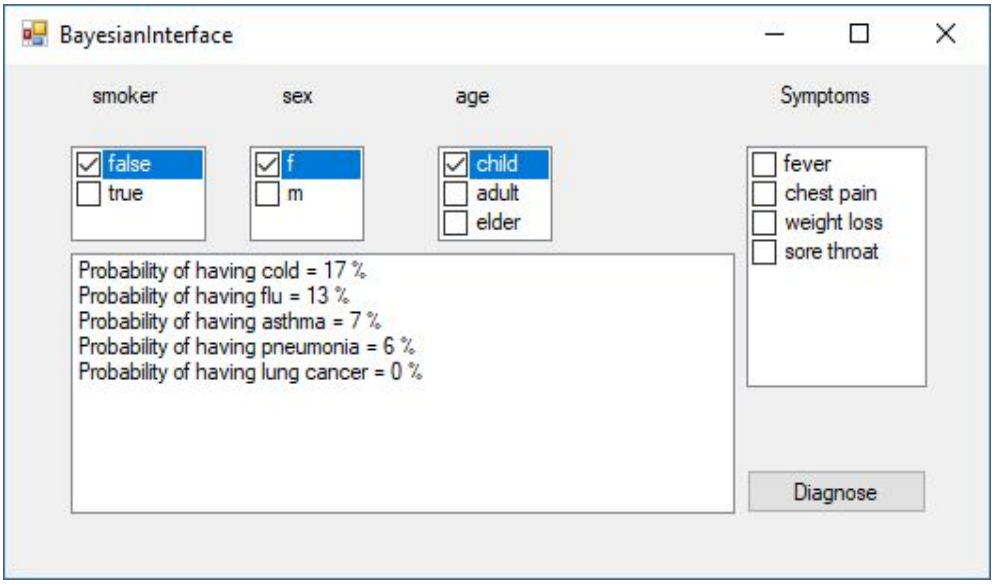
- Pentru cazul în care pacientul prezintă simptomele: febră(*fever*) şi durere în gât(*sore throat*), este adult, de gen masculin şi este fumător, rezultatele sunt următoarele:

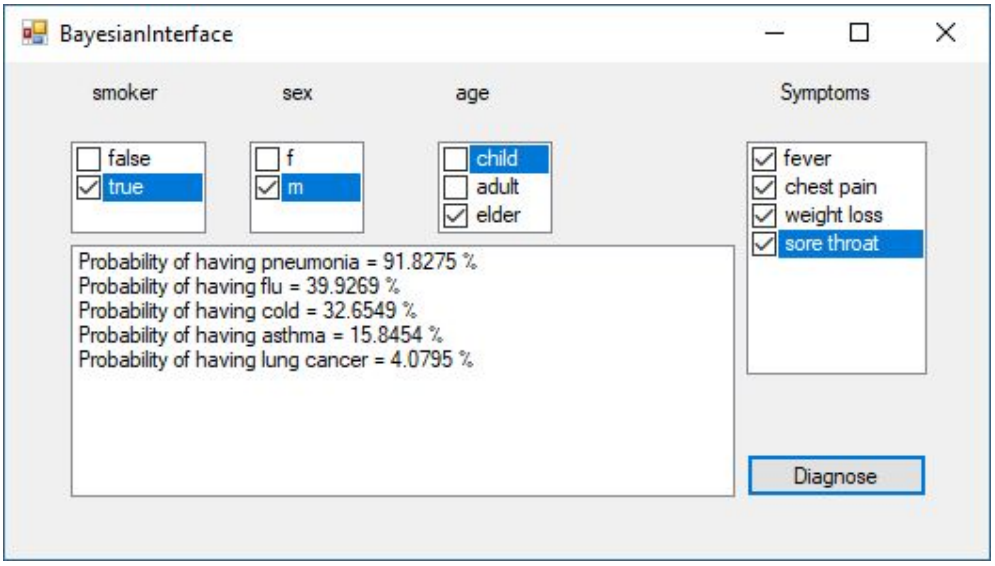


- Rezultate se obţin şi în momentul în care nu se cunosc date ca vârsta, sex-ul sau dacă pacientul este fumător sau nefumător, ci doar simptomele acestuia:



- Alte rezultate pot fi observate în continuare:





Concluzii

Având în vedere rezultatele bune obţinute în urma rulării câtorva scenarii posibile, prezentate mai sus, putem spune că metoda inferenţei prin enumerare, a fost folosită cu succes pentru rezolvarea problemei propuse. Cu toate acestea, metoda utilizată, este aplicabilă numai pe o reţea de dimensiuni reduse, gestionarea unei reţele mai mari (50 - 70 sau mai multe noduri, din care unele cu distribuţii non-binare) devenind foarte dificilă.

Contribuții

Brînză Paul-Cristian
- Documentaţie
- Modulul de parsat XML

Covaș Andrei-Iulius
- Documentaţie
- Interfaţă
- Creare model (fişierul XML)

Dobreanu Mircea-Constantin
- Modulul de logică a reţelelor bayesiene
- Documentaţie

Bibliografie

Russell, Stuart J., and Peter Norvig. *Artificial Intelligence: a Modern Approach*. Pearson, 2016.

Leon, Florin, "*Curs 10 IA - Probabilităţi*", Cursuri Inteligenţă Artificială, 2018, http://florinleon.byethost24.com/Curs_IA/IA10_Probabilitati.pdf

"Asthma." *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 15 May 2018, www.cdc.gov/asthma/most_recent_data.htm.

"Influenza (Flu)." *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 25 Oct. 2018, www.cdc.gov/flu/about/burden/2014-2015.html.

Monto, Arnold S. "Clinical Signs and Symptoms Predicting Influenza Infection." *JAMA*, American Medical Association, 27 Nov. 2000, jamanetwork.com/journals/jamainternalmedicine/fullarticle/485554.

Simasek, Madeline, and David Blandino. “Treatment of the Common Cold.” *AAFP Home*, 15 Feb. 2007, www.aafp.org/afp/2007/0215/p515.html.

“Symptoms - Chest Pain.” *LungCancer.net*, lungcancer.net/symptoms/chest-pain/.

“USCS Data Visualizations.” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, gis.cdc.gov/Cancer/USCS/DataViz.html.

“Viral Pneumonia.” *NeuroImage*, Academic Press, 22 Mar. 2011, www.sciencedirect.com/science/article/pii/S0140673610614596?via=ihub.

“The World Factbook: World.” *Central Intelligence Agency*, Central Intelligence Agency, 1 Feb. 2018, www.cia.gov/library/publications/the-world-factbook/geos/xx.html.

Arroll, Bruce “Common Cold.” *National Center for Biotechnology Information*, 16 Mar. 2011, www.ncbi.nlm.nih.gov/pmc/articles/PMC3275147/